EK-DSRVA-TM-001

Networks·Communication

DECserver 100 Terminal Server

Technical Manual

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, July 1985

Printed in U.S.A.

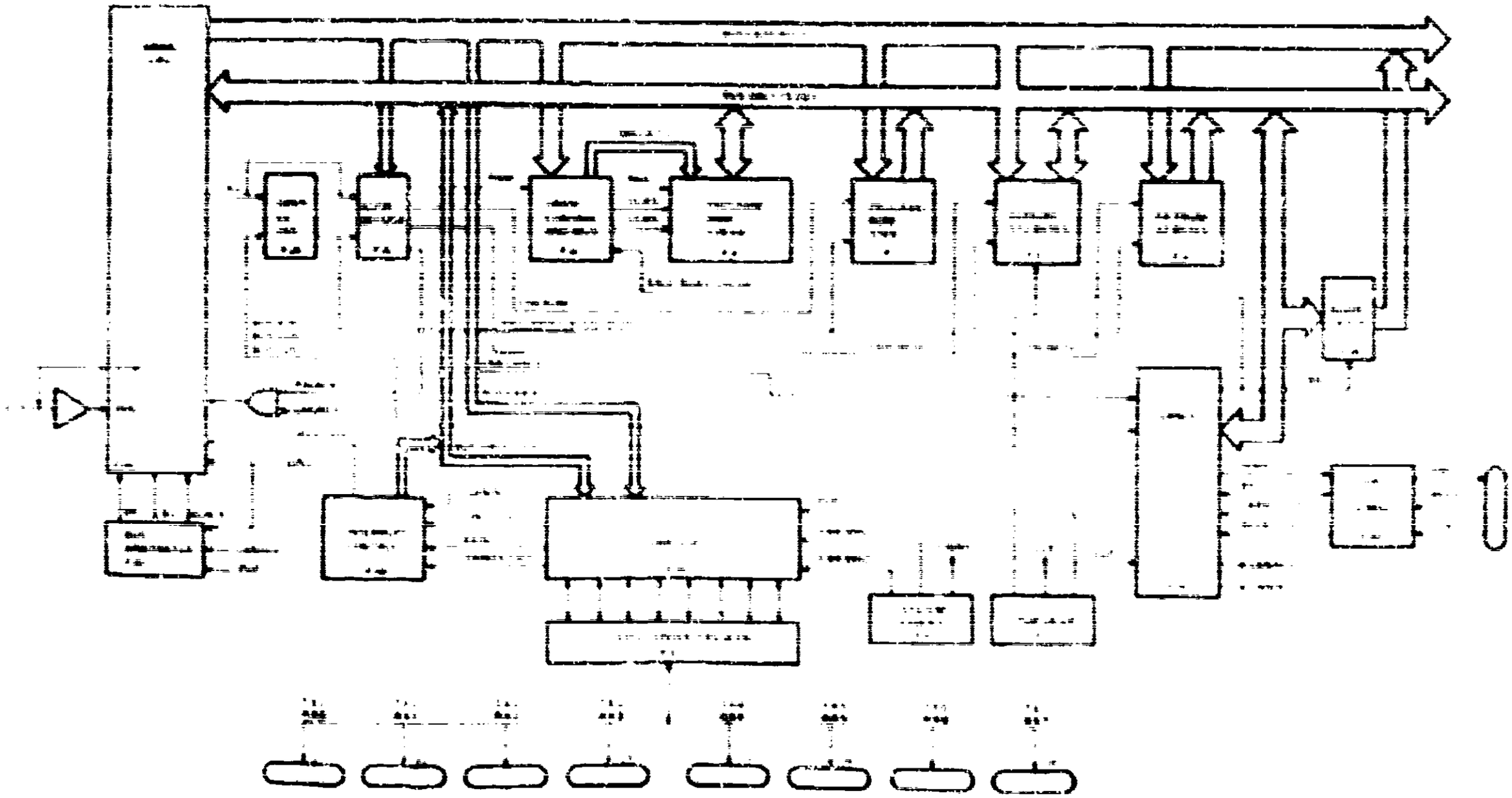


Figure 6-1: DECserver 100 Terminal Server Logic Block Diagram

2 Self-Test Program

3 Initialize Program

4 On-Line Debugging Tool (CDT)

5 Functional Logic Description

Figures

Tables

Preface

The DECserver [illegible] *Terminal Server Technical Manual* provides general operating and hardware information, as well as a technical description of the logical functions.

Intended Audience

This material is intended for use as a resource for training, field service, and manufacturing and is one of many documents supporting the DECserver [illegible] terminal server. It provides detailed technical information and assumes previous training or experience with Ethernet networks and with DIGITAL's [illegible] architecture.

Manual Structure

The manual consists of six chapters that provide the following information:

Chapter 1 – Introduces the DECserver [illegible] terminal server and the Ethernet communications system. Describes the local mode commands available to all users.

Chapter 2 – Describes the self test program, diagnostic modes and test sequences.

Chapter [illegible] – Describes the initialize program download load and upline dump procedures. Explains the status and error messages.

Chapter 4 – Describes the on-line debugging tool (ODT) commands and explains how they are used.

Chapter 5 – Describes terminal server logic functions to the block diagram level.

Chapter 6 – Lists the terminal server hardware and cable options and the physical and electrical specifications.

Other DECserver 100 Documents

- *DECserver 100 Terminal Server Site Preparation/Hardware Installation Guide* (AA-EK13A-TK)
- *DECserver 100 Terminal Server Software Installation Guide (VAX/VMS)* (AA-EJ17A-TE)
- *DECserver 100 Terminal Server Operations Guide* (AA-Z08FA-TK)
- *DECserver 100 Terminal Server User's Pocket Guide* (AV-Z08EA-TK)
- *DECserver 100 Terminal Server Identification Card* (AV-DJ35A-TK)
- *LAT Network Manager's Guide* (AA-DJ18A-TK)

Associated Documents

- *The Ethernet – A Local Area Network – Data Link Layer and Physical Layer Specifications* (Version 2.0, November 1982) (AA-K759B-TK)
- *Ethernet Installation Guide* (EK-ETHER-IN)
- *Ethernet Networks: Ethernet Products and Services Guide* (EC-J3404-12)
- *H4000 DIGITAL Ethernet Transceiver Installation Manual* (EK-H4000-IN)
- *H4000 DIGITAL Ethernet Transceiver Technical Manual* (EK-H4000-TM)
- *DELNI Installation/Owner's Manual* (EK-DELNI-IN)
- *Installing Ethernet* (EK-DEXIE-IN)

CHAPTER 1

1
Introduction

1.1 General

This chapter introduces the DEC server 100 Terminal Server and the Ethernet communications system. It also describes the local mode commands available to all users and explains their use.

1.2 [illegible] Server 100 Terminal Server

The [illegible] hardware consists of [illegible] and is mounted in a metal enclosure that provides radio frequency (RF) shielding. The unit is powered [illegible]

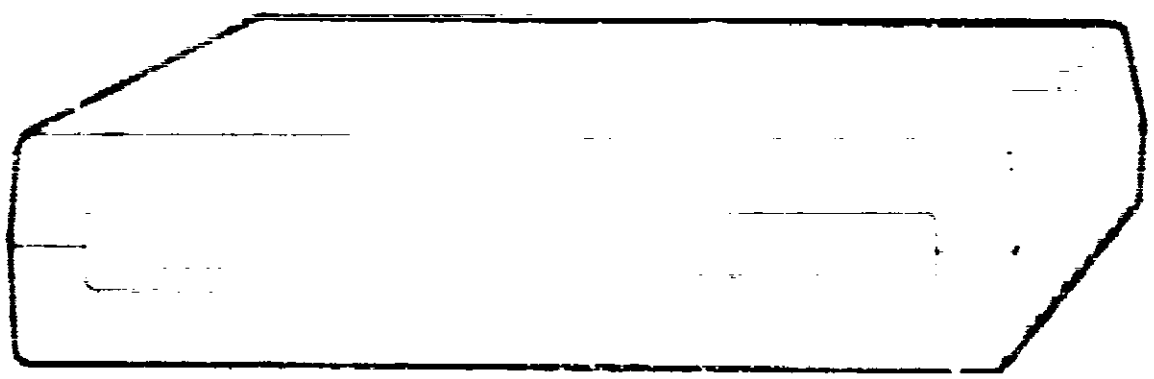


Figure 1-1 DECserver 100 Terminal Server

- DSRVA-AA - for installations using 120 V ac power
- DSRVA-AB - for installations using 240 V ac power

The DECserver 100 Terminal Server is a free-standing, intelligent terminal server that can [illegible] up to eight asynchronous [illegible] to one or more processors (called servers) on an Ethernet local area network. It may be connected directly to the network through an H4000 DIGITAL Ethernet transceiver or may be part of an expanded system through a DIGITAL Ethernet local network interconnect (DELNI) unit.

When a user is connected to a service node, response times and throughput are comparable with a directly connected on-line terminal. Any number of terminal servers and service nodes can be installed on a network within the limits described in the Ethernet specification.

1.2.1 Service Node and Load Host

Processors supporting the DECserver 100 Terminal Server on an Ethernet system provide the following types of service:

- Service node - Any processor on the local area network (LAN) that implements the local area transport (LAT) communication protocol. This may also be a specialized LAN server providing communication capabilities to non-LAN nodes.
- Load host - Service nodes on the LAN which also implement the DECnet Phase IV [illegible] down-line load the LAT operating software to any requesting server.

1.2 Firmware and Software

The DECserver 100 is supplied with the following integrated preboot and operational software:

- Self-test program
- Initialization program
- Down-line loading program (MOP)

[illegible] LAT operating software from a host [illegible] following the nine steps [illegible]

1. When power is applied to the terminal server, the self-test program performs a series of diagnostic tests and then calls the initialize program.

2. The initialize program transmits a request for a downline load of the LAT operating software from an [illegible] on the LAN.

3. When suitable configured Phase IV DECnet nodes volunteer to load the LAT software, the DECnet node from the first one that responds.

4. Following the load, the LAT software starts up and continues execution until the next power-up sequence or software initialization. At this time, the server is fully operational and terminal users can gain access to any authorized service node.

1.2.3 Controls and Connectors

A light-emitting diode (LED) on the rear panel indicates the terminal server hardware status. Figure 1-2 shows the rear panel features of the cable connectors, switches, and the LED. Table 1-1 describes their functions.

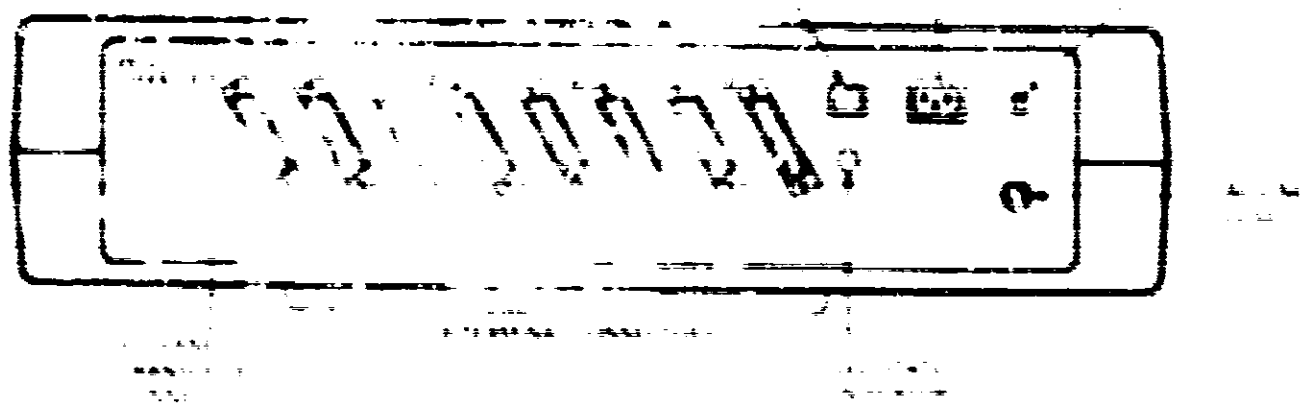


Figure 1-2 DECserver 100 Terminal Server Rear Panel

Table 1-1: Rear Panel Element Description

Element	Function
LED Indicator State	
On	The server has successfully passed the power up self tests and requested a downline load of the LAT operating software from a Phase IV DECnet host. If the load is successful, the LAT operating software (the server software) is in place and the server is fully operational.
Blinking	The server has failed one or more of the power up diagnostic self tests. The problem may be related to the network connection or may indicate a non-fatal hardware error in the server or terminal lines. (Section 4 of the *DECserver 100 Terminal Server Operations Guide* provides troubleshooting information on this condition.)
OFF	Power is off or fatal error encountered. If a fatal error occurs, the server hardware has failed and must be replaced.
Reset-to-Factory-Settings Switch (S1)	Used to reset the permanent server and terminal characteristics to their factory set values. The switch is used as follows when the user is required or when the current values cause the server to be unusable:
	1 Press and hold the reset-to-factory-settings switch.
	2 Unplug the AC power cord.
	3 Plug in the AC power cord.
	4 Hold the reset-to-factory-settings switch for 1 second then release.
Terminal Connection (25-pin)	Eight 25-pin female connectors each connect a serial asynchronous terminal to the terminal server through a BC22J-nn null modem cable or BC22F-nn straight-through cable.

Table 1-1 Rear Panel Element Description (Cont)

Element	Function
Ethernet Transceiver Connector (15)	One 15-pin female connector connects the terminal server to an Ethernet DIGITAL Ethernet transceiver or DELNI through a BNE3A transceiver cable
AC Line Voltage Input Selector Switch	Allows the installer to set the server for 47 - 63 Hz input power as follows: [illegible] for 100 - 120 Vac [illegible] for 220 - 240 Vac
AC Power Cord Receptacle	A country-specific power cord is supplied with each terminal server
AC Line Fuse	The fuse depends on the server installation. [illegible] (A, 1.0A, 5 H, 250 V ac) [illegible] (mm, 1.0A, 250 V ac)

1.2.4 Power-Up and Initialization

When the terminal server is installed and power is applied, a terminal must be plugged into the [illegible] port (port 1) for the self-test results or the console load messages to be reported.

The terminal must be set to the factory-set default characteristics listed in Table 1-2. When the server is in operation, the server manager may define another terminal as the console or change the characteristics.

Table 1-2 Basic Factory-Set Default Terminal Characteristics

Characteristic	Default Value
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

Power-up and initialize sequence – When the terminal server is plugged into an ac power source, it initiates a power-up sequence that performs the following functions:

1. The terminal server starts up the self-test program that performs a series of tests on program ROM and RAM and verifies the operating state of the server hardware. On a fatal error, the self-test halts without turning on the LED indicator.

2. The self-test turns on the LED indicator if no errors are seen. It then checks for nonfatal self-test errors. It then calls the initialize program that requests a download load of the server software (the LAT operating software).

3. The console terminal displays a number of messages in the following format that display the progress of the downline load and report any problems or error conditions:

   Local -xxx- Message

   In the message, xxx represents a decimal value of error code.

4. When the downline load is complete, the console terminal displays the "Image load complete" message. The terminal server is connected to the Ethernet system and is ready for local or on-line operation.

5. When the terminal server is fully operational, the server manager may invoke the self-test and initialize programs at any time.

1.2.5 Operating Modes

The server software provides two user operating modes under normal operation:

- **Local mode** – Users communicate with the terminal server and connect to service nodes using the local mode commands.

- **Service mode** – Users communicate with connected service nodes.

1.2.5.1 Local Mode Operation – Local mode provides the following types of command functions:

- Nonprivileged commands – The nonprivileged commands are designed for use by all terminal users.
- Privileged commands – The privileged commands are designed for the terminal server manager responsible for setting up and monitoring server operations.

The nonprivileged commands allow users to:

- Connect to and disconnect from service nodes on the network
- Set and show terminal characteristics
- Show available services, nodes, current sessions and other system information
- Make use of other functions such as terminal testing, locking and broadcast functions

In addition, the privileged commands allow the server manager to:

- Configure the operating characteristics for all terminals
- Set up the server operating characteristics
- Perform Ethernet and terminal loopback tests
- Zero all counters
- Control user access to the terminal server and service nodes
- Initialize the terminal server
- Perform extended diagnostic self-test functions

1.2.5.2 Service Mode Operation – In service mode, terminal users perform on-line operations with any service node they are authorized to access. Users may connect and maintain sessions with up to four systems at once.

Except for a few control key functions (XOFF, XON, BREAK and SWITCH characters), the server software does not process characters but passes them between the terminal and the service node. Printable keyboard characters are echoed in the service node.

1.2.5.3 Application Modes – When a user logs in on the terminal server, the terminal-to-service node operating mode may be set by command or determined dynamically by preset characteristics.

The terminal server supports the following operating modes and speed ranges:

- Interactive terminal mode – 75 to 19,200 bits/s
- Flow-controlled block mode or file-transfer mode – 75 to 9,600 bits/s

Block mode applications supporting XOFF/XON flow control are supported. The main use of this mode is by intelligent video terminals that support screen editing and send an entire screen of characters to the service node.

File transfer operations are supported for the following DIGITAL personal computers at speeds of up to 9,600 bits/s:

- Professional 350/380 series
- Rainbow 100, 100+
- DECmate II

Chapter 6 provides a list of the currently approved DIGITAL terminals.

1.2.6 Local Mode Commands

The local mode commands consist of the following three functional groups:

- Operating commands – Used to make connection and perform sessions with authorized service nodes – described in Table 1-3.
- Terminal commands – Used to find information on and set terminal characteristics as described in Table 1-4.
- System commands – Used to find information on system services, states and sessions as described in Table 1-5.

Table 1-3: Local Mode Operating Commands

Command	Description
Connect/Disconnect Commands	
CONNECT SERVICE service-name	Connects the terminal to the specified service. Users may connect to as many as four services if desired, or as many as defined by the server manager. Each session then has an assigned number (1 - 4) that may be referenced in certain commands. A list of available services may be displayed with the SHOW SERVICES command.
DISCONNECT	Disconnects the terminal from the current session.
DISCONNECT ALL	Disconnects the terminal from all active sessions.
DISCONNECT SESSION n	Disconnects the terminal from session n (1 - 4).
LOCK	Allows the user to lock the terminal, leaving all sessions active. The user supplies a password at the "Lock password:" prompt. The password then be used to regain control of the terminal in the local mode.
LOGOUT	Terminates the user's entire terminal session, disconnecting from all connected nodes and logging out the user from the terminal server.
Break/Resume Session Commands	
BREAK key	Leaves the service mode on the current session and enters the local mode. A break character may also be defined for this function.
BACKWARDS	Resumes the previous session in service mode. A switch character may also be defined for this function.
FORWARD	Resumes the next session in the service mode. A switch character may also be defined for this function.
RESUME	Resumes the current session in the service mode.
RESUME SESSION n	Resumes the designated session (1 - 4) in the service mode.

Table 1-4: [illegible] Mode Terminal Commands

Command	Description
BROADCAST TERMINAL n "text"	Allows the user to send a message of up to about 120 characters to another user on the server. (Omitting the quotes displays it in uppercase. The message is displayed only if the broadcast characteristic is enabled for the receiving terminal.)
SET TERMINAL characteristic	Sets the dynamic terminal characteristics listed in Table 1-5. Two or more characteristics may be entered if they are separated by a comma, space, or both. The dynamic characteristics remain in effect until the user logs out.
DEFINE TERMINAL characteristic	Sets the permanent terminal characteristics in EEPROM listed in Table 1-5. Two or more characteristics may be entered if separated by a comma, a space, or both. DEFINE does not change the dynamic characteristics currently in effect. The permanent values in EEPROM are placed in effect only after a terminal user logs out or on a server initialization.
HELP	Displays a list of the basic local mode commands. The optional syntax for each command is displayed in brackets.
HELP MORE	Displays a list of the advanced local mode commands.
HELP TERMINAL	Displays the user terminal characteristics that are valid for the SET and DEFINE commands.
SHOW TERMINAL	Displays the current characteristics for the [illegible] terminal.
SHOW TERMINAL ALL	Displays the current characteristics for all terminals.
SHOW TERMINAL n	Displays the current characteristics for the specified terminal.
TEST TERMINAL [WIDTH nn] [COLOR n]	Displays a rotating ASCII pattern on the terminal with a certain WIDTH in [illegible] characters. If specified, the [illegible] have been. If the line COLOR n is not specified, the display pattern continues until it is stopped by pressing any character key or the BREAK key.

Table 1-6: Local Mode System Commands

Command	Description
SHOW COUNTERS	Displays information on the Ethernet datalink and the terminal server LAT counters
SHOW NODES	Displays a list of service nodes providing the services to which the user has access. Provides status and descriptor information on each node and specifies those with reachable or unknown status
SHOW NODES ALL	Same as the SHOW NODES command but includes nodes that are currently unreachable
SHOW NODES *node-name*	Displays detailed information for the specified node
SHOW SERVER	Displays terminal server information including the Ethernet address and the server uptime, status, and characteristics that can be set
SHOW SERVICES	Displays a list of the authorized services for the user. Provides the status and descriptor information on each service and specifies those with available or unknown status.
SHOW SERVICES ALL	Same as the SHOW SERVICES command, but includes currently unavailable services
SHOW SERVICES *service-name*	Displays a list of authorized nodes that provide the specified service and includes status and descriptor information
SHOW SESSIONS	Displays all sessions for the user's terminal and provides the session number with the service name and description
SHOW SESSIONS ALL	Same as the SHOW SESSIONS command for all active terminals
SHOW SESSIONS TERMINAL *n*	Same as the SHOW SESSIONS command for the specified terminal
SHOW USERS	Displays the terminal number, user name, status, and active services for each active terminal

A command is entered by typing it on one line. Pressing the RETURN key terminates and executes the command.

Commands may be entered in either upper or lower case. Each entry word may be abbreviated to the extent that it remains unique. Entries shown in [square brackets] are optional for the user.

The local mode commands are available to all users and are listed in the *DECserver 100 Terminal Server User's Pocket Guide*.

1.2.8.1 Control Keys – The following control keys/characters are implemented to facilitate using the local mode commands:

- DELETE key – Deletes the last character of the current line
- CTRL/U – Deletes the current line and returns a prompt
- CTRL/R – Redisplays the current line and returns a prompt

1.2.8.2 Logging In – Users log in on a terminal by pressing RETURN until the following message and prompt appear.

```
DECserver 100 Terminal Server V1.0 (BL6) + LAT V5.5
Enter username>
```

When the user types in the username and presses RETURN, the server responds with the Local> prompt. The user may also be required at the option of the server manager to enter a login password at the pound sign (#) prompt in order to gain access to the server.

1.2.8.3 Setting Terminal Characteristics – The characteristics for each terminal can be changed by the user with the SET TERMINAL and DEFINE TERMINAL commands.

Table 1-6 lists the terminal characteristics and shows the command syntax and factory-set default values.

Table 1-6 User Terminal Characteristics

Characteristic	Default Value Selectable Value
[illegible]	
[illegible]	NONE* [illegible]
[illegible]	DISABLED ENABLED
[illegible]	
[illegible]	NONE [illegible]
[illegible]	NONE [illegible]
[illegible]	NONE [illegible]
[illegible]	
[illegible]	ENABLED [illegible]
[illegible]	ENABLED [illegible]
[illegible]	ENABLED [illegible]
[illegible]	
[illegible]	HARDCOPY [illegible] [illegible] [illegible]
[illegible] [illegible]	NONE [illegible]

*[illegible]

Table 1-6. User Terminal Characteristics (Cont)

Characteristic	Default Value/ Selectable Value
Terminal Control	
CHARACTER SIZE	8* 7
PARITY	NONE EVEN ODD
INPUT SPEED	9600 speed
OUTPUT SPEED	9600 speed
SPEED	9600 speed
INT. FLOW CONTROL	ENABLED DISABLED
OUTPUT FLOW CONTROL	ENABLED DISABLED
FLOW CONTROL	ENABLED DISABLED
CONNECTION ALARM	ENABLED DISABLED

* Bold type indicates the default value

Table 1-7 describes the function of each characteristic.

Table 1-7: Description of the Terminal Characteristic Functions

Characteristic	Description
Preferred Service	
PREFERRED SERVICE	Specifies a preferred service node for auto-connecting on a LOGIN or a CONNECT command.
AUTOCONNECT	When enabled and a dedicated or preferred service is specified, the user connects the terminal to the specified service on a LOGIN. The service also attempts to reconnect the terminal on a connection dropout or on a failure to make connection with the specified service node. For a preferred service, connection attempts are made at the service node until the user re-enters local mode (Local mode is not implemented for dedicated services).
Service Characteristics	
HARDWARE SWITCH	Specifies a hex character for switching back to the previous session.
LOCAL SWITCH	Specifies a hex character for switching back to the local session. Switch characters are interpreted by the server and are not sent to the host session or echoed back to the terminal.
Messages	
BROADCAST	Enables or disables whether broadcast messages from the host are displayed.
MESSAGES	Enables or disables whether the following codes are displayed with user or error messages:
VERIFICATION	Enables or disables whether verification messages are displayed when the terminal connects to a service.
Terminal Emulation	

Table 1-7: Description of the Terminal Characteristic Functions (Cont)

Characteristic	Description
TYPE	Specifies the type of terminal being used on the port and selects some of its functions.
	HARDCOPY The DELETE key echoes the entered key between slashes.
	SOFTCOPY The DELETE key erases the last character and backspaces the cursor.
	ANSI The VT100 ANSI escape sequences are supported. The DELETE key erases the last character and backspaces the cursor.
	OTHER The driver is not a terminal status messages and the local mode are disabled. Corrections must be made in Autocount use by the server managers with a printing terminal.
[illegible]	Specifies a name up to 12 characters for the port terminal when logging into the server. The name must be enclosed in quotes.
[illegible]	
CHARACTER SIZE	Sets the number of bits per character for the port. Valid values are 7 or 8.
PARITY	Specifies the type of parity being used on the terminal. Valid values are NONE, EVEN, or ODD.
INPUT SPEED	Sets the server input baud rate of the terminal. The input and output speeds must be the same if Autobaud is enabled on the server manager. Refer to Appendix A for the allowable speeds.
OUTPUT SPEED	Sets the server output baud rate of the terminal. The input and output speeds must be the same if Autobaud is enabled on the server manager. Refer to Appendix A for the allowable speeds.

Table 1-7. Description of the Terminal Characteristic Functions (Cont)

Characteristic	Description
SPEED	Sets the INPUT SPEED and OUTPUT SPEED to the same value. See Table 1-8.
INPUT FLOW CONTROL	Enables/disables XOFF-XON from the server to the terminal
OUTPUT FLOW CONTROL	Enables/disables whether XOFF-XON from the terminal should be recognized in the server
FLOW CONTROL	Enables/disables INPUT FLOW CONTROL and OUTPUT FLOW CONTROL together
DATA NOTIFICATION	If enabled the server returns a DATA READY code over the terminal line when accepting data from the keyboard (the input buffer is full or a data error occurred)

Table 1-8 lists the terminal transmit and receive speeds recognized by the terminal server.

Table 1-8: Terminal Transmit/Receive Speeds and Stop Bits

Speed	Stop Bits	Speed	Stop Bits	Speed	Stop Bits
75	2	300	1	2000	1
110	2	600	1	2400	1
134.5	2	1200	1	4800	1
150	2	1800	1	9600	1
				19200	1

1.2.8.4 Status and Error Message Types – Table 1-9 lists the types of message codes that may be returned by the server software during operation. Status and error messages are displayed in the following format where xxx (unless underlined) is a decimal status or error code.

%MOP-I-xxx, Command completed or error detected

Table 1-9: Server Status and Error Message Types

Code Range	Message Type
000 to 099	Informational messages, normal command completion
100 to 199	Warning messages
200 to 299	Command error messages
300 to 399	Server specific informational messages
400 to 499	Server specific warning messages
500 to 599	Server user and command error messages
600 to 999	Status and error messages issued by the firmware resident program MOP

1.3 Ethernet Systems

This is an introduction to the methods used and supported by devices connected to an Ethernet system. For further details, refer to the Ethernet specification.

Ethernet is a local area network (LAN) communication method based on a specification jointly developed by the following companies:

- Digital Equipment Corporation
- The Intel Corporation
- The Xerox Corporation

The local area network (LAN) communication system supports a 10 Mbit data rate over interconnected coaxial cables. The specification defines the physical properties of the coaxial cable, the transceiver, and some of the connected hardware. It also defines the basic rules for network access, device addressing, and data frame format.

The Ethernet system provides a facility for high-speed data exchanges between computers and other digital devices in a moderately sized geographic area. It fills the gap between long-distance, low-speed communication networks that carry data for hundreds of thousands of kilometers, and the specialized high speed interconnections that are limited to a few tens of meters. Its main use is in such areas as office automation and in distributed data processing requiring terminal access

1.3.1 System Elements

Figure 1-3 shows a basic Ethernet system that may be constructed using the following hardware elements:

- **Ethernet coaxial cable** - A 50-ohm coaxial cable segment of up to 500 meters provides the connection medium for all nodes and stations.
- **Ethernet transceiver** - The H4000 DIGITAL Ethernet transceiver connects directly to the coaxial cable without cutting the cable or disrupting communication.
- **Transceiver cable** - A shielded cable containing four shielded twisted pairs connects the transceiver with a DEC server 100, a processor, or a communication station.
- **Repeater (optional)** - The network may be extended by connecting another cable segment through a bidirectional amplifier (repeater).

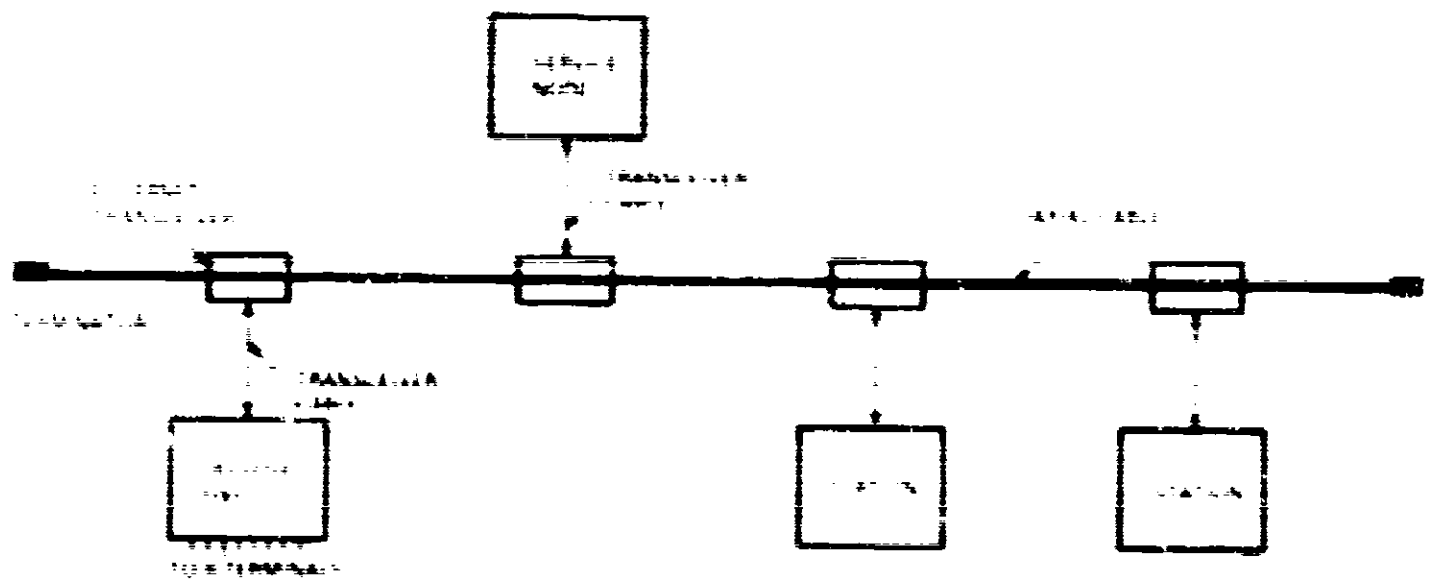


Figure 1-3. Basic Ethernet Configuration

1.3.2 H4000 DIGITAL Ethernet Transceiver

Figure 1-4 shows an Ethernet transceiver connected to a coaxial cable segment. It also shows the DECserver 700 cable and connector.

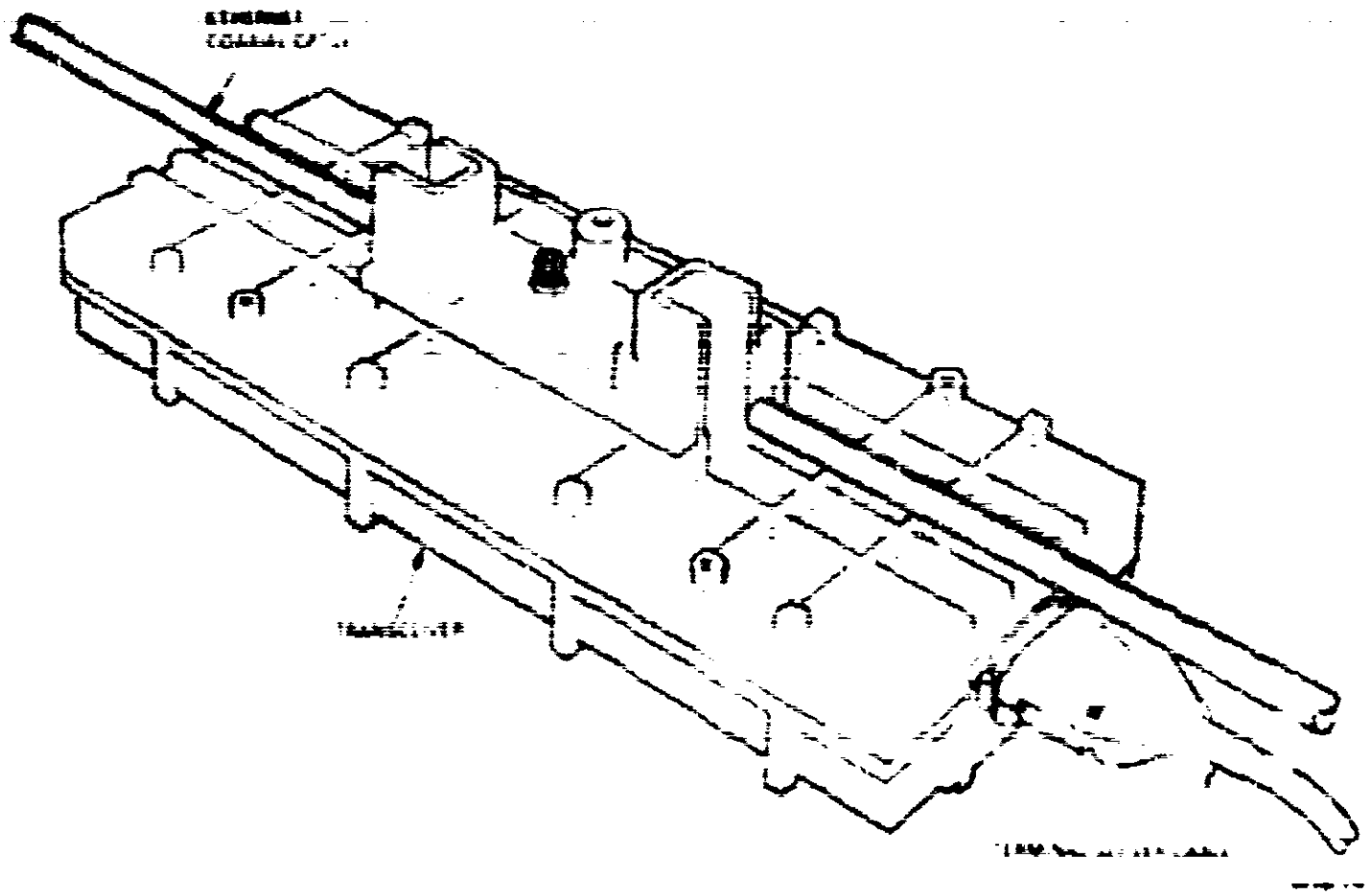


Figure 1-4 H4000 DIGITAL Ethernet Transceiver

1.3.3 Configuration Criteria

Physical limits are imposed on Ethernet networks to ensure optimum performance. The considerations for configuring an Ethernet system are as follows:

- A single coaxial cable segment may be of any length up to 500 meters (1640 feet) and must be terminated at both ends in its characteristic impedance (50 ohms).
- Up to 100 stations may be installed on a single coaxial cable segment, but transceivers must be spaced at least 2.5 meters (8.2 feet) apart.
- A transceiver cable may be of any length up to 50 meters (164 feet).
- A repeater may be used to extend the system by connecting two coaxial cable segments through two transceivers as shown in Figure 1-5.
- The network may also be extended using two remote repeaters with an optical point-to-point link of up to 1,000 meters (3,280 feet) as shown in Figure 1-6.
- Up to two repeater or remote repeater combinations may be used between any two points on the system.
- Up to 1,024 stations may be combined in an extended network. Such a network must be configured with no more than 2.5 km (1.5 miles) of separation between any two stations on the system including point-to-point links and transceiver cables.

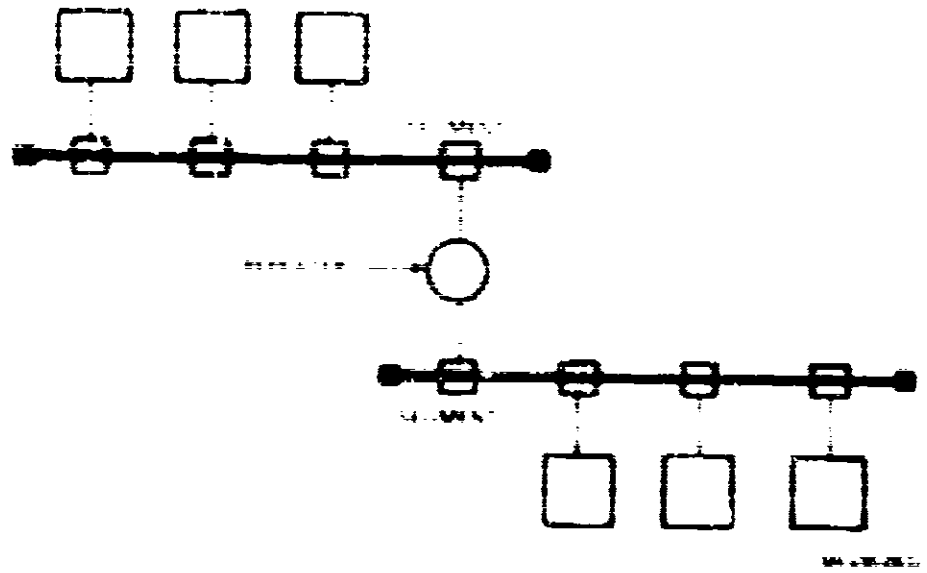


Figure 1-5. Extended Ethernet Configuration

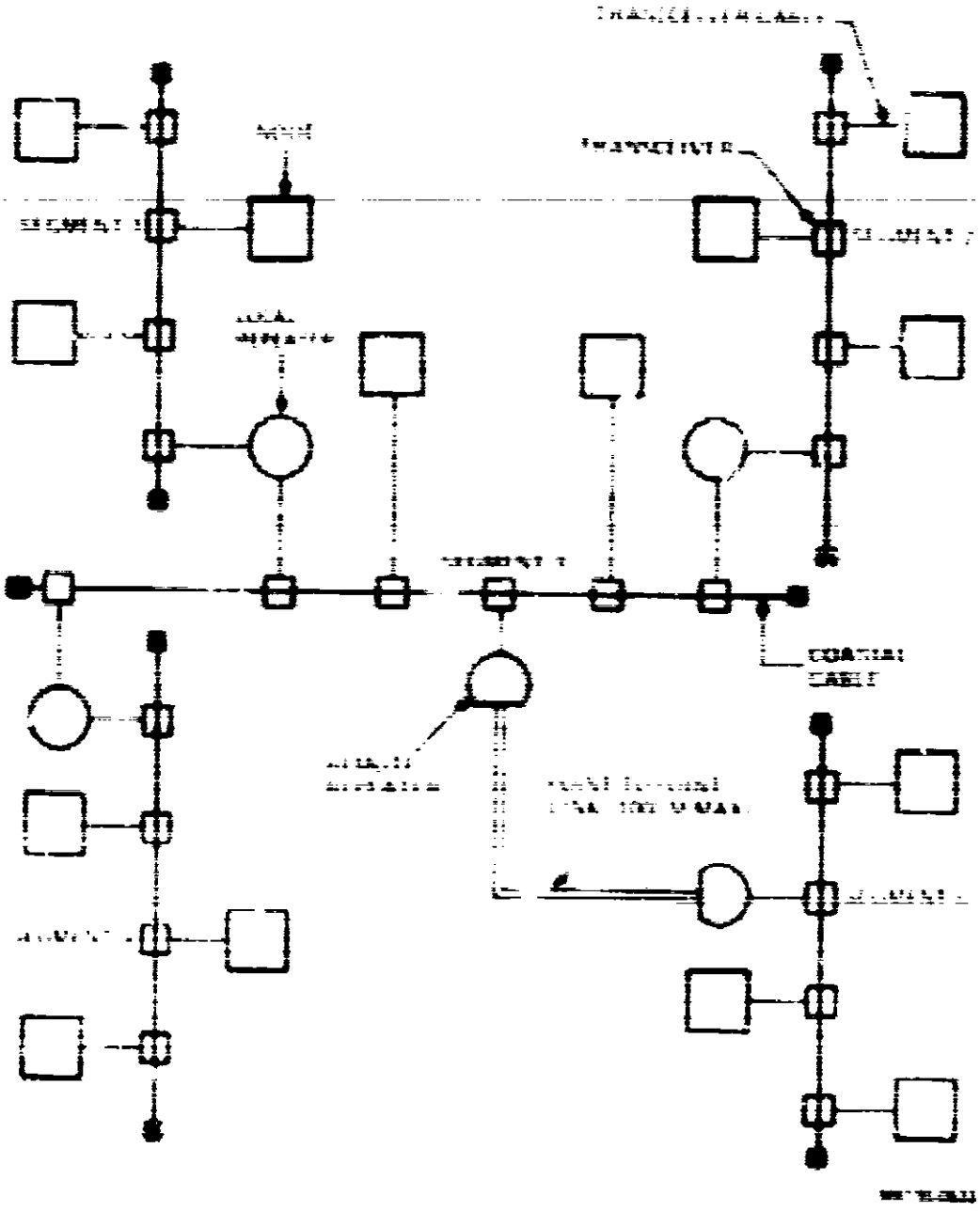


Figure 1-6 Typical Large-Scale Ethernet Configuration

1.3.4 Communication Methods

The network method used in Ethernet systems is called branching bus topology and uses coaxial cable as the transmission medium.

The signaling method is Manchester-encoded digital baseband at a data rate of 10 MHz. The Manchester technique uses positive transitions in the center of the bit-cell to represent ones and negative transitions to represent zeros.

Each station provides a preamble at the beginning of its transmission on which the receiving stations synchronize and derive their internal timing. This reduces problems with line skew or variations in carrier frequency. During the data transmission, a station calculates a cyclic redundancy check (CRC) value, which it then appends to the end of the transmission.

1.3.5 Network Access and Communication Protocol

The server supplies power to the Ethernet transceiver on one twisted pair of the transceiver cable. Communication between the server and transceiver takes place on the following three signal pairs:

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

The server operates in a half-duplex mode and must function in either the transmit or receive mode at any instant in time. When the server is not transmitting, the transceiver monitors the coaxial cable for signal presence and reports the initiation of a transmission. The server reads the Ethernet address of the intended receiver and decides whether or not to accept the transmission.

- Collision pair – The transceiver reports a collision condition with another station by sending a 10 MHz carrier to the server on the collision pair. It also provides a test for the server's collision detection circuits by sending back a 10 MHz burst of about 2μs duration after every transmission. This is called the *heartbeat* check.

1.3.5.1 **Network Access** – There is no priority arbitration for access to an Ethernet system and all nodes and stations have equal access. Access is accomplished using a line contention/collision detection protocol called carrier sense multiple with collision detection (CSMA/CD).

1.3.5.2 **Collision Detection** – When two or more stations transmit at the same time, a collision occurs, causing garbled data. On detecting a collision, each transmitting station continues to transmit for a predetermined length of time. This enforces the collision to ensure its detection by all stations on the network.

- [illegible]

- [illegible]

- [illegible]

1.3.6.3 Data Frame Format – Each data transmits a frame of serial data in the format shown in Figure 1-7. Each byte of data is right shifted within the format. Table 1-10 describes the four fields that make up a data frame.

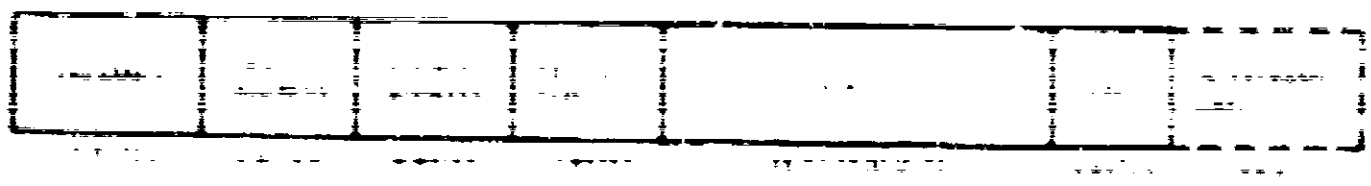


Figure 1-7 Ethernet Data Frame Format

Table 1-10 Ethernet Data Frame Fields

Field	Name	Description
1	Preamble	A 64 bit (8 byte) pattern of alternating 1s and 0s which allows receiver synchronization. The last two bits are consecutive 1s indicating that all following information is to be interpreted as data.
2	Destination Address	A 48 bit (6 byte) field which specifies the destination station. Each station has a unique 6 byte address and examines this field to determine if a received packet is to be passed to the data link layer. If the high order bit of the first byte is a 0, the destination address is unique to a particular station. If it is a 1, a logical group of stations can be specified as 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	This 48 bit (6 byte) address identifies the transmitting station.
4	Protocol Type	This 16 bit (2 byte) value identifies the higher level protocol associated with the frame. When Ethernet is used, higher level protocols can be mixed in any combination within the data field of a single type.
5	Data Field	This field contains a block of data which is passed by the sender station's data link layer to the receiver station's data link layer. The length of the data field is determined by the transmitter station.
6	Pad Field	A variable length field which is used to ensure that the data frame is long enough for proper collision detection.
7	Frame Check Sequence	A 32 bit cyclic redundancy check value used by the receiving station to detect if the frame was damaged in transit.

1.3.5.4 **Ethernet Characteristics** – Table 1-11 summarizes the main characteristics of an Ethernet system.

Table 1-11 Summary of Ethernet Characteristics

Characteristic	Value or Definition
Topology	Branching bus
Transmission Medium	Coaxial cable using Manchester encoded digital baseband signaling
Data Rate	10 million bits per second (10 Mbps)
Maximum Separation of Stations	2.8 kilometers (1.74 miles)
Maximum Number of Stations	1024 stations
Network Control	Multiple access with carrier sense and collision detection
Access Control	Carrier sense multiple access with collision detection (CSMA/CD)
Data Frame Length	Variable, but limited to a maximum of 1526 octets between frame delimiters

CHAPTER 2

2
Self-Test Program

2.1 General

The self-test program resides in terminal server program read-only memory (ROM). This chapter summarizes the self-test diagnostic modes and test sequences.

The program has four modes – three start-up modes under which it performs diagnostic hardware tests and one that reports fatal errors.

1. Power-up mode – Entered when power is applied, this mode resets and initializes the server logic.
2. Initialized mode – Entered only from the LAT operating software on an operational server.
3. Manufacturing mode – Entered only when the server is connected to a test station. All manufacturing mode failures are fatal errors.
4. Error mode – Entered when the self-test detects a fatal error.

The self-test can also be run under on-line debugging tool (ODT) procedures.

The initialized mode can only be entered from the LAT operating software. It allows the server manager to perform diagnostic hardware checks and keep error counts as described in the *DECserver Terminal Server Operations Guide*.

2.2 Self-Test Modes

2.2.1 Power-Up Mode

When power is applied, this mode resets and initializes the terminal server logic. If the reset-to-factory-settings switch procedure is performed, it also restores the permanent server parameters to the nonvolatile erasable programmable read only memory (EEPROM).

When power is applied, the JAM instruction of the addressing logic is set, supposing the selection of program ROM and program RAM memory space. When the power has stabilized, the CPU starts up by fetching the initial stack pointer from location 0 in program ROM and its initial program counter from location 2.

2.2.1.1 Power-Up Flag – The memory test module checks the power-up flag and, if set, tests all 128K bytes of program RAM and clears the upper 128 bytes before calling the initialize program. If the power-up flag is not set, the upper 128 bytes of program RAM are neither tested nor cleared. The memory locations in that area containing the initialization parameters remain as before.

2.2.1.2 Reset Flag – If the reset flag is present in the reset-to-factory-settings switch, the power-up routine writes the factory default server parameters from program ROM to EEPROM and clears the reset flag.

2.2.2 Initialized Mode

This mode is entered from the power-up mode when the self-test is complete, the EEPROM contents is the parameter area, and the reset flag is not set. In this mode, normal server operations can be used or changed as required.

Starting at the first test executed in the power-up mode, it repeats the self-test, memory map, and checksum of the upper 128 bytes of program RAM. It also checksums the parameter area of EEPROM.

Table 2-1 Initialized Mode Parameter Byte Bit Usage

Bits	Mode Name	Test Description
[illegible]	[illegible]	When set, enables destructive testing of program RAM
[illegible]	[illegible]	When cleared, disables testing of all eight terminal I ports When set, enables testing of the terminal selected by bits [illegible]
[illegible]	[illegible]	Select one of eight terminal ports for testing when bit [illegible] is set
<reserved>	LOOP MODE n	Select one of a combination of the following four test modes
		[illegible] Test Mode Selection
		[illegible] Loop Self-Test - Enables the self-test loopback path when the EEPROM read/write test
		[illegible] External Loopback Terminal - Enables external loopback tests of the terminal ports
		[illegible] Enable NI External Loopback - Enables external loopback testing of the LANCE and NI when used in the External Loopback mode

All program RAM memory addresses are tested except for the upper [illegible] bytes and the [illegible] addresses 1FFF through 1FFF. The parameters stored in that area can be saved or changed [illegible]

2.2.3 Manufacturing Mode

The unit test checks whether the manufacturing mode flag is asserted or not. If the flag is not asserted, the following changes are made in the test modes:

- The unit test program keeps continually toggling the WATCHDOG output.

  Extensive dynamic memory tests of program RAM are made.

- The Ethernet port and all terminal ports are tested in the external loopback mode.

- The read/write tests of EEPROM are disabled because of its limited life (about 10,000 write cycles per byte).

- All errors are considered fatal, halting the test.

- The unit test can only be left by resetting the microprocessor or the jumper and restarting the program.

2.2.4 Error Mode

The unit test calls the error mode when it detects a fatal error. (All errors are considered fatal in the manufacturing mode.)

The error routine writes the error code to a location in EEPROM if the location of the error code has been requested. It then stops the watchdog timer and halts.

In the manufacturing mode, all tests except the EEPROM tests are disabled until the error code in EEPROM is cleared. This is the only special procedure.

Some hardware failures are considered to be either fatal (hard) errors or marginal (soft) errors depending on their effect on the server's operation.

- Soft errors. Soft errors are failures that may or may not interfere with normal operation or cause the server to operate at less than full capacity.

  The unit test program sets the front panel LED indicator to blink and generates a status packet addressed to each host that identifies the error. The server is then allowed to continue operation.

- Hard errors. Hard errors are failures that disable the server or cause unacceptable or unpredictable operation.

  The front panel LED is set to ON and the error code is stored in nonvolatile memory. All manufacturing tests failures are hard errors.

2.2.4.1 Nonfatal (Soft) Errors - Soft errors consist of the following types of failures:

- [illegible]
- [illegible]
- [illegible]

Field replacement of the server is normally not necessary, but may be desirable. The server attempts to enter a normal operating state and signals that soft errors were encountered by blinking the back panel LED indicator.

Table 2-2 describes the soft error types. Table 2-3 describes the status parameter longword 2.14 bytes that the self-test pushes onto the stack before issuing a downline load request.

Table 2-2 Nonfatal (Soft) Error Types

Error	Type Description
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	When the server is operating on the external loopback mode, the self-test [illegible] external loopback on the LANCE. The external loopback mode [illegible] entering the external loopback mode.
[illegible]	If any terminal ports [illegible] or packet error, the self-test [illegible] this nonfatal error. The server [illegible] all the eight ports are [illegible] the server is not [illegible] from a [illegible] condition.

Table 2-3. Error Status Parameter Longword

Error Bits	Description
High Word	
<31:16>	Not used
<15>	NI heartbeat error
<14>	NI external loopback error
<13>	EEPROM checksum error in the DECLANCE memory area
<12>	EEPROM checksum error in the server parameter area
<11:08>	EEPROM checksum error in the parameter areas for terminal ports 8 - 1
Low Word	
<07:04>	Not used
<03:00>	Terminal error in ports 4 - 1

2.2.4.2 Fatal (Hard) Errors – Hard errors consist of the following types of failures:

- Program RAM data error
- Program ROM CRC error
- Timer error
- JAM error
- LANCE error
- Terminal port error (when all ports are not usable)

Table 2-4 describes the hard error types and Table 2-5 describes the hard error codes that are written to EEPROM. For the dual asynchronous receiver transmitter (DART) tests under number 6, bit <05> of the error code is set and is ORed with the bits identifying the failing port. Thus, 25 (hex) as an example, indicates that all five tests have failed.

Table 2-4. Fatal (Hard) Error Types

Error Type	Description
Program RAM Data Error	A program RAM data write detected in the [illegible] test
Program ROM CRC Error	Any error detected on a CRC-16 calculation of program ROM.
Timer Error	Any failure detected in the refresh or watchdog timer test.
IAM Error	The test failed to [illegible] from program ROM and continue from the [illegible] RAM. Testing cannot be completed from program RAM.
[illegible] Error	Any error detected during initialization of the on chip UART [illegible] test.
Terminal Port Error	If one of the eight terminal ports is unable [illegible]

Table 2-5. Fatal (Hard) Error Codes Written to STATCOM

Error Code	Logic Test
Module 1	
[illegible]	[illegible] program RAM read/write data test
[illegible]	[illegible] program ROM CRC test
[illegible]	[illegible] EA EPROM checksum test
[illegible]	[illegible] EEPROM checksum test
[illegible]	[illegible] program RAM dynamic read/write data test
[illegible]	Refresh timer test
[illegible]	Scan timer test
[illegible]	EEPROM read/write data test
Module 2	
[illegible]	RXRDY character length parity test
[illegible]	Transmit/receive break test
[illegible]	Force overrun/FIFO digital test
[illegible]	Force framing error test
[illegible]	Baud band rate test

Table 2-5: Fatal (Hard) Error Codes Written to EEPROM (Cont)

Error Code	Logic Test
Module C	
50	LANCE internal loopback test with multiple data frames
51	LANCE broadcast address test
52	Transmit CRC logic test
53	Receive CRC logic test (good CRC)
54	Receive CRC logic test (bad CRC)
55	Collision detection and retry test
56	Accept multicast address test
57	Reject multicast address test
58	Reject physical address test
59	External loopback test
5B	Network interconnect (NI) heartbeat test (soft error)
Module D	
41	Ethernet terminal port test failure
42	Ethernet LANCE test failure
43	Ethernet terminal port and LANCE test failure

Table 2-6 lists the byte address allocation for EEPROM. The circuit timer and keep-alive timer values, and the server number, type, name and location are referenced and maintained by the LAT software. They are not related to, and should not be confused with, server hardware constants such as the watchdog timer and the refresh timer.

Table 2-6: EEPROM Address Space Allocation

Byte Addresses	Usage
000	Checksum value
001	Keep-alive timer
002 – 003	Server number
004 – 005	Server type
006	Server name length
007 – 016	Server name
017	Location length
018 – 027	Location
028	Privileged password length
029 – 02E	Privileged password
02F	Login password length
030 – 037	Login password
038	Software ID length
039 – 03F	Software ID
040	Console operator
041	Server characteristics
042	Login time
043	Spare
044	Checksum
045	Hardware options
046	Diagnostic test byte
047 – 049	Spare
04A	NVR password level
04B	Self-test password level
04C	Spare
04D	SCO password level
04E	LANCE password level
04F	Checksum
050 – 06F	Terminal 1 parameters
070 – 08F	Terminal 2 parameters
090 – 0AF	Terminal 3 parameters
0B0 – 0CF	Terminal 4 parameters
0D0 – 0EF	Terminal 5 parameters
0F0 – 10F	Terminal 6 parameters
110 – 12F	Terminal 7 parameters
130 – 14F	Terminal 8 parameters

2.3 Diagnostic Modules

The self-test provides four main routines that run the following diagnostic tests on the terminal server logic:

- Module A – Server memory and timer tests
- Module B – DART transmit/receive tests
- Module C – Network interconnect (NI) transmit/receive tests
- Module D – Hardware exerciser

The self-test performs a step-by-step probe of the server hardware with each test module being called and performed in sequence. Module A is entered following a power-up sequence or when the self-test is entered in the initialized mode. On a "Successful Pass" indicator, each test proceeds to the next one unless a fatal error is detected.

2.3.1 Module A – Memory and Timer Tests

Module A executes from program ROM and performs the sequence of tests shown in Table 2-7.

Table 2-7. Module A – Memory and Timer Tests

Test Number	Test Name	Test Function
1	Program RAM Base Read Write Test	Tests all program RAM memory locations. The upper 128 bytes are not tested unless the power-up flag is set.
2	Program ROM CRC Test	Performs CRC-16 calculations on all program ROM locations including the stored CRC value.
3	IA PROM Checksum Test	Performs a 16-bit checksum calculation on all Ethernet physical address PROM (IA PROM) locations according to the Ethernet specification.
4	EEPROM Checksum Test	Performs a 16-bit checksum calculation on all EEPROM locations.

Table 2-7: Module A – Memory and Timer Tests (Cont)

Test Number	Test Name	Test Function
6	Program RAM Dynamic Read/ Write Test	Performs extended dynamic tests of program RAM using a modified moving inversion algorithm. This test is performed only when the long memory test (LONG) mode is set in the initialized or manufacturing modes
6	Refresh Timer Test (IPL7)	Tests refresh timer interrupts at IPL level 7.
7	Watchdog Timer Test (IPL4)	Sets the CPU to IPL7 and loads the counter with a 10 ms value. It then decrements the IPL of the CPU and checks that the timer interrupts with the correct vector address when the CPU level is at IPL3.
8	EEPROM Read/ Write Test	Verifies that the EEPROM can be written and read. This test is not enabled to exit the manufacturing or self-test loop modes.

2.3.2 Module B – DART Transmit/Receive Tests

An image of module B is written to program RAM from where it performs the sequence of tests shown in Table 2-8.

Table 2-8: Module B – DART Transmit/Receive Tests

Test Number	Test Name	Test Function
1	RXRDY, Character Length, and Parity Test	Transmits characters of varying length and parity. Services the activity under receive interrupts and checks that the received data is zero-extended to the most significant bit for the different character lengths.
2	Start/Stop Break Command Transmit and Receive Test	Issues the START BREAK and STOP BREAK commands, checking that the change break status bit sets and clears and that the first FIFO location is loaded with the break character
3	FIFO Depth and Force Overrun Error Test	Transmits data sequences that fill the FIFO buffer and tests for interrupts on the FIFO full and overrun error flag
	Force Framing Error Test	Selects different transmit and receive speeds (external loop mode only). Tests the framing error flag and for any lost stop bits.
5	Basic Transmit, Receive and Baud Rate Select Test	Transmits characters at varying baud rates and checks that they are received within timed intervals

2.3.3 Module C – Network Interconnect (NI) Transmit/Receive Tests

An image of module C is written to program RAM from where it performs the sequence of LANCE/SIA tests shown in Table 2-9. Tests 1 through 6 are internal loopback tests and any failure is a fatal error.

Table 2-9: Module C – Network Interconnect (NI) Transmit/Receive Tests

Test Number	Test Name	Test Function
1	Internal Loopback Test	Transmits internal data frame loops using the LANCE physical Ethernet address as both the destination and source address.
2	CRC Logic Test	This test performs the following three subtests:
		A Transmit CRC Logic Test – Transmits a data frame with the transmit CRC enabled. Compares the received CRC with a pre-calculated value.
		B Receive CRC Logic Test – Transmits a data frame with a pre-calculated CRC value and with the transmit CRC disabled. Checks that the receive CRC logic detects the error.
		C Receive Bad CRC Test – Transmits a data frame with a bad CRC value and with the transmit CRC disabled. Checks that the receive CRC logic detects the error.
3	Collision Test	Transmits a data frame with the collision detected bit set in the LANCE mode register. The LANCE should set the MERR error bit after the 16th detection and transmission retry failure.
4	Multicast Address Test	Tests the ability of the LANCE to accept or reject data frames with the multicast bit set in the destination address.
5	Physical Address Test	Transmits a data frame with a destination address not equal to the LANCE. Checks on the transmit status and unexpected interrupts.
6	Data Chain, Byte Swap, Broadcast Address Test	Transmits a data frame to the broadcast address with the byte swap bit set. The receiver should accept the data frame and store it in two receive buffers.
7	External Loopback Test	Transmits loopback data frames to verify the path from the LANCE through the SIA and the Ethernet transceiver. The test is normally enabled but may be disabled in the unattended mode.

2.3.4 Module D – Hardware Exercise

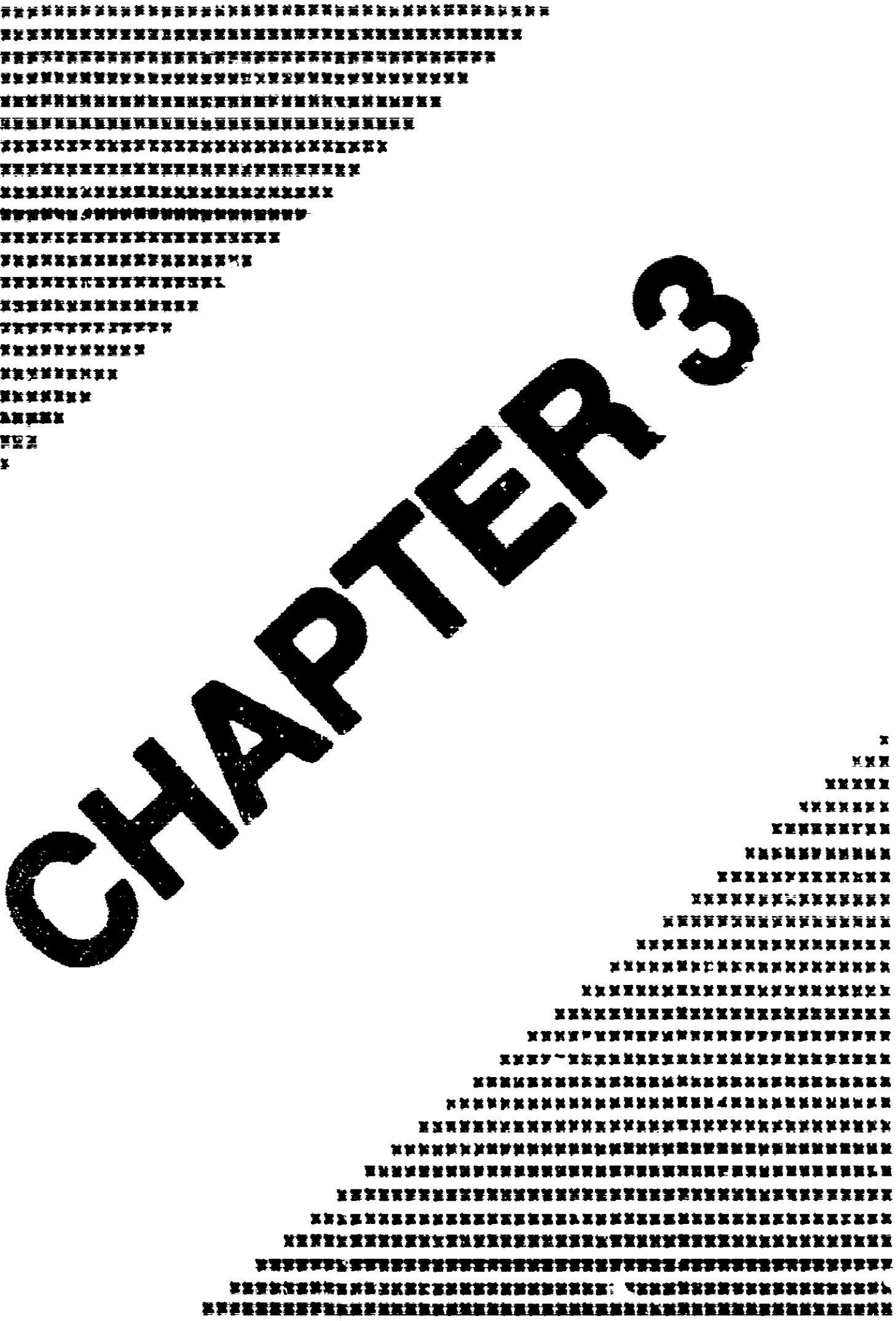
An image of module D is written to program RAM from where it performs the types of hardware exercises shown in Table 2-10.

Table 2-10: Module D – Hardware Exercise

Test Number	Test Name	Test Function
1	Maximum Activity Test	[illegible]

[illegible]

CHAPTER 3



3
Initialize Program

3.1 General

The initialize program resides in the program ROM and is entered on completion of the self-test program or on a fatal bugcheck error.

Following the self-test the initialize program transmits a request for a dumping load of the LAT operating software over the Ethernet loopback test failed. For a DECserver 100 to be placed into operation, at least one active DECnet Phase IV host is required on the network.

If contained in the server memory, the initialize program loads a request for an upline dump of the server memory into program RAM on a fatal bugcheck if not caused by the watchdog timer.

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

Part 1 — Transmits a request program load message on the Ethernet for a dumping load of the server software.

Part 2 — Processes the down-line load records from the host host, stores the server files in program RAM and displays the load status on the console terminal.

Part 3 — Displays the "Fatal bugcheck" message on the console terminal if a fatal error condition is detected by the LAT operating software.

Part 4 — If enabled in the server software, initiates the upline dump of all server program RAM to a host dump host after a fatal bugcheck.

3.2 Downline Load

The initiator program displays a series of messages on the console terminal that report the progress of the downline load procedure.

3.2.1 Downline Load Messages

If the self-test did not detect any errors, it turns on the LED indicator and exits the initiator program. If any nonfatal (soft) hardware errors were detected, however, it sets the LED indicator to blink and exits the initiator program which displays the soft error WARNING message (described in Section 3.1).

When the load procedure begins, the first of the following messages is displayed on the terminal screen. It identifies the Ethernet address and the firmware version number. The second message is displayed by the server following transmission of the required program load message. The message is repeated every 30 seconds if the load fails or if a load host responder is not immediately available.

```
Local VMS Initializing DECserver 90 00 2B 2C 11 13• MOP V3.0 X-V0.0-A A

Local VMS waiting for load host
```

When a load host responder responds, the load procedure is started and the following message is displayed, identifying the load host Ethernet address.

```
Local VMS loading from host 08 00 2B 11 11 11
```

If the load procedure is interrupted or the downline load message is not received from the host within 30 seconds, the following message is displayed and the load procedure is restarted.

```
Local VMS load failure timeout
```

On successful completion of the load the following message is displayed and the program code just transferred to the server software now is [DDT, if enabled].

```
Local VMS image load complete
```

A user logs into the server by pressing the RETURN key and entering a valid user name at the prompt. The server software responds and returns the Local> prompt as follows.

```
DECserver 90 Terminal Server V1.0 (BL6)

Enter username> MOM <RET>

Local>
```

3.2.2 Downline Load Procedure

The downline load procedure uses the standard maintenance operations protocol (MOP). The procedure between the terminal server and the load host consists of the following steps.

1. On completion of the self-test program with other than a hard error or Ethernet loopback failure, the server transmits a request program load message to the load assistance volunteer multicast address.

2. Host systems supporting the software identification field in the request program load message attempt to locate the file under their system load directory without accessing a DECnet database.

   Host systems that support the downline load, but not the software identification in the file name, search the DECnet database for the server's Ethernet address and attempt to load the file defined in the database entry.

   Host systems that locate the file and volunteer load assistance, send an assistance volunteer message to the requesting server.

3. The server selects the first assistance volunteer it receives and retransmits the request program load message, addressed to the selected assistance volunteer (the load host).

4. The load host initiates the downline load by sending the server a memory load message containing the first record of the server software image file.

5. The server loads the first memory load message to memory then sends another request memory load message to the load host.

6. The load host then sends subsequent memory load messages containing sequential server software records. The server responds to each memory load message by sending another request memory load message.

7. When all records have been loaded to the server, the load host sends the parameter load with transfer address message and terminates the downline load process.

8. The server acknowledges with a boot request the new load message and starts up the acquired server software to become fully operational.

3.3 Uplink Dump

For an uplink dump to take place, the uplink dump parameter must be enabled in the server manager. The initiate program then sends an uplink dump request to the original load host on a data logging net and waits for a response.

During an uplink dump procedure, the initiate program transfers an image of the entire server memory context (including the LAT operating software) using the standard Digital maintenance protocol (MOP). It also displays a series of messages on the console terminal that report the progress of the dump.

3.3.1 Uplink Dump Messages

The following message is displayed when the server has successfully completed an uplink dump.

```
%%%% Waiting for image dump
```

When the dump host responds and the uplink dump procedure is started, the following message is displayed, identifying the dump host Ethernet address.

```
%%%% Dumping to host AA-BB-CC-DD-EE-FF
```

The successful completion of the dump, the following message is displayed and initiate returns to the command program and to DCL, if required.

```
%%%% MOP image dump complete
```

3.3.2 Uplink Dump Procedure

The uplink dump procedure that takes place between the server and the dump host consists of the following steps:

1. To initiate a dump, the initiator program sends a request dump service message to the original load host. If the original load host serves the request, it sends a request terminate dump message to the server and proceeds to step 3, eliminating the next three steps.

2. If the original load host does not respond to the request dump service message in the specified time, the server retransmits the request dump service message to the load assistance volunteer machines registered.

3. Host systems supporting the request dump service message first verify if they support dumps from the requesting server. Dump host load assistance volunteer machines send an available volunteer message to the server.

4. The server selects the first load assistance volunteer that responds and retransmits the request dump service message, addressed to the selected load assistance volunteer dump host.

5. The linked dump host starts the uplink dump sequence by sending a request memory dump message to the server.

6. The server responds to a request memory dump message from the host with a requested memory dump record containing the image data of each section of program RAM. As server memory contents are dumped, initializing instructions and data.

7. The linked dump host receives each memory dump message, stores the image in sequential records on a server dump file and sends another request memory dump message to the server.

8. The process continues until the server receives a dump complete message from the linked dump host, and the dump host terminates the uplink dump process. The server then enters IDLE or enabled for load dump commands or enters the wait for the state.

3.4 Status and Error Messages

Following a downstream load, the server software provides status and error information to the initiator and maintains audit trail records generated by the load host. Dump or load test failures may cause the downstream load procedure to be aborted or invalidated.

3.4.1 Ethernet Loopback Error Message

If an Ethernet loopback test failed under the self-test, a downline load is not attempted and the following message is displayed:

[illegible] Image load not attempted, network communication failure

A power-up or server re-initialization ("INIT") is then required to clear the condition.

The above message indicates that the server cannot access the Ethernet, possibly because of transceiver or cabling problems that should first be corrected. Typing a "INIT" on the console terminal re-initializes the server and prevents another downline load request.

Re-initialization takes place only if the downline load procedure fails. ("INIT" has no effect if the LAT software starts up successfully after a successful downline load. This prevents unauthorized users from re-initializing the server during a normal downline load process that does not affect later requests in the server memory.)

3.4.2 Nonfatal (Soft) Error Message

If the self-test did not detect any errors, it turns on the LED indicator and calls the initialize program.

If any nonfatal (soft) errors were detected, however, it sets the LED indicator to blink and calls the initialize program. The initialize program displays the following message:

[illegible] Nonfatal hardware error detected
Server code xxxx, terminal ports may be unavailable

The nonfatal hardware error message is followed by one of the following messages:

- Server code xxxx

This four-bit code indicates the nonfatal error type. Reading from left to right, each x may be a 0 or 1 with a 1 indicating the following problem:

- [illegible] hardware error
- Ethernet loopback error
- [illegible] checksum error
- Server [illegible] checksum error

Terminal codes nn nn nn nn nn nn nn nn.

Reading from left to right, each two-bit code indicates a nonfatal error condition for a terminal (1 through 8).

A 1 in any pair of bits, reading from left to right, indicates either or both of the following problems with the terminal:

- Terminal port error
- Terminal parameters checksum error

3.4.3 Load Failure or Timeout Error Message

If the load procedure is interrupted or the downline load message is not received from the host within 30 seconds, the following message is displayed and the load procedure restarted.

```
Load 912 load failure, timeout
```

3.4.4 Fatal Exception Error Message

The following message is displayed on a fatal exception.

```
Load 913 fatal exception PCnnn, MCnnn, Mnnn, Innn, CODEnnn
```

The message displays the contents of the CPU program counter (PC), stack pointer (SP), and status register (SR) at the time of the failure.

The MEM field displays the illegal memory address on an addressing error or displays the address of the instruction that may have caused the failure. The CODE field gives the reason for the failure as listed in Table 3-1.

Table 3-1: System 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 Initialization Program (STIM)	
11	NI port hardware memory error
12	NI port [illegible] error
14	NI port transmit buffer error
[illegible]	[illegible]
[illegible]	Unknown error
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
Server Initialization Program (SIM)	
208	LAT software checksum error
211	N' port hardware memory error
212	NI port [illegible] error
214	NI port transmit buffer error
21A	Stack value corrupted in idle loop
216	Unknown error
217	Unallocated error

Table 3-1 System Crash Error Codes (Cont)

Code	Description
218	Unable to allocate XCB
219	Unmasked completion error
220	Local output completion error
221	**REPROM write block error**
222	Query on output queue with no circuit
223	Transition too long
224	Cannot find status
225	No available virtual control block
226	Link pool allocation error
227	Illegal local output state
228	Service defined with no nodes
229	Duplicate node service name found

3.4.5 Timeout, Abort Dump Messages

When a timeout occurs during an upline dump, the following error message is displayed and the dump aborted. Program control then transfers to the self-test routine (DIAT, if enabled):

Local 914 Timeout, dump aborted

3.4.6 Load or Dump Failure Message

The following error message is displayed when a downline load or upline dump procedure fails to successfully complete a load or dump message after ten attempts:

Local 915 Transmission failure after ten attempts

For a downline load, the procedure is restarted. For an upline dump, the procedure is aborted and program control transfers to the self-test routine (DIAT, if enabled).

3.4.7 Bad Image File Message

The following error message is displayed on these conditions:

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

Load 916 Invalid image file, load aborted

CHAPTER 4

4
On-Line Debugging Tool (ODT)

4.1 General

The on-line debugging tool (ODT) is a specialized console program that resides in the program ROM. It can be entered only at the console terminal from the privileged mode of the LAT operating software or from the self-test manufacturing mode. If enabled in the server manager, it may also be entered following a download load or fatal bugcheck.

ODT provides the local input language for server managers or service personnel and takes the place of most of the keys and switches normally present on a CPU front panel. It is intended for use by diagnostic specialists under ongoing reliability test (ORT) procedures and by authorized service personnel or network managers in isolating hardware or network problems.

4.1.1 ODT Requirements

The following requirements must be considered before attempting to enter 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).

- When entered from the LAT software, ODT uses the existing characteristics for the console terminal.

4.1.2 Entering ODT

ODT may be called by one of several conditions or procedures and, when invoked, returns the asterisk (*) prompt.

4.1.2.1 Entering ODT from the Self-Test Manufacturing Mode – The manufacturing operator enters ODT by typing a CTRL/SHIFT/S on the console keyboard (holding down the CTRL and SHIFT keys then pressing the S key). The operator can then run, single-step, or loop on the self-test program, set test parameters and loop on errors.

4.1.2.2 Entering ODT from the LAT Operating Software – Before the server manager can enter ODT, it must be enabled by one of the following privileged commands:

- [illegible] – Sets the dynamic parameter, enabling ODT
- [illegible] – Sets the permanent parameter in EEPROM, but does not enable ODT unless the server is re-initialized

While it is enabled, ODT is entered if one of the following conditions occurs:

- When the privileged [illegible] command is given
- On a system crash (if enabled)
- On a server re-initialization (if enabled)
- On the execution of a program-set ODT breakpoint

4.2 ODT Command Functions

ODT commands consist of one to three characters typed after the asterisk (*) prompt and are terminated by a carriage return or a line feed. ODT interprets command inputs and output values as hexadecimal (hex) digits.

4.2.1 Command Input Errors

On a command input error, ODT responds with one of the error message codes listed in Table 4-1.

Table 4-1: ODT Command Error Codes

Error Code	Name	Definition
RE	Range Error	The lower memory dump address is higher than the highest specified address or the specified CPU register address is out of range (0–7).
CE	Command Error	Unrecognized command input.

4.2.2 Command Summary

ODT commands are organized into the following functional groups:

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

Table 4-2 lists the ODT commands according to command groups and briefly describes their functions.

Table 4-2: On-Line Debugging Tool (ODT) Commands

Command	Name	Function
[illegible]		
E	Examine	Opens and displays the contents of word address where E is the address of the word to be examined. The word address is modified to produce an even address.
EB	Examine byte	Opens and displays the contents of a byte address where E is the even word address of the byte to be examined.
<RETURN>	Carriage return	Terminates an ODT command and/or closes a currently open address.
<LINE FEED>	Line feed	Closes the currently open address and opens and displays the next sequential address.
[illegible]		
R0 or R1 or R2 or R3	R0, R1, R2, or R3 register	Opens and displays the contents of R0, R1, R2, or R3. R0 specifies the register number as 0.
R[n]	[illegible] register	Opens and displays the contents of the R[n] register.
[illegible]		
M	Memory dump	Displays the contents of a block of memory addresses where E is the user-supplied address of the first word and F is the last address.
H or HALT	Halt dump	Halts a memory dump and returns control to the ODT prompt (*).
RD	Register dump	Displays the contents of the internal CPU registers.
[illegible]		
B	Set	Sets breakpoints at specific addresses when a program is to be executed. Breakpoints cause program execution to suspend and return control to the current ODT address.

Table 4-2: On-Line Debugging Tool (ODT) Commands (Cont)

Command	Name	Function
SS	Single step enable	Enables the single step mode, allowing the current program to be executed one step at a time.
NS	Single step disable	Disables the single step mode.
Breakpoint Commands		
BR n	Breakpoint set	Sets a breakpoint where n is the breakpoint number (0 to 7) and n specifies the memory address for the breakpoint. Breakpoints 0 - 7 allow a software trap from up to 8 locations. You can set a breakpoint at any memory address in the user address space.
BR n	Breakpoint clear	Clears breakpoint n (0 - 7). If n is not specified, all nine breakpoints are cleared.
BH	Display breakpoints	Displays the nine breakpoint table entries. The first eight entries contain the address of the eight breakpoint addresses and can be changed or accessed using the EXA, DEP, etc.

4.3 Accessing Terminal Server Address Space

The server manager or privileged operator has access to all terminal server memory and device addresses as shown in Table 4-3.

Table 4-3: Terminal Server Address Ranges

Server Memory or Device	Physical Address Range (in hex)
Program RAM	[illegible]
EEPROM	[illegible]
DART 0	[illegible]
DART 1	[illegible]
DART 2	[illegible]
DART 3	[illegible]
LANCE	[illegible]
Program ROM	[illegible]
PA PROM	[illegible]

An ODT command accepts up to 6 hexadecimal (hex) digits in the address specifier. However, the procedures given in Chapter 3 for accessing program ROM, program RAM, and the DART and LANCE external registers must be followed. ODT provides commands that access the CPU internal registers directly because they are not part of the external memory or device address space.

4.3.1 Memory and Device Register Commands

Only one memory or device register location can be opened at a time to display the contents for examination or changes. Opening a location with another location open, closes the currently open location before opening the next one.

4.3.1.1 **Examine (E) Command** – To open a word location, type E followed by a space and the address to be displayed. If an odd address is entered, the low-order bit is masked out and the even word address is opened. Pressing the RETURN key (<RET>) terminates the command.

For example:

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

If the command location 1000 is open and no change is desired, pressing RETURN closes the location. DDT returns the asterisk prompt and another command may be issued.

If the contents of the currently open word location are to be changed, type the new contents before pressing RETURN to close the location.

For example:

```
*E 1000<RET>          - Entry
001000 = 0123 4710<RET>   - Response and change
```

Pressing the LINE FEED key (<LF>) instead of RETURN closes the currently open word location and opens the next sequential word location.

For example:

```
*E 1000<RET>          - Entry
001000 = 0123<LF>     - Response
001002 = 4567         - Response
```

4.3.1.2 **Examine Byte (EB) Command** – To open a byte location, type EB followed by a space and the address to be displayed. Pressing the RETURN key (<RET>) executes the command.

For example:

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

or

```
*EB 1001<LF>          - Entry
001001 = 01           - Response
```

If the command location 1000 is open and no change is desired, pressing RETURN closes the location. ODT returns the asterisk prompt and another command may be issued.

If the contents of the currently open byte location are to be changed, type the new contents before pressing RETURN to close the location.

For example:

```
*B 1000<RET>          Entry
001000 = 21 54<RET>   Response and change
```

Pressing the LINE FEED key (<LF>) instead of RETURN closes the currently open byte location and opens and displays the next sequential byte location.

For example:

```
*B 1000<RET>          Entry
001000 = 21<LF>       Response
001001 = 21           Response
```

4.3.2 CPU Register Commands

4.3.2.1 CPU Address Register (.An) Command – One of the CPU address registers, An, is opened and displayed using the following command format:

```
*.An<RET>             Where n is the address register number (0-7)
```

For example:

```
*.A5<RET>             Entry
A5 = 00C5FF4         Response (contents of address register A5)
```

4.3.2.2 CPU Data Register (SDn) Command – One of the CPU data registers, D0-D7, is opened and displayed using the following command format:

*Dn<RET>	- Where *n* is the data register number (0-7)

For example:

*D0<RET>	- Entry
D0 = 00000234	- Response (contents of data register D0)

When these registers are opened, the contents are available for examination. The register contents may be changed by typing in the new value before pressing LINE FEED or RETURN as described for the Examine (E) command.

If the input value for *n* in the command is not in the specified range (0 – 7), the range error (RE) message is displayed, followed by the asterisk prompt.

4.3.2.3 CPU Status Register (SSR) Command – The CPU status register is opened and displayed using the following command format:

*SR<RET>	Entry
SR = 2701	Response

The procedure for changing the status register is the same as for the address and data registers.

4.3.3 Dump Commands

The Dump commands are useful during troubleshooting to display program and hardware status.

4.3.3.1 Memory Dump (D) Command – The contents of a block of terminal server memory and DART register addresses may be dumped using the D command. Although LANCE address space can be included in the dump, LANCE registers cannot be accessed with the D command and are always read as zeros.

A memory dump is started by typing the D command, followed by a space and the low address, followed by a space and the high address, as given in the following format:

*D *n k*<RET>	- Where *n* is the address of the first location and *k* is the address of the last location

The specified range of memory addresses are then dumped in word format as shown in the following example.

```
*D 1000 1020[<RET>                    - Entry

001000  nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn aannnnnnnnnnnnnn
001010  nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn aannnnnnnnnnnnnn
001020  nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn aannnnnnnnnnnnnn
```

Here, *n* represents the hex value for each nibble (4 bits) of the dumped memory locations. Each *a* represents the ASCII code equivalent of each pair of bytes (8 bits). 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 of memory are dumped per line. If an odd word address is entered, the low-order bit is masked out, producing an even address. If the low address is entered greater than the high address, the range error (RE) message is displayed and the prompt for another command input is given.

4.3.3.2 Halt Dump (CTRL/C) Command – Typing a (CTRL/C) (pressing and holding the CTRL key then pressing the C key) halts the memory dump and ODT returns to the asterisk (*) prompt. This is most useful if errors are made while typing in the address range.

4.3.3.3 Register Dump (RD) Command – Typing RD dumps 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)

The register contents are displayed in the following format:

*R<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				

Here, n represents a hexadecimal digit. Register A7 is always the supervisor stack pointer because the self-test and LAT programs execute in the supervisor mode. The above registers are not opened for changes. The CPU register commands must be used to change their contents.

4.4 Running Programs Under ODT

This is a description of the ODT command functions that may be used when debugging under the self-test program or the LAT operating software.

4.4.1 Program Control Commands

4.4.1.1 Go (G) Command – The Go (G) command starts the program or continues its operation after a breakpoint or program halt. It is given in the following format:

*G<RET> – Where r is the location to begin program execution. If the r is omitted, execution begins at the address specified by the current PC. If an odd address is entered as the new PC, the low-order bit is masked out to produce an even address.

4.4.1.2 Single Step Enable (SS) Command – When enabled, the single step mode allows each instruction to be executed in sequence. The command is given in 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 may then be used to start or continue program execution.

For example:

*SS<RET>	- Entry (single step enable command)
*G 1000<RET>	- Entry (The Go command starts the program at location 1000)
SS 001002	- Response. (Go executes one instruction, then displays SS, followed by the updated PC value.)
*G<RET>	- Entry (Go continues, executing the next instruction.)
SS 001004	- Response
*	- Prompt for the next command)

4.4.1.3 Single Step Disable (NSS) Command – Single step is disabled by giving the following command:

*NS<RET>	- Entry. (QUIT leaves the single step mode.)

4.4.2 Breakpoint Commands

ODT maintains a breakpoint table that may be displayed but not opened for changes. Any changes to the breakpoint table must be made using the defined breakpoint commands.

Up to nine breakpoints (0 to 8) can be set. The breakpoints do not allow the software to trap and suspend operation from up to eight locations. A trap from breakpoint 8 causes a temporary suspension of the program while ODT generates a synchronizing signal to the emulation coprocessor on the terminal server module.

A breakpoint can be set in any location fetched from program RAM as an instruction but not as data. When a breakpoint is set, ODT replaces the content of the breakpoint location with a trap instruction that suspends program operation and returns to ODT.

4.4.2.1 Breakpoint Set (Bn) Command – A breakpoint may be set or changed using the following command:

*Bn r	- Where n is the breakpoint number and r is the breakpoint location

For example:

[illegible] – Breakpoint entry. (Program halts operation when it tries to fetch location 1000.)

4.4.2.2 Breakpoint Clear (BC) Command – A breakpoint may be cleared using the following command:

[illegible] – Where n is the number of the breakpoint to be cleared. If the input value for n is not in range (1..8), DDT returns the number error (#N) message followed by the asterisk prompt.

If n is not specified, all breakpoints are cleared.

4.4.2.3 Display Breakpoints (BB) Command – The BB command displays (but does not open) the first entry in the breakpoint table. The next eight breakpoints can then be displayed in sequence using the line feed (<LF>) key.

For example:

[illegible] – Main
[illegible] – Response and <LF> pressed
[illegible] – Response and <LF> pressed (The ^.BET entry terminates the command.)

4.4.2.4 Breakpoint Message – When the program encounters a breakpoint, DDT suspends execution and displays a message in the following format:

[illegible] – Where n is the breakpoint number and x is the breakpoint address.

For example:

[illegible] – Displays breakpoint message
* – Returns the DDT prompt

In this example, the program halted after reaching breakpoint 1 at location [illegible]. DDT displays the asterisk prompt and waits for a command input.

CHAPTER 5

5
Functional Logic Description

5.1 General

This chapter provides a functional description of the terminal server logic. Descriptions are to the block diagram level and are intended for use with the maintenance print set.

Figure 5-1 illustrates the following four logic sections:

- Host computer logic
  - Central processing unit (CPU)
  - Data address bus
  - Firmware read logic
  - System clock logic
  - Bus arbitrator logic
  - Interrupt control logic
  - Counters and timers
- Terminal server memory
  - Address decode logic
  - 128K bytes of program random access memory (program RAM)
  - 32K bytes of program read-only memory (program ROM)
  - 512 bytes of electrically erasable programmable ROM (EEPROM)
  - 32 bytes of physical address programmable ROM (PA PROM)

- Terminal interface

  Four dual asynchronous receiver-transmitter (DART) chips
  Eight RS-232 (CCITT V.24 compatible) driver-receivers

- Ethernet interface

  Local area network controller for Ethernet (LANCE)
  Serial interface adapter (SIA)

5.2 Microcomputer Logic

This section describes the CPU and data/address bus signals and the supporting CPU logic.

5.2.1 Central Processor Unit (CPU) and Data/Address Bus

Figure 5-2 is a block diagram of the 68000 CPU as it is used in the DECserver 100 configuration.

The 68000 is a fully implemented 16-bit microcomputer that uses very large scale integration (VLSI) technology and provides three modes of addressing for up to 16 megabytes (16M bytes) of operating memory. Its features include a comprehensive instruction set that uses multiple addressing modes and data types.

Table 5-1 describes the CPU and data/address bus signals used in the DECserver 100 configuration.

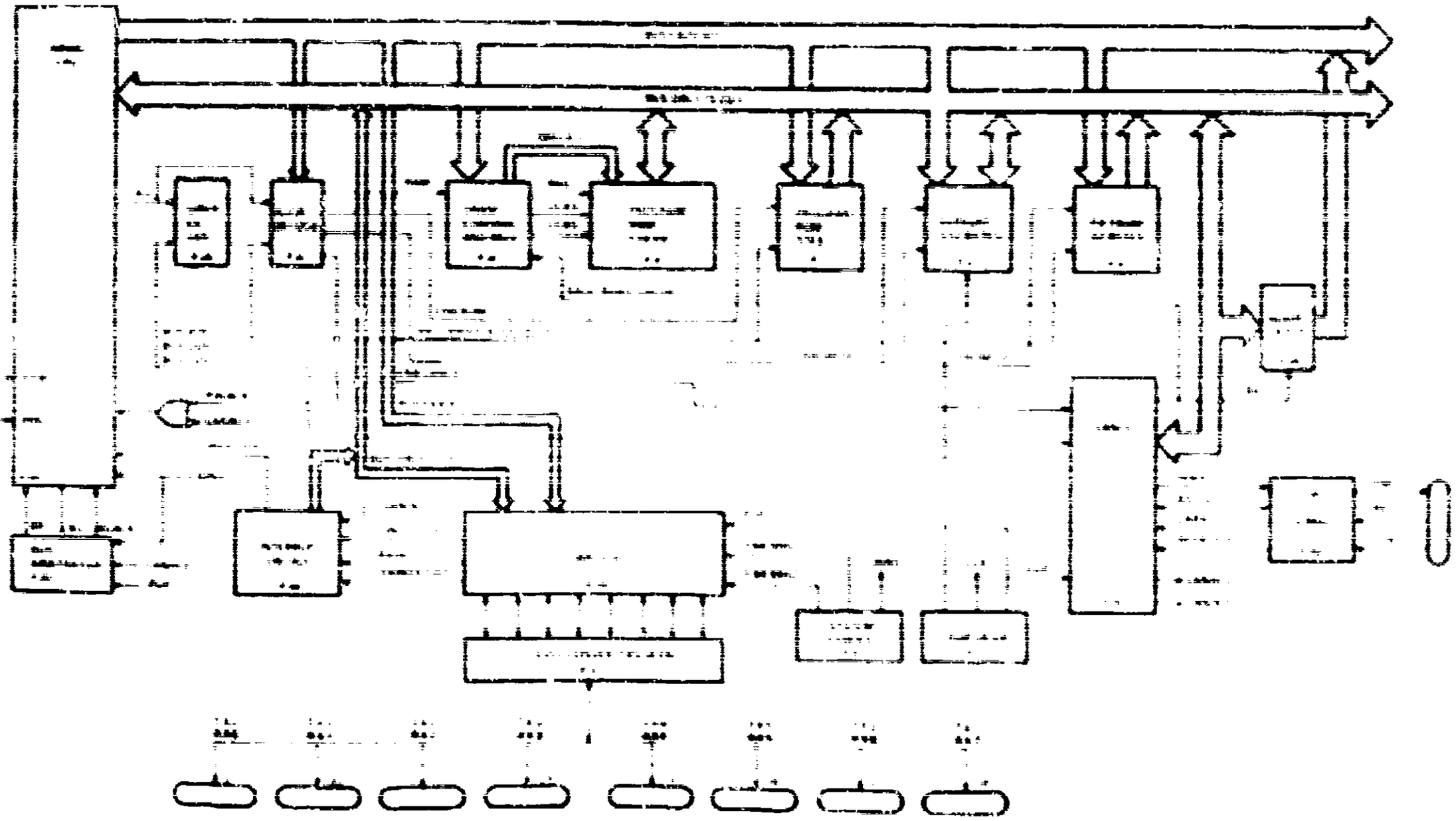


Figure 5-1. DECserver 100 Terminal Server Logic Block Diagram

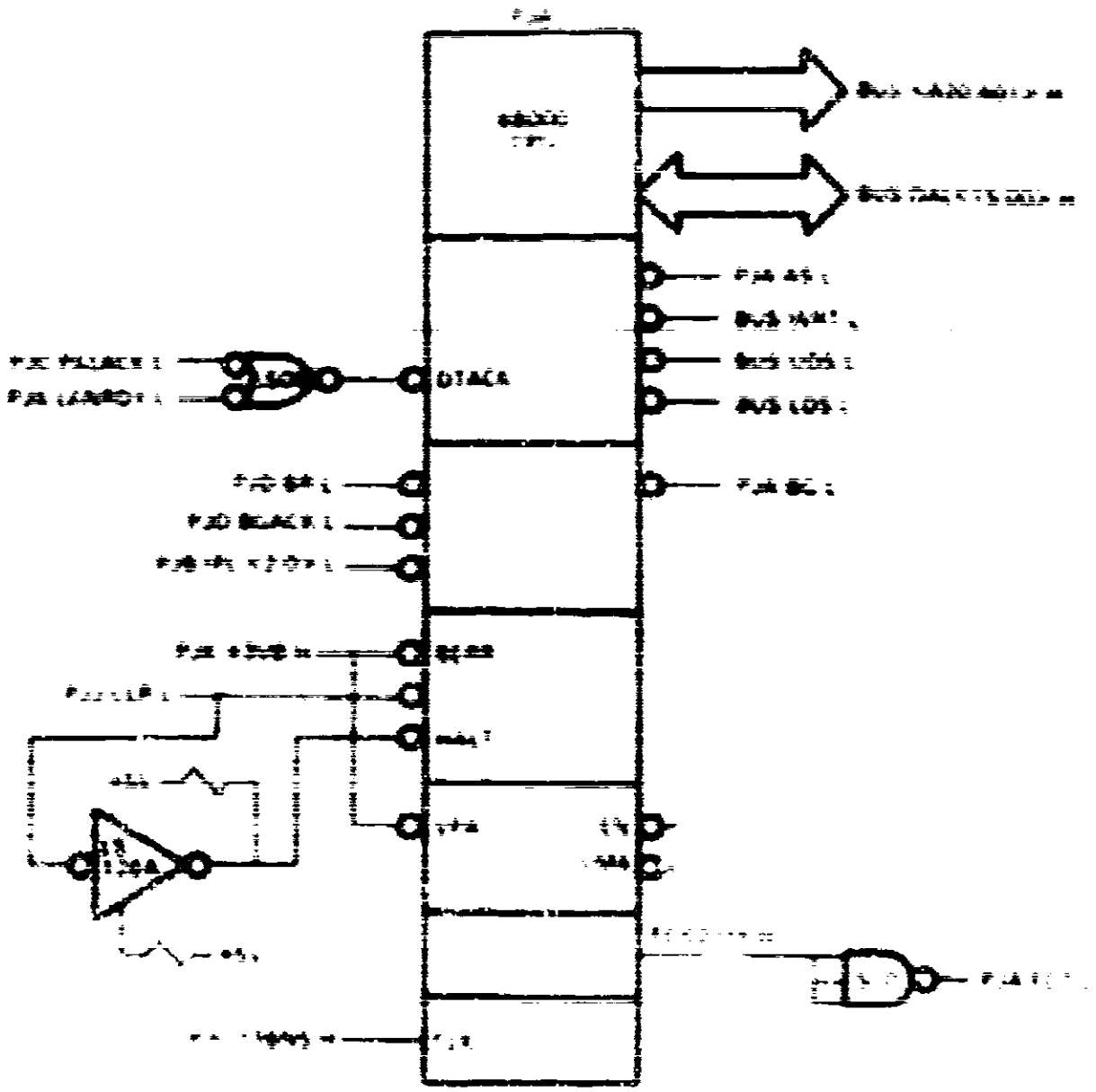


Figure 6-2 CPU Block Diagram

Table 5-1: CPU and Data/Address Bus Signal Description

Signal Name	Description
Data/Address Bus Signals	
Address Bus (BUS A<23:0> or BUS A<31:0>)	The terminal server uses 19 of the 23 twisted address lines to provide direct addressing for up to 2 megabytes (2M bytes) of data (BUS A<18:0>) as shown in Table 5-2. These lines are driven only by the CPU. The server does not use BUS A<23:19>. However, the CPU and LANCE both use the full address bus A<23:0> to generate BUS DAL<15:0> or BUS DAL<31:0> data transfers. Both signals are generated on a word transfer. **NOTE** The CPU uses the complement of A<23:0> for byte transfers (A<23:0> on odd words) BUS DAL<15:0> and on even words BUS DAL<31:16>.
Data Bus (BUS DAL<15:0> or BUS DAL<31:0>)	The 16-bit bidirectional transfers use the same data path for all transfers to and from local and device address space.
Asynchronous Bus Control Signals	
Address Strobe (BUS AS L)	Asserted by the CPU to indicate a register transfer with an external device. The LANCE or a DART indicates that the CPU has a valid address asserted on the address bus.
Write (BUS WRT L)	A write/~read signal, BUS WRT L, is asserted by the CPU or the LANCE to define a bus transfer as follows: Negated (High) = Read Asserted (Low) = Write
Upper Lower Byte Strobes (Byte Strobes) (BUS UDS L, BUS LDS L)	The CPU uses either or both of these signals with BUS WRT L to control bus transfers. The read data latches for a read or write transfer are enabled as follows:

Table 9-1: CPU and Data/Address Bus Signal Description (Cont)

Signal Name	Description				
				Valid Data asserted on:	
	BUS UDS L	BUS LDS L	BUS WRT L	BUS DAL <15:08>	BUS DAL <07:00>
	0	0		No	No
	0	1	0	No	Yes
	1	0	0	Yes	No
	1	1	0	Yes	Yes
	1	1	1	No	No

(0 = asserted high, 1 = asserted low)

As a bus slave, the LANCE ignores the above signals and assumes word transfers. As a slave DMA signals they are driven by the LANCE and the CPU at different times and are buffered in the CPU logic as follows:

Input	Buffered Output
PJA AS L	PJA SAS L, used by the LANCE
PJA WRT L	PJA RWRT H L
PJA UDS L	PJA RUDS H L
PJA LDS L	PJA RLDS H L

Data Transfer Acknowledge (DTACK L)	Asserted by the selected device in response to PJA AS L from the CPU. DTACK L is asserted by the following signal:
	PJC PALACK L from the PAL is asserted during CPU transfers with memory locations or DART registers.
	PJK LANRDY L from the LANCE is asserted during CPU transfers with LANCE registers.

Table 6-1. CPU and Data/Address Bus Signal Description (Cont)

Signal Name	Description
Interrupt Control Signals	
Interrupt Priority Level (PJB IPL<2:0>L)	Asserted by the interrupt priority logic for the highest asserted interrupt request. Any asserted code (other than zero) generates a program interrupt request to the CPU. The CPU contains a 3-bit mask that indicates the current processor IPL and holds an interrupt pending whenever its present level is higher than the requesting device. The CPU grants the interrupt when the current IPL becomes lower than the requesting device. The only exception is IPL 7, which allows an unmasked interrupt for the program RAM refresh timer. During the interrupt cycle, the CPU asserts the IPL of the device being serviced on address bus lines BB [5:3] <A23:A21>. It asserts all other address bus lines high.
Bus Arbitration Signals	
Bus Request (PJD BR L)	Asserted by the LANCE to the bus arbitrator to gain access to the data/address bus and perform a DMA transfer with program RAM.
Bus Grant (PJA BG L)	The CPU responds to a bus request, indicating that the CPU will relinquish control of the bus at the end of the current bus cycle.
Bus Grant Acknowledge (PJD BGACK L)	Asserted by the LANCE to the bus arbitrator in response to bus grant. The LANCE assumes control of the data/address bus when the arbitrator asserts PJD BGACK L and PJD LHLDA L.
Processor Control Signals	
Halt (HALT L)	The CPU asserts this bidirectional line to external devices when it halts. If the line is asserted by an external device, the CPU halts at the end of the current bus cycle. In the DFC server, the halt and clear lines are asserted together to perform a system reset.

Table 5-1: CPU and Data/Address Bus Signal Description (Cont)

Signal Name	Description
Clear (PJJ CLR L)	The CPU asserts this bidirectional line to initialize external devices with the RESET instruction. The CPU could be initialized when the line is asserted externally. PJJ CLR L is asserted on a power up for a minimum of 100 ms. When the signal negates, the CPU starts program execution at memory location zero.
Function Code (FC<2:0> H)	Indicates to external devices the type of reference being executed by the CPU. The CPU has two levels of access privileges: supervisor and user. The reference types are defined as follows (only level 7 is used in the DECserver 100): FC<2:0> Value — Reference Type 0 1 — User Data 2 — User Program 3 4 5 — Supervisor Data 6 — Supervisor Program 7 — Interrupt Acknowledge (PJA H L)
CPU Clock (P H 10% 5 V)	The TTL-compatible input is buffered internally to develop the CPU clocks.

For further information on the MC68000 architecture and instruction set, refer to the following publications:

MC68000 16/32 Microprocessor Programmer's Reference Manual. Prentice-Hall (or Englewood Cliffs, NJ 07632. Fourth Edition: 1992).

The 68000: Principles and Programming (ISBN 0-574-21865-5) by Leo J. Scanlon. Howard W. Sams & Co., Inc., 4300 W. 62nd Street, Indianapolis, IN 46268. 1987.

MC68000 is a trademark of Motorola Incorporated

5.2.2 Power-Up Sequencer Logic

When power is applied, the power-up sequencer asserts the signals shown in Table 5-2 for at least 100 milliseconds after the +5 Vdc power reaches 90 percent of its nominal value.

Table 5-2: Power-Up Sequencer Signal Functions

Signal	Name	Function
[illegible]	PJJCLRL	Asserts a system reset to the CPU
Power-Up	PJJPUPH	Asserts a system reset to the DARTs
Power-Up	PJJPUPL	Asserts a system reset to the LANCE
Write Protect	PJJWRT PROTL	Disables the write inputs to the EEPROM

The power-up sequencer also uses the logical signals shown in Table 5-3.

Table 5-3: Power-Up Sequencer Logical Signals

Signal	Name	Function
Power Fail	PJMP WFR FAIL L	Used during manufacturing tests to generate power-off/power-on cycle resets.
Counter Timer 2	PJIL[illegible]	The main loop timer in the third DART generates a power-up to maintain alarm. If it is not serviced, a reset is the program.

5.2.3 System Clock [illegible]

The primary terminal server oscillator operates at a base frequency of 29.4912 MHz. A frequency divider produces the operating frequencies shown in Table 5-4.

Table 5-4: Operating Frequencies

Frequency	Signal Name	Function
[illegible] MHz	PJC 136NS H	Provides the time base for the CPU, PAL, and bus arbitrator
[illegible] MHz	PJC 1.68MHZ H	Provides the transmit and receive time base for the DARTs.
[illegible] MHz	PJC 1.84MHZ H	Provides the counter/timer clock for the first two DARTs.

5.2.4 Bus Arbitrator

The LANCE is the only device (other than the CPU) that accesses the data/address bus as a bus master. The LANCE initiates a DMA transfer as follows:

1. The LANCE requests access to the bus by asserting buffered LANCE hold (PJK BLANHLD L) to the arbitrator which asserts bus request (PJD BR L) to the CPU.
2. When the CPU returns bus grant (PJA BG L) to the arbitrator, it is synchronized by the CPU clock (PJC 136NS H) which asserts bus grant acknowledge (PJD BGACK L).
3. LANCE hold acknowledge (PJD LHLDA L) is clocked and asserted to the LANCE on the next CPU clock, and the LANCE assumes control of the bus.
4. When the LANCE completes its bus cycle it deasserts PKJ BLANHLD L, deasserting PJD BGACK L and releasing control of the bus.

5.2.5 Interrupt Control

The CPU interprets any level other than zero on its IPL<2:0> inputs as a program interrupt request. A zero input indicates that no interrupts are being requested.

5.2.6.1 Priority Level Encoders and Vector Addresses – The interrupt priority levels (IPLs) are defined by the first of three priority encoders as shown in Table 5-5. IPL7 has the highest priority and IPL2 has the lowest.

Table 5-6: Priority Encoder 1 Inputs

IPL Level	Input Signals to Priority Encoder 1	Interrupt Source
7	PJH CTINT0 L	Refresh Timer
6	PJH RXINT<7:0> L	DART Receivers (from priority encoder 2)
5	PJK LANINT L	LANCE Interrupt
4	PJH CTINT1 L	Slot Timer
2	PJH TXINT<7:0> L	DART Transmitters (from priority encoder 3)

Table 5-6 lists the vector address asserted by each IPL on an interrupt. The terminal transmit and receive lines are prioritized according to the terminal line number from the DART:

- At IPL6, receive interrupt level RXINT7 has the highest priority and RXINT0 has the lowest.
- At IPL2, transmit interrupt level TXINT7 has the highest priority and TXINT0 has the lowest.

Each of the terminal request lines is connected to the encoder through a simple stage synchronizer.

Table 5-6. Interrupt Vector Addresses

Interrupt Source	Level	Name	Vector
Refresh Timer	IPL7	CTINT0	1F0
DART Receivers	IPL6	RXINT7	1D0
		RXINT6	1D4
		RXINT5	1D8
		RXINT4	1DC
		RXINT3	1E0
		RXINT2	1E4
		RXINT1	1E8
		RXINT0	1EC
LANCE	IPL5	LANINT	1B4
Slot Timer	IPL4	CTINT1	190
DART Transmitters	IPL2	TXINT7	150
		TXINT6	154
		TXINT5	158
		TXINT4	15C
		TXINT3	140
		TXINT2	144
		TXINT1	148
		TXINT0	14C

5.2.5.2 CPU Response to an Interrupt Request – The CPU contains a three-bit priority mask register that indicates the current IPL of the CPU. It is used to determine whether the current CPU priority level is higher, equal, or lower than the pending request.

The CPU responds to an interrupt request as follows:

1. The mask register holds an interrupt pending whenever the CPU priority level is higher than or equal to the requesting device.

2. The CPU grants an interrupt when its current IPL becomes lower than the requesting device. The only exception is IPL7 which is not inhibited by the CPU interrupt priority mask. IPL7 allows unmasked interrupts by the Refresh Timer for refresh operations to dynamic program RAM.

3. The first priority encoder asserts the highest requesting IPL to the CPU as an interrupt request.

4. The interrupt is then granted at the end of the current instruction or at the end of a bus cycle with AS, BGIACK, and DTACK negated (a DMA cycle has completed and released the bus).

5.2.5.3 CPU Interrupt Cycle – When entering an interrupt cycle, the CPU first disables tracing and sets the privilege level to that of the requesting interrupt.

The CPU then performs the following actions on the interrupt cycle:

1. Saves the current contents of the status register (SR) and the program counter (PC) on the stack. The saved PC value is the address of the next instruction that would have executed if the interrupt had not occurred.

2. Asserts FUNCTION CODE (interrupt acknowledge).

3. Asserts the level of the acknowledged interrupt on address bus bits A[3:1] (the value held by the priority mask). It asserts all other address bus bits high.

4. Reads the vector address from the interrupt logic on data bus bits D[7:0] (or more). DTACK and BERR are not asserted and remain in the high impedance (HI-Z) state.

5. Multiplies the vector by four (shifts it two places to the left) to obtain the interrupt exception vector address.

5.2.6 Hardware Counters and Timers

The DECserver 100 terminal server makes use of the following counter/timer functions which are asserted from output port 3 (OP3) of each DUART:

- Four DUART – The refresh timer interrupts the CPU to refresh the dynamic program RAM. It is also used as the clock input to the watchdog and LED timers.

  PULSE(TINTR), interrupts the CPU at IPL7 and asserts an interrupt vector of 1F (hex). On each interrupt, the service routine accesses 128 consecutive program RAM locations.

- Second DART - The slot timer is used by the operating software

  PWR_INT1_L interrupts the CPU at IPL4 and asserts an interrupt vector (150 (hex)

- Third DART - The watchdog timer ensures that the CPU does not become hung in a loop. The counter is first loaded during initialization and must be periodically reset by the server software. If it times out for any reason, PWR_INT2_L triggers the power-up sequencer which re-initializes the server and requests a reload of the LAT operating software.

- Fourth DART - The LED timer is used by the self-test program to blink the front panel LED indicator on residual errors. The self-test turns it on for no errors, and leaves it off and halts on fatal errors.

5.3 Terminal Server Memory

Figure 5-3 shows the allocation of terminal server address space. The addressing logic allows the CPU, under program control, to direct its access the following memory and devices:

- DART and LANCE control registers

- 128K bytes of dynamic program random access memory (program RAM)

- 32K bytes of program read only memory (program ROM)

- 512 bytes of electrically erasable programmable ROM (EEPROM)

- 32 bytes of physical address programmable ROM (PA PROM)

It also allows the LANCE to perform direct memory access (DMA) transfers with program RAM

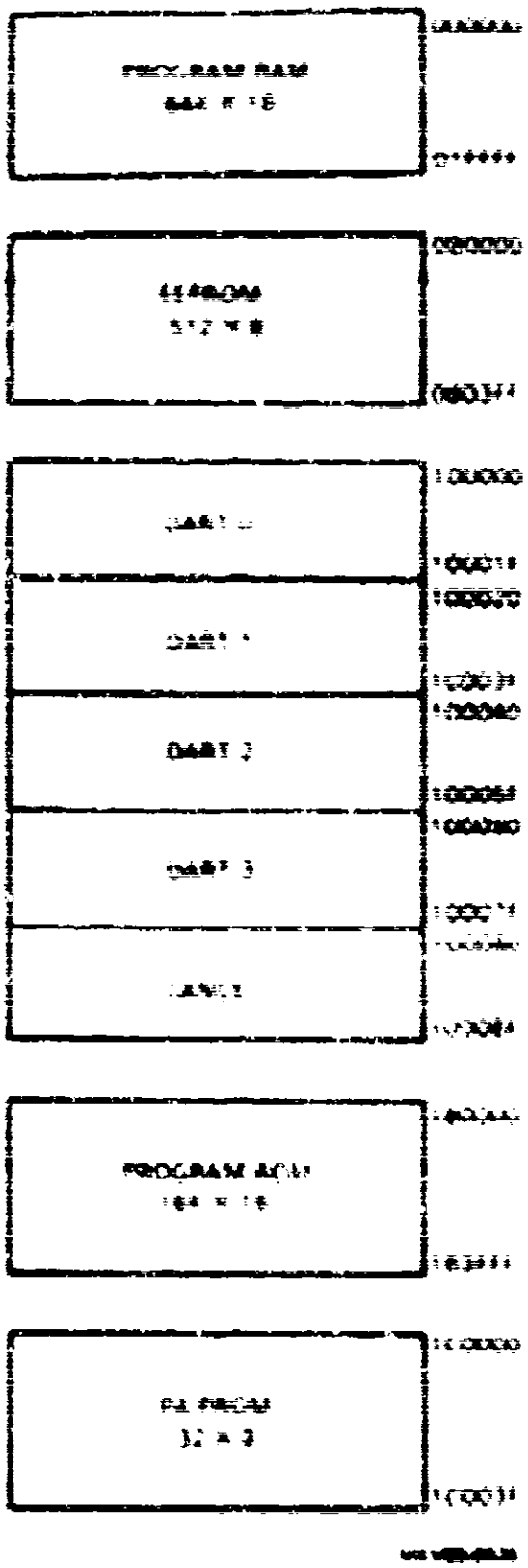


Figure 6-3: DECserver 100 Terminal Server Address Space Allocation

5.3.1 Address Selection

Terminal server address space is selected by program address bits B<31:16> <A23:A08>, as shown in Figure 5-4. Unused bits are reserved for future expansion.

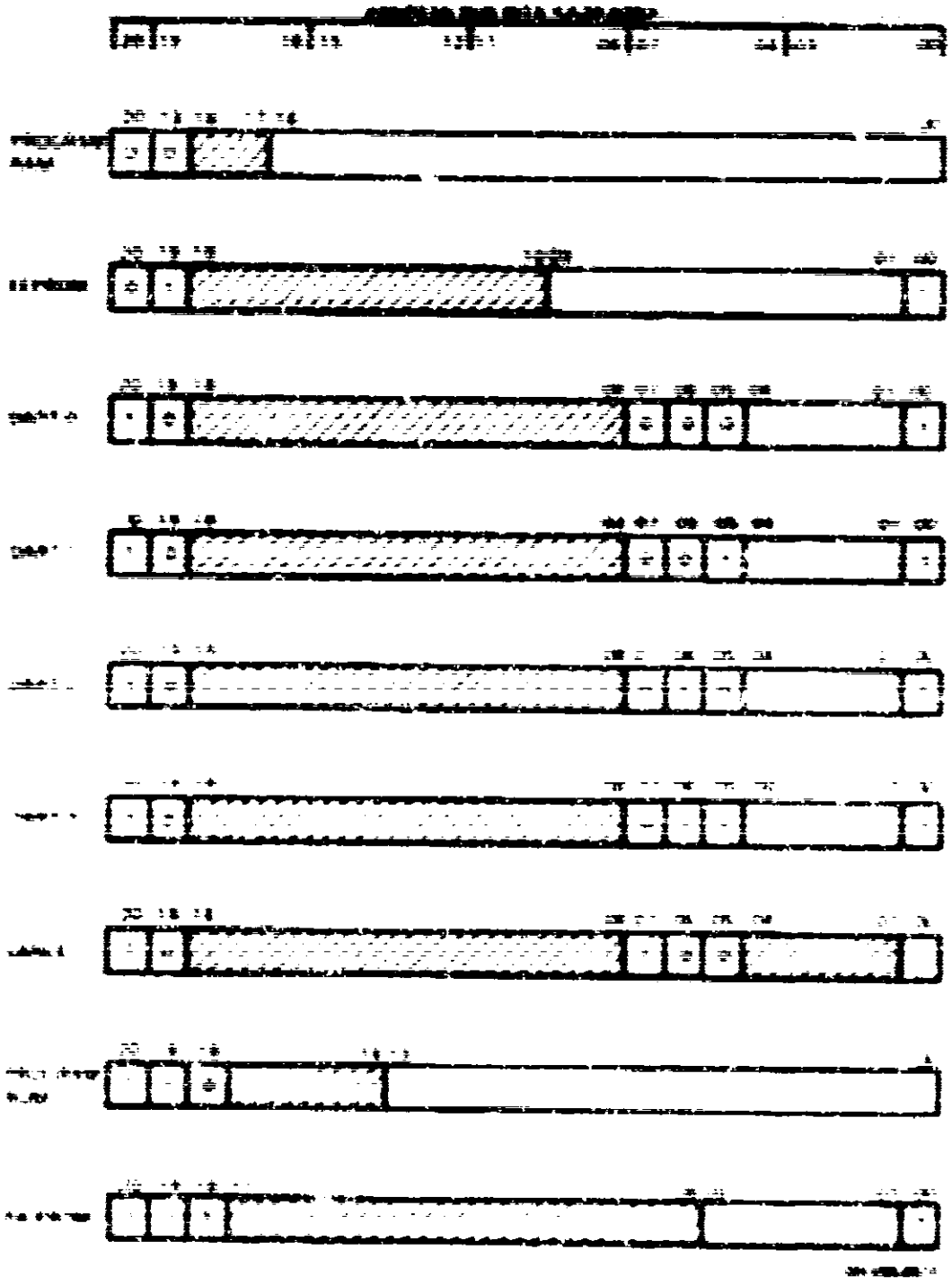


Figure 5-4 CPU Selection of DECserver 100 Terminal Server Address Space

Figure 5-5 shows the four-bit programmable array logic (PAL) which is programmed to assert the memory and device select signals listed in Table 5-7. Address bits BUS<A20:A01> bits are used by the terminal server except for BUS<A17> (unconnected).

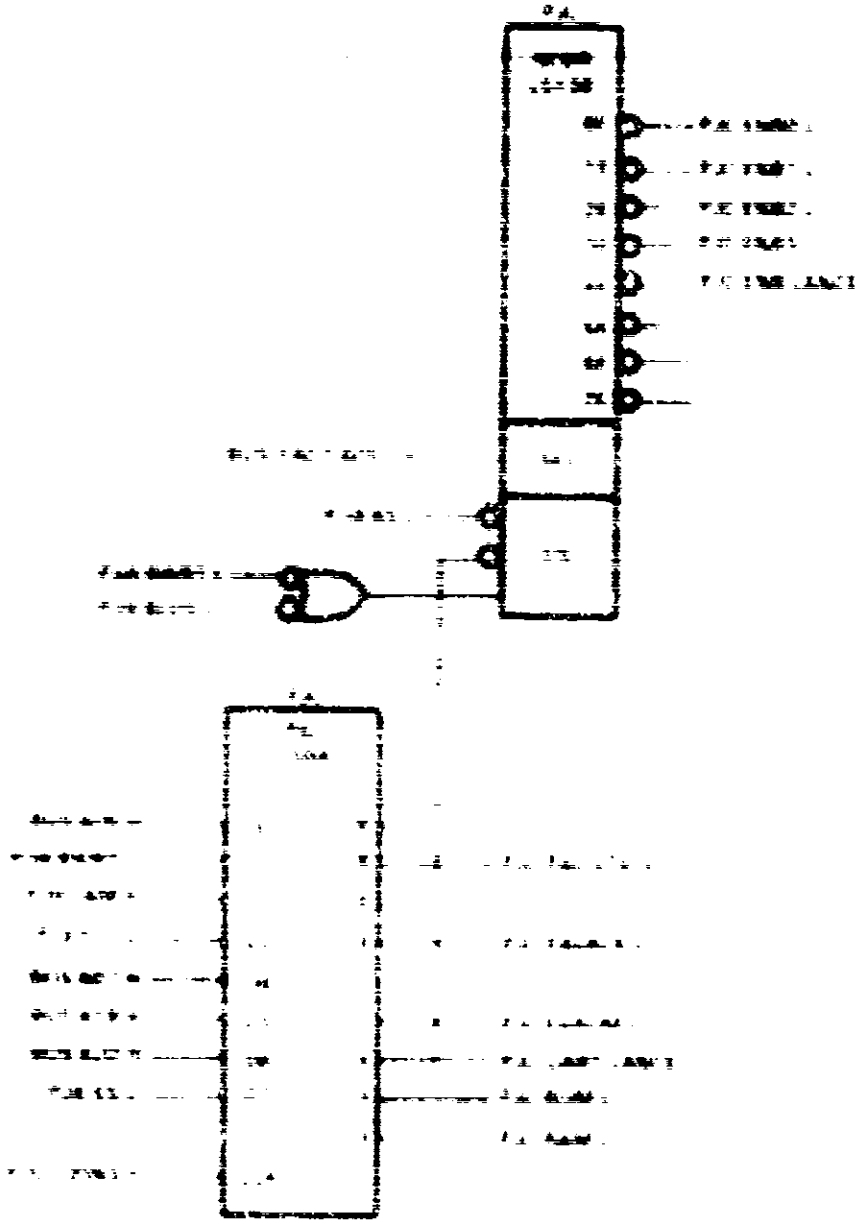


Figure 5-5 Address Decoder and PAL Block Diagram

Table 6-7 Programmable Array Logic (PAL) Output Functions

Signal	Function
PAL RAM L	Selects program RAM
PAL ROM L	Selects program ROM
PAL EEPROM L	Selects EEPROM
PAL UART LANCE L	Enables the address decoder to allow assertion of the following signals: PAL ENB [...] L: Selects one of the four DARs; PAL ENB LANCE L: Selects the LANCE
PAL PALACK L	PAL response to AS L from the CPU on transfers with memory address space or DMA registers (LANCE selects PAL LANCE L on the CPU to transfers with LANCE registers)
PAL PALSTATE [ ] L	Enables writes to EEPROM in order to change the parameter select variables

BUS ADR [ ] is also not used. However, address bus ADR [ ] is used internally by the CPU and the LANCE to assert the upper data strobe or lower data strobe signals (BUS UDS L or BUS LDS L) on a byte transfer. (Both are asserted on a word transfer.)

The CPU uses the complement of address bus ADR [ ] as a bit number:

- ADR [ ] = 0: Selects data bits BUS DAL [ ] (lower byte)
- ADR [ ] = 1: Selects data bits BUS DAL [ ] (upper byte)

As bus slave, the LANCE is addressable only on word boundaries. As bus master, it uses the standard DIGITAL convention in selecting the upper or lower byte:

- ADR [ ] = 0: Selects data bits BUS DAL [ ] (lower byte)
- ADR [ ] = 1: Selects data bits BUS DAL [ ] (upper byte)

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

6.3.2 Power-up Addressing

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

Location 0 is normally defined in program RAM address space. However, the PAL logically swaps program ROM and program RAM address space, selecting program ROM under the following power up conditions.

1. Output pin 2 from each DART asserts the complement of output port register bit OP0 (OP0 = 0) on power up.

2. P3H JAM H UP2 from the fourth DART is asserted high to the PAL which asserts PAL ROM L, enabling program ROM.

3. Location 0 of program ROM points to the self-test power up routine. P3H JAM H is then negated or asserted under program control.

The self-test program starts up in location 0 of the program ROM from where it performs the first part of the tests. It then copies images of the later routines to program RAM from where it performs the remaining tests. It then jumps to the runnable program which issues a request for a downline load of the LAT operating software from a host but it volunteers. Sections 3.2 and 3.3 introduce and describe the downline load process.

6.3.3 DART and LANCE Register Addressing

The CPU selects DART and LANCE internal registers by asserting address lines into BI S AD0-AD7 as follows.

1. With BI S AD0 AD7 H equal to 10, the PAL asserts PAL LANCE/DART L, enabling the address decoder.

2. BI S AD0-AD7 to the address decoder select one of four DART s or the LANCE by asserting one of the signals PAL ENABLE DART 0 L or PAL ENABLE LANCE L.

3. An internal DART register is selected by BI S AD4-AD5 for a read or write transfer. DART registers are described in Section 3.4.1.

4. The LANCE contains an internal register address port (RAP) that selects one of four control status registers (CSR). The LANCE registers are described in Section 3.5.1.

5.3.4 Program Random Access Memory (Program RAM)

The terminal server uses 128K bytes of dynamic RAM as the operating memory where it stores and executes the LAT operating software. Program RAM uses a multiplexed eight-input row and column addressing method selected and clocked by a delay line as shown in Figure 5-6.

CPU-initiated transfer – The CPU initiates a transfer with program RAM as follows:

1. The CPU asserts an address on BUS < A20:A01 > and asserts buffered address strobe (*PJA RAS L*).
2. With BUS < A20:A01 > equal to (0), the PAL asserts *PJC RAM L*, enabling the CPU input to the delay line.

LANCE-initiated transfer – The LANCE initiates a transfer with program RAM as follows:

1. When the LANCE gains access to the bus, the bus arbitrator asserts LANCE hold acknowledge (*PJD LHLDA L*), enabling the LANCE input to the delay line.
2. The LANCE asserts an address on address BUS < A16:A01 > and asserts buffered address latch enable (*PJK BALE H*), which causes the address latch to latch BUS < A16:A01 >.

Data transfer cycle – The data cycle of a DMA transfer initiated by either the CPU or the LANCE is completed as follows:

1. Buffered write (*PJA BWRT L*) defines the transfer as a read or write.
2. When the CPU asserts buffered address strobe (*PJA RAS L*) or the LANCE deasserts *PJK BALE H*, the low-to-high transition to the delay line generates row address strobe (*PJE RAS L*).
3. On the low-to-high transition at the delay line output, the row and column addresses are loaded to program RAM, selecting the transfer address as follows:
   a. The address value on BUS < A16:A01 > is asserted on *PJE MA < 8:1 >* from the address multiplexer and is clocked to the program RAM row address register by *PJE RAS L*.
   b. When RAM acknowledge (*PJE RAMACK L*) is asserted by the delay line, the address value on BUS < A16:A08 > is asserted on *PJE MA < 8:1 >* from the address multiplexer.

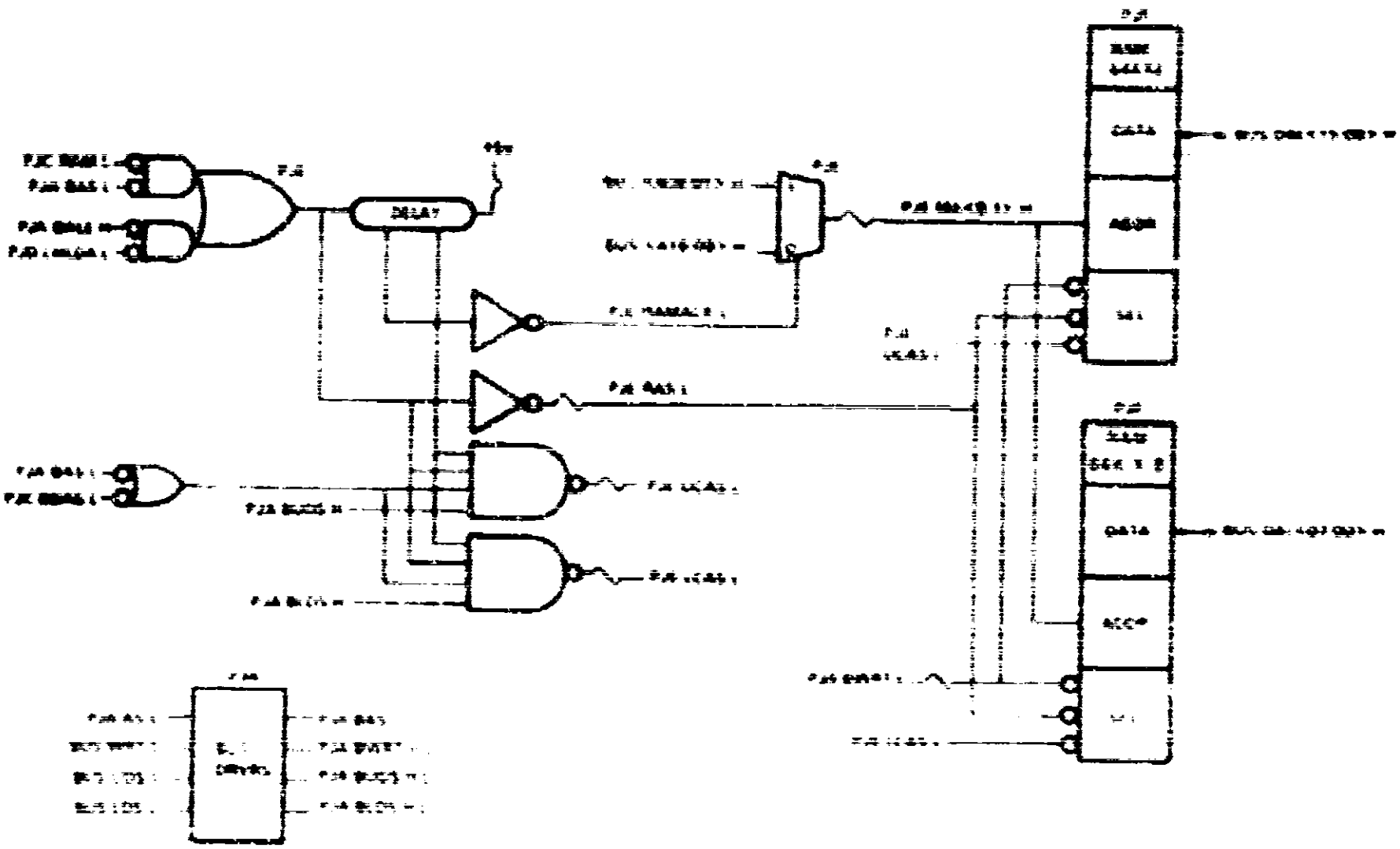


Figure 6-6: Program RAM Block Diagram

When the CPU or LANCE asserts BUS UDS L and/or BUS LDS L, PJE MA<8:1> are clocked to the program RAM column address register by either or both of the following signals.

- upperbyte column address strobe (PJE UCAS L)
- lowerbyte column address strobe (PJE LCAS L)

On a read, the program RAM data lines are enabled as data outputs by PJE UCAS L and/or PJE LCAS L.

6.3.6 Program Read-Only Memory (Program ROM)

The terminal server uses 16K bytes of program ROM, shown in Figure 6-7, to permanently store the following firmware programs:

- Self-test program
- Initialize program
- On-line debugging tool (ODT) program

Program ROM locations are selected as follows:

1. With BUS AS L and BUS A<15:14> equal to 0, the PAL asserts PJE ROM1. With BUS AS L and PJE ROM1 EN L asserted, selecting program ROM.
2. BUS A<13:4> then select one of 1K word locations. The upper and lower bytes are selected by PJE BUHDS L and PJE BLDS L.

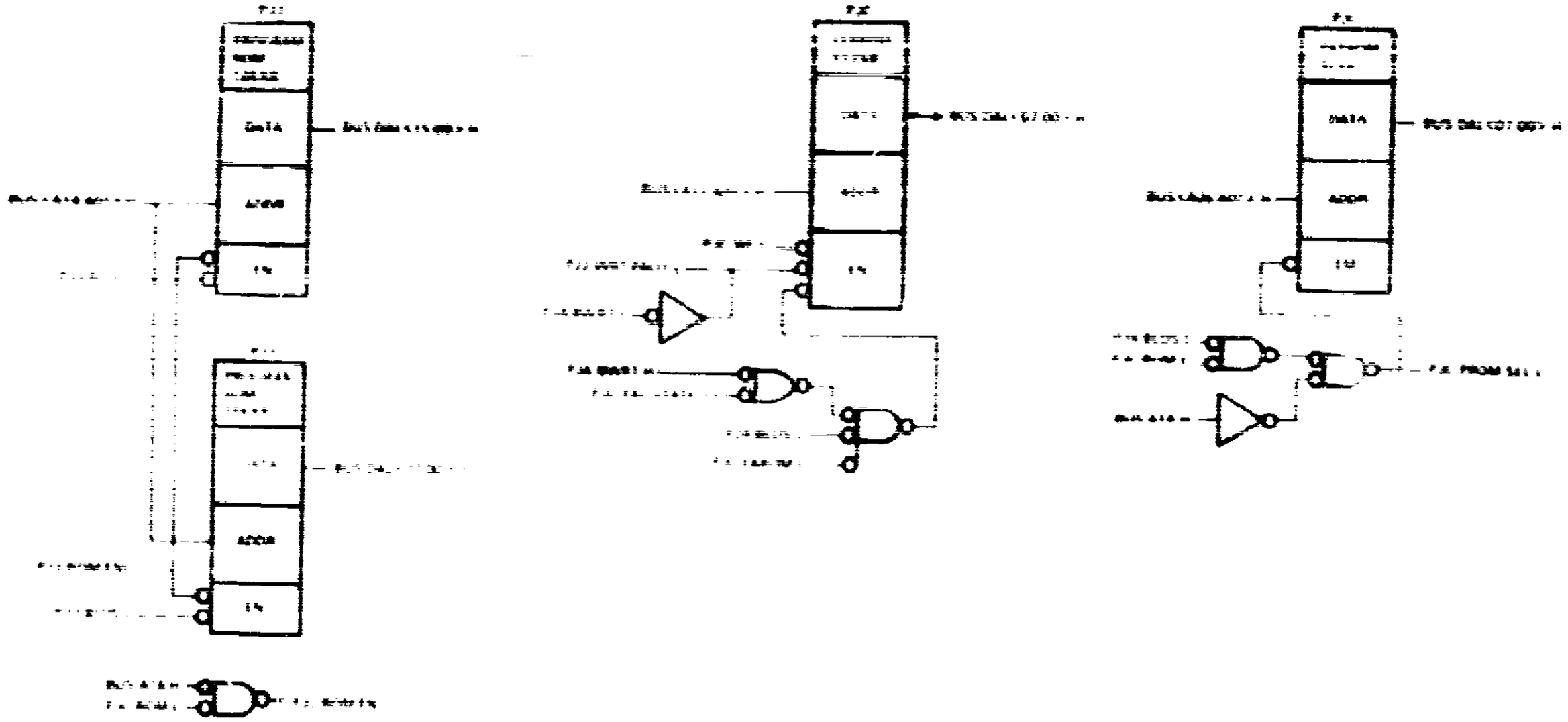


Figure 6-7: Program ROM, EEPROM, and PA PROM Block Diagram

5.4 Terminal interface

The DEC server 100 Terminal Server uses four dual asynchronous receiver/transmitter (DART) chips to communicate with up to eight ASCII terminals. Each DART operates under program control of the CPU and is serviced by program interrupt.

The CPU selects DART internal registers on address bus bits BUS<A20:A19> and BUS<A07:A05> as shown in Figure 5-4.

In the program, address bit <A00> must be a 1 to select the lower byte for a data transfer. BUS<A04:A01> then select the hexadecimal DART register address for a read or write function as shown in Figure 5-4.

Table 5-6 provides a summary of the DART registers, their hexadecimal bus addresses and their read and write functions. A dash (–) indicates that the function is undefined.

NOTE

Except for the read/write registers, the read and write functions of all addresses are mutually exclusive.

Table 5-5: Summary of DART Register Address Functions

Hex Bus Address	Read Function	Write Function
[illegible]	Mode Register A	Mode Register A
[illegible]	Status Register A	Clock Select Register A
[illegible]		Command Register A
[illegible]	RX Holding Register A	TX Holding Register A
[illegible]	Input Port Change Register	Auxiliary Control Register
[illegible]	Interrupt Status Register	Interrupt Mask Register
[illegible]	Counter/Timer Upper Register	Counter/Timer Upper Register
[illegible]	Counter/Timer Lower Register	Counter/Timer Lower Register
[illegible]	Mode Register B	Mode Register B
[illegible]	Status Register B	Clock Select Register B
[illegible]		Command Register B
[illegible]	RX Holding Register B	TX Holding Register B
[illegible]		
[illegible]	Input Port State Register	Output Port Configuration Register
[illegible]	Start Counter/Timer Command	Set Output Port Register Bits
[illegible]	Stop Counter/Timer Command	Clear Output Port Register Bits

NOTE

The above addresses select DART 0.
To select DART 1, add an offset of [illegible]
To select DART 2, add an offset of [illegible]
To select DART 3, add an offset of [illegible]

5.4.1 Dual Asynchronous Receiver/Transmitter (DART)

Figure 5-8 shows the DART interface and control lines. Table 5-9 describes the DART operating signals as they are used in the DECserver 700 configuration.

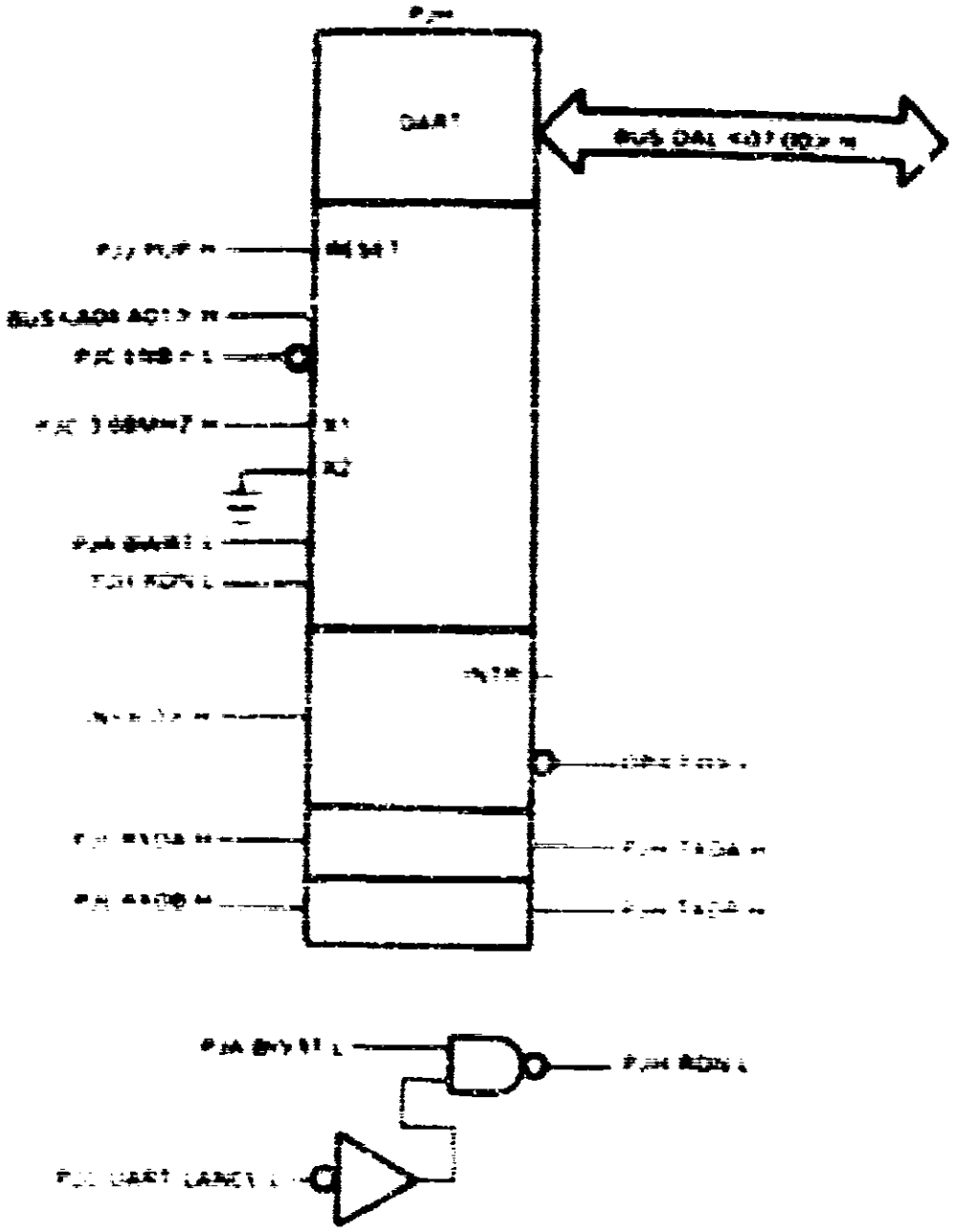


Figure 5-8 DART Block Diagram

Table 5-9 DART Signal Descriptions

Signal Name	Description
Intel Address Bus Signals	
Address Bus (BE[3:0]# and A[4:2])	Internal DART registers are selected by the CPU with four of the transfer address bus lines. These are reverse-order lines for the DART, which can only be a bus slave.
Data Bus (D[31:0])	The lower byte of the bidirectional tristate data bus is used to transfer status and register data on a CPU-initiated transfer with a DART.
Asynchronous Bus Control Signals	
Enable (EN#)	Selects one of four DARTs on a CPU-initiated transfer, where 0 is the DART number.
Bus Write (BUS_WR#)	Enables the inputs of the selected register to receive write data from the data bus.
Read Enable (RD#)	Enables the tristate data bus outputs and asserts read data from the selected register.
Processor Local Bus (Peripheral Signals	
Receive Data (RCV_DAT[3:0])	Accepts serial data from the receive line selected by the receive port number (0 - 3).
Transmit Data (XMT_DAT[3:0])	Outputs serial data to the transmit line selected by the transmit port number (0 - 3).
Clock Input X1 (CLK_IN_MHZ)	Provides the basic clock (16 x baud) from which the transmit and receive baud rates are derived for each channel.
Output Port Signals	
Chipset Port Select (CPS_EN[3:0])	One of four (0 - 3) active-low signals selected from channels A and B (A and B) by the receive or transmit port number (0 - 3).
Output Port Data (OPD_DAT[3:0])	One of four (0 - 3) general-purpose output signals selected from channels A and B (A and B) by the receive or transmit port number (0 - 3).

Table 6-9 DART Signal Descriptions (cont'd)

Signal Name	Description
Output Port (IP):	(IP): asserts the following counter/timer signals from each DART.
First DART	PIN (TINT): from the refresh timer interrupts the CPU to refresh the dynamic program RAM. It is also used as the clock input to the watchdog timer and LED timer.
Second DART	PIN (TINT): from the counter/timer is used as the slot timer for the group on.
Third DART	PIN (T2): from the watchdog timer interrupts the CPU to initialize the peripheral servos and reload the servo voltages of the timer as not previously stored by the program.
Fourth DART	PIN (PAS): from the counter/timer blank the LED indicator on to indicate self-test errors and turns it off for the others.
Output Port (IP):	(IP): asserts the following selection signals from each DART.
First DART	PIN (INT): from the DART.
Second DART	PIN (AS/DS/R/W): to register writing and reading.
Third DART	Not used.
Fourth DART	PIN (AS): when combined with a power-up signal from the MMU and MAM address space.
Output Port (Write):	(Write): are used in the DART control register for the first DART chip (PIN): It selects the state of output test signals in the DART. In the maintenance mode (IP): the signals are used by the self-test program to provide an oscilloscope synchronizing signal.
Input Port Select:	
Input Port (Read):	Not used in the DART control port.

Table 5-11: Mode Register MR2A and MR2B Bit Functions (Read/Write Address 100201 and 100211)

Bits	Name	Description
<7:6>	Channel Mode Select CHNL MODE SEL A/B:	Selects the channel mode as follows: 0 Normal (full duplex) 1 Auto Echo (transmitter and receiver data) 2 Local Loopback (receiver and transmitter data) 3 Remote Loopback (transmitter and receiver data)
<5:4>	Transmit Request to Send Select TX RTS SEL A/B:	With this bit set, output ports OP0 and OP1 assert the following states: OP0 = TX RTS A OP1 = TX RTS B
<3:2>	Transmit Clear to Send Select TX CTS SEL A/B:	With this bit set, output ports OP0 and OP1 assert the following states: OP0 = TX CTS A OP1 = TX CTS B
<1:0>	Stop Bit Length STOP BIT LGTH A/B:	Sets the length of the stop bit in bit time increments. The DET device will support the following stop bit values: 0 [illegible] 1 [illegible] The receiver only checks for a programmed level of the bit time values.

5.4.2.3 Clock Select Registers (CSRA and CSRB) – Table 5-13 describes the bit write functions for CSRA and CSRB.

Each register selects the receive and transmit baud rates for its channel from one of two sets, depending on the state of auxiliary control register bit ACR<07> (see Section 5.4.2.7).

Table 5-13 Clock Select Register CSRA and CSRB Bit Functions (Write Address 100003 and 100013)

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

The ranges of baud rates are selected for either channel A or the following states of ACR<07> (see Table 5-11):

RCV/XMT CLK SEL Value	Set 0 (ACR<07> equals 0)	SET 1 (ACR<07> equals 1)
0000	50	75
0001	110	110
0010	134.5	134.5
0011	200	150
0100	300	300
0101	600	600
0110	1200	1200
0111	1050	2000
1000	2400	2400
1001	4800	4800
1010	7200	1800
1011	9600	9600
1100	38.4K	19.2K
1101	Timer	Timer
1110	IP4 – 16X mode	IP4 – 16X mode
1111	IP5 – 16X mode	IP5 – 16X mode

5.4.2.4 Command Registers (CRA and CRB) – Table 5-14 describes the bit write functions for CRA and CRB. Each register, when written, supplies the specified command to its channel.

Table 5-14: Command Register CRA and CRB Bit Functions (Write Address 100005 and 100015)

Bits	Name	Description
[illegible]	MISC	Unused and must be zero.
[illegible]	Miscellaneous Command (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 (RBCI) (see Table 5-17); 6 – Start Break; 7 – Stop Break.
[illegible]	Disable Transmitter (DIS TX A-B)	Disables the channel A or B transmitter. Resets the TX EMPTY and TX RDY status bits (SRA or SRB) (see Table 5-12).
[illegible]	Enable Transmitter (EN TX A-B)	Enables the channel A or B transmitter. Sets the TX RDY status bit (SRA or SRB) (see Table 5-12).
[illegible]	Disable Receiver (DIS RX A-B)	Disables the channel A or B receiver, except in the multidrop mode (MR1A or MR1B – see Table 5-10).
[illegible]	Enable Receiver (EN RX A-B)	Enables the channel A or B receiver. (The [illegible] special wake-up mode is not used in the [illegible] mode.)

5.4.2.5 Receive Holding Registers (RHRA and RHRB) – RHRA and RHRB are the receive data read registers.

First-in/First-out (FIFO) buffer – RHRA and RHRB each consist of a three-byte FIFO that acts as a small 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 RX RDY B bit is set (SRA or SRB bit <0>) 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 <0>) if all three locations for the channel are filled with data. Reading a character empties a FIFO location and negates FFULL (unless another 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 DEC server 100 Terminal Server.

5.4.2.6 Transmit Holding Registers (THRA and THRB) – THRA and THRB are the transmit data write registers.

When a register is loaded with a character by the CPU, the byte is parallel loaded to the transmit shift register from where it is shifted out serially to the terminal.

5.4.2.7 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> Bit Function (Write Address 100000)

Bit	Name	Description
[illegible]	Baud Rate Generator Select (BRG SEL)	Selects one of two transmit and receive baud rate clocks as described for CSRA and CSRB bit fields <07:04> and <03:00>, see Table [illegible]
[illegible]		Used with the counter/timer register, see Table [illegible]
[illegible]		Used with the input port change register, see Table [illegible]

5.4.3 Input and Output Port Registers

Figure 5-10 shows the register bit formats, hexadecimal addresses, and read/write functions for the input port (IP) and output port (OP) registers.

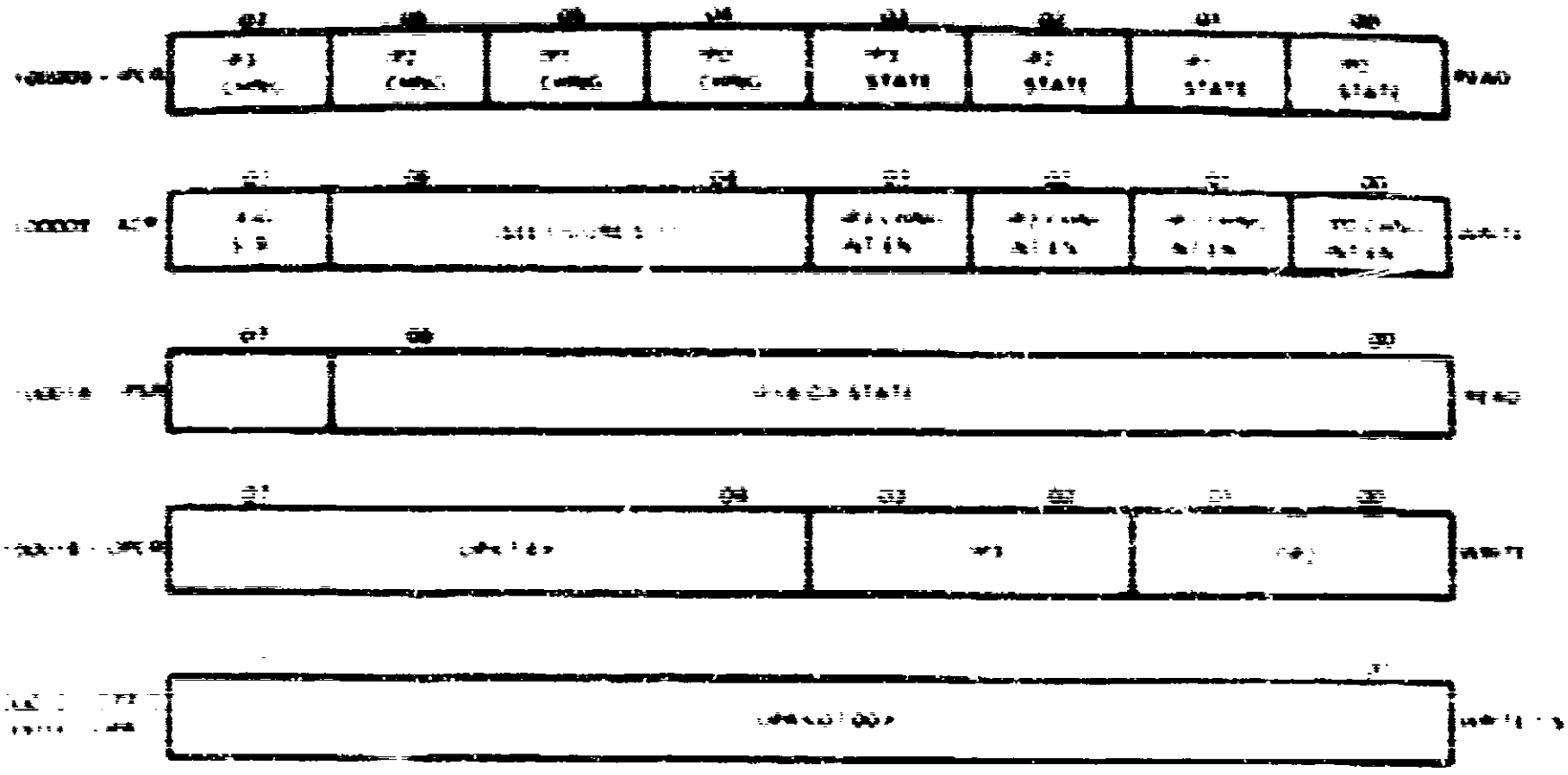


Figure 5-10 DART Input and Output Port Register Bit Formats

5.4.3.1 Input Port Change Register (IPCR) – Table 5-16 describes the bit read functions for the IPCR.

Table 6-16 Input Port Change Register (IPCR) Bit Functions (Read Address 100002)

Bits	Name	Description
<7:04>	Input Port <3:0> Change IP <3:0> CHG	A bit is set by a change of state of the signal on the corresponding input port. The change can be a transition to the new state
		A bit being set, with the corresponding bit set in ACR <02:00> (see Table 6-17), disarms IRR <07> (see Table 6-9). The INTR output is also asserted high (unless the IRR is serviced).
		All bits are cleared coincident with [illegible] when the register is read or address 4
<03:00>	Input Port <3:0> State IP <3:0> STATE	These bits assert the states of the signals on input ports IP <3:0>

6.4.3.2 **Auxiliary Control Register (ACR <03:00>)** - Table 6-17 describes the bit write functions for ACR bits ACR <3:0> which are used with the IPCR.

Table 6-17 Auxiliary Control Register ACR <03:00> Bit Functions (Write Address 100002)

Bits	Name	Description
<7:4>		Used with the input port registers (see Table 6-16)
<3:2>		Used with the counter/timer register (see Table 6-12)
<1:0>	Input Port <3:0> Change Interrupt Enable IP3:IP2:IP1:IP0	When set, one the corresponding bit becomes set in IPCR <07:04> (see Table 6-16). INTR <07> is also set (see Table 6-9). The INTR output is also asserted high (unless the IRR is serviced).

6.4.3.3 **Input Port State Register (IPSR)** - The current states of the signals connected to input ports IP3:IP0 are read on IPSR <07:04>. IPSR <03:00> is not used and IPSR <07:00> always reads as a 0.

5.4.3.4 Output Port Configuration Register (OPCR) – Table 5-19 describes the bit write functions for the OPCR which controls the function of output ports OP7:0.

Table 5-19. Output Port Configuration Register (OPCR) Bit Functions (Write Address 1BH, 1BH)

Bits	Name	Description
7:0	Output Port 7:4 Select (OP7:4 SEL)	Control output ports OP7:4 as one of the following two ways: Logical Each clear bit enables the complement of the corresponding OPR bit to output port OP7:4. Set Each set bit causes the corresponding output port to assert the following signal: OP7 – Asserts channel B transmit interrupt request after completion of TX BUFFER. OP6 – Asserts channel B transmit interrupt request after completion of TX BUFFER. OP5 – Asserts channel B receive interrupt request after completion of RX BUFFER. OP4 – Asserts channel A transmit interrupt request after completion of RX BUFFER A and in the [illegible] mode.
3:0	Output Port 3 Select (OP3 SEL)	Enable output port OP3 and OP2 to assert the following signals: 0 – Output ports OP3 and OP2 assert the complement of output port register bits OPR3 and OPR2. 1 – Asserts the counter/timer output at the complement frequency divided by two in square wave mode.

Table 9-10. Output Port Configuration Register (OPCR) Bit Functions (Write Address 1B0018) (Cont)

Bits	Name	Description
<illegible>		[illegible]
[illegible]	Output Port [illegible] Select (OP3–OP0)	[illegible]

9.4.3.5 Output Port Register (OPR) – The OPR may be used as a general purpose register or a control register. It is written with [illegible] of two addresses that set or clear the selected bits as follows:

- **Address [illegible] (set bits)** – Writing a byte of data to the address sets the selected bits as follows:

  1 = Set the bit
  0 = No change

- **Address [illegible] (clear bits)** – Writing a byte of data to the address clears the selected bits as follows:

  1 = Clear the bit
  0 = No change

8.4.4 Interrupt Control and Counter/Timer Registers

[illegible] shows the register bit formats, hexadecimal addresses, and read/write functions for the interrupt control and [illegible]

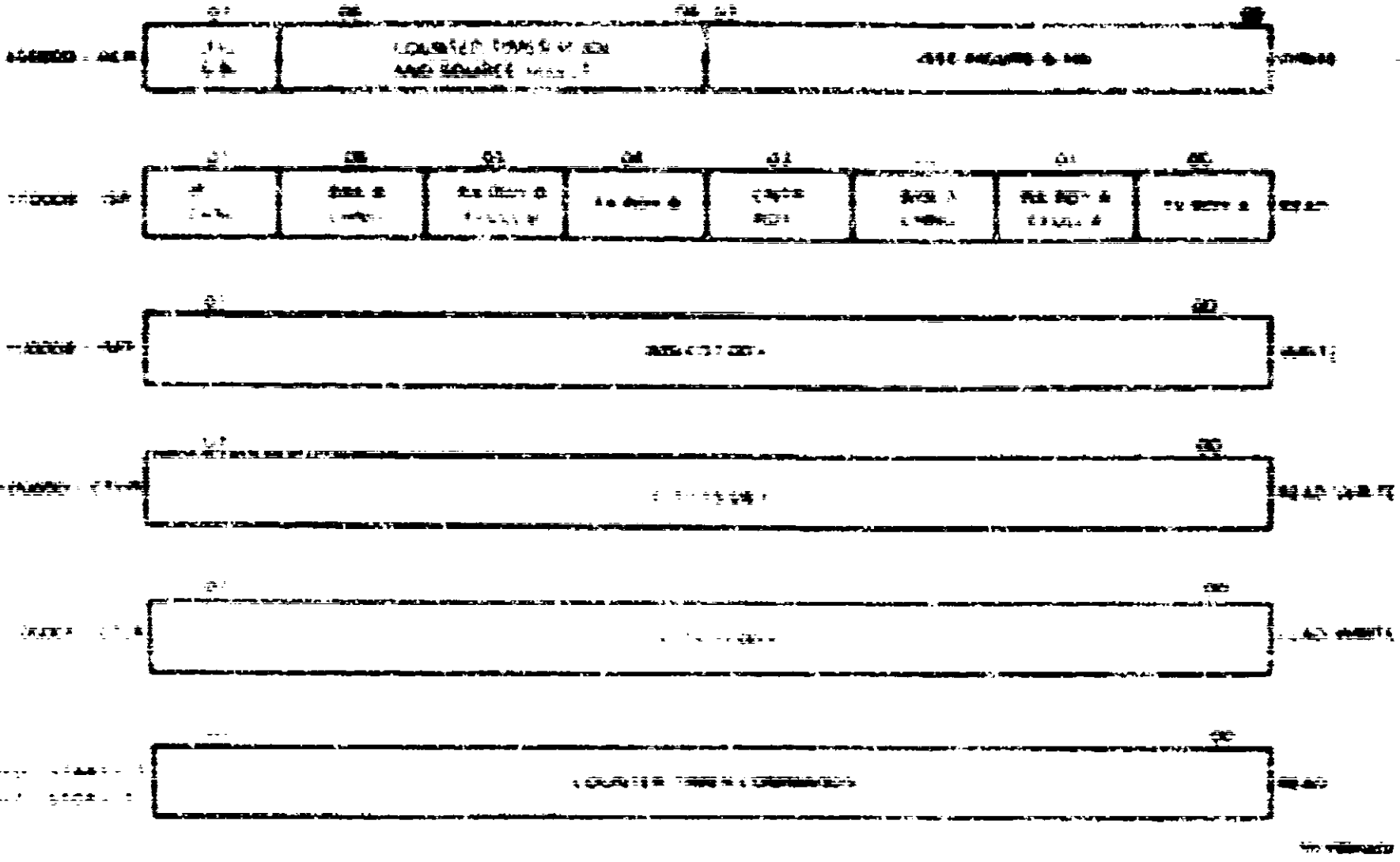


Figure 8-11 DUART Interrupt Control and Counter/Timer Register Bit Formats

5.4.4.1 Auxiliary Control Register (ACR<02:04>) – Table 5-19 describes the bit write functions for ACR bits ACR<02:04> which are used with the counter/timer register.

Table 5-19 Auxiliary Control Register ACR<02:04> Bit Functions (Write Address [illegible])

Bits	Name	Description
[illegible]	-	Used with the clock select registers (see Table 5-11).
[illegible]	Counter/Timer Mode and Source Select (CTMS bits)	Set the counter/timer register to the counter or timer mode and select the clock source as follows:

Code	Mode	Clock Source
0	Counter	External input (IP2)
1	Counter	TxCA (channel A Tx) the transmit clock
2	Counter	TxCB (channel B Tx) 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

[illegible]	-	Used with the input port register (see Table [illegible]).

5.4.4.2 Interrupt Status Register (ISR) – Table 5-20 describes the bit read functions for the ISR status flags. The ISR is a status register that reflects the states of the interrupt flags.

The interrupt (INTR) signal output is asserted from the DART if the corresponding flag bit is set in the IMR.

Table 5-20: Interrupt Status Register (ISR) Bit Functions (Read Address 100000)

Bits	Name	Description
<07>	Input Port Change (IP CHNG)	Reads as a 1 if any IPCR <07:04> bit is set (see Table 5-16) and the corresponding bit is set in ACR<07:04> (see Table 5-17). The INTR output is also asserted from the DART reserved on the DEC server 100.
<06>	Channel B Change = Break (BRK B CHNG)	Indicates that the channel B receiver detected the beginning or end of a break. Cleared by the reset break change interrupt command from command register B (see Table 5-14).
<05>	Receive Ready B or FIFO Full B (RX RDY/FFULL B)	Asserts either of the following states, depending on the state of MR1B <06>. MR1B<06> = 0 Asserts SRB bit <05> (RX RDY B). MR1B<06> = 1 Asserts SRB bit <05> (FFULL B).
<04>	Transmit Ready B (TX RDY B)	Asserts the state of SRB bit <04> (TX RDY B).
<03>	Counter Ready (C/T R DY)	Set in the counter/timer register depending on the register mode. Counter Mode Set when the counter register reaches the count value (loaded by the stop counter command) stops the counter. Timer Mode Set each time the counter/timer register reaches a zero count (loaded by the stop counter command) and does not stop the counter. The counter is reloaded from the holding register on each zero count.

Table 5-20 Interrupt Status Register (ISR)
Bit Functions (Read Address [illegible])
(Cont)

Bits	Name	Description
<7:6>	(Re (T (F	Indicates that the channel A receiver detected the beginning or end of a break. Is cleared to the reset break change interrupt command from command register A (see Table 5-14)
<5:4>	Receiver Ready A or FIFO Full A (RX RDY A/FFUL A)	Asserts either of the following states, depending on the mode of MR1A <6>.
		MR1A<6> = 0 Asserts SRA bit <1> (RX RDY A)
		MR1A<6> = 1 Asserts SRA bit <1> (FFUL A)
<3>	Transmit Ready A (TX RDY A)	Asserts the state of SRA bit <2> (TX RDY A)

5.4.4.3 Interrupt Mask Register (IMR) – Writing a 1 to a bit in the IMR enables the interrupt (INTR) output for the corresponding flag in the ISR.

5.4.4.4 Counter/Timer Registers (CTUR and CTLR) – The counter/timer contains two functional 16-bit registers, a holding register and a count register. The count register function depends on whether it is used in the counter or timer mode.

CTUR and CTLR are holding registers that store the upper and lower byte values as shown in Figure 5-11. The register addresses are write-only to the holding registers in the counter mode. They can only be read in the timer mode. The maximum value that can be loaded is [illegible] (hex).

Counter mode – The count register acts as a bit pre-down counter that sets an interrupt status register bit ISR<3> (CNTR RDY) when it reaches a final value of [illegible] (hex). A output port (IP2) may be programmed to assert the state of ISR<3>.

The commands have the following effects in the counter mode:

- Stop Counter – Stops the count register and clears ISR<3>.
- Start Counter – Loads the count register with the holding register contents and starts the count cycle.

Timer mode – The count register runs continuously as a binary up-counter, generating a square wave output at twice the time value stored in the holding register.

Each overflow sets interrupt status register bit ISR<03> (CNTR RDY) and reloads the count register from the holding register. (Output port OP3 may be programmed to invert the square-wave output.)

The commands have the following effects in the timer mode:

- Stop Timer – Clears ISR<03> but does not stop the count register.
- Start Timer – Terminates the current timing cycle, loads the count register with the holding register content, and starts a new timing cycle.

5.4.4.6 Counter/Timer Start and Stop Commands – Reading either of these two addresses issues the following counter/timer command:

Address 1300011 - Start counter/timer
Address 1300013 – Stop counter/timer

These commands have different functions in the counter or timer mode as described in Section 5.4.4.4.

5.5 Ethernet Interface

The terminal server uses the following chip combination that provides the control and signal interfacing to an Ethernet system:

- Local area network controller for Ethernet (LANCE)
- Serial interface adapter (SIA)

This section provides a summary of the signals and protocol supported by the LANCE and SIA and describes the LANCE internal register formats and functions.

5.6.1 Serial Interface Adapter (SIA)

Figure 5-12 is a block diagram of the SIA which provides an interface between the TTL signals of the LANCE and the differential signals of the H4000 DIGITAL Ethernet transceiver.

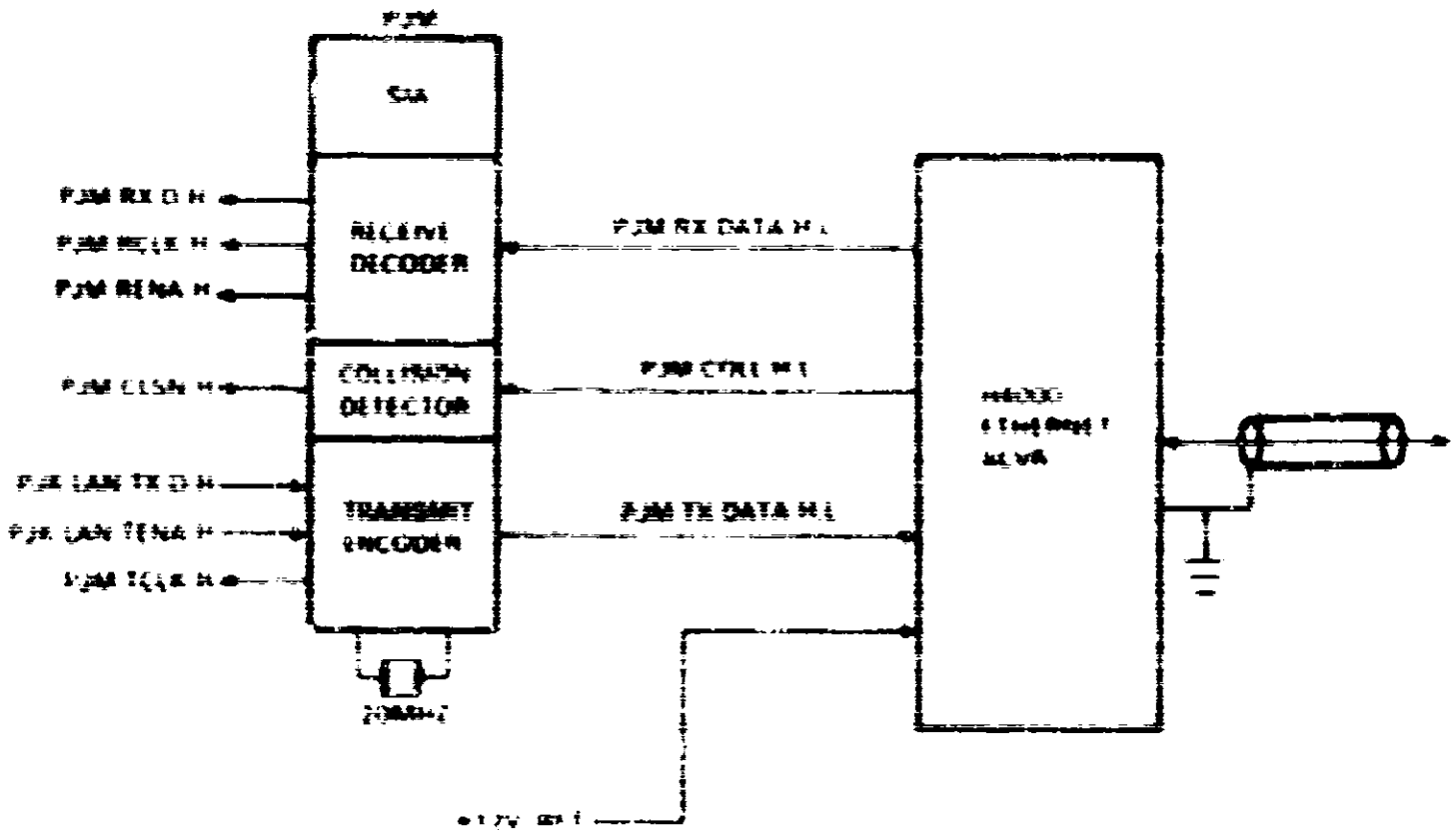


Figure 5-12: SIA Connection to an Ethernet Transceiver

Table 5-21 describes 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 differential signal from the collision comparator in the Ethernet transceiver to the SIA collision detector. Occurs on an actual collision or on completion of a normal transmission (heartbeat check)

Table 5-22 describes the TTL interface signals passed 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 the presence of a collision on the Ethernet.
Receive Enable (PJM RENA H)	Asserted by the SIA to indicate the presence of a valid signal on the Ethernet and where the SIA receiver/Manchester decoder is locked and producing the RCLK and RX D signals.
Receive Clock (PJM RCLK H)	The 10 MHz clock produced by the SIA is the synchronizing clock for the LANCE serial data receiver.
Receive Data (PJM RX D H)	The serial data stream produced by the SIA is asserted to the LANCE serial data receiver where it is clocked by RCLK.
LANCE Transmit Signals	
Transmit Clock (PJM TCLK H)	The 10 MHz clock from the SIA is the main clock for the LANCE state processor and the synchronizing clock for the LANCE serial data transmitter.
Transmit Enable (PJM LAN TENA H)	Asserted by the LANCE to enable the SIA transmitter driver to the Ethernet transceiver.
Transmit Data (PJM LAN TX D H)	The serial transmit data stream produced by the LANCE is asserted to the Manchester data encoder in the SIA where it is encoded and sent to the Ethernet transceiver.

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

5.5.2 Local Area Network Controller for Ethernet (LANCE)

Figure 5-13 is a block diagram of the LANCE as it is used in the DECserver 100 configuration.

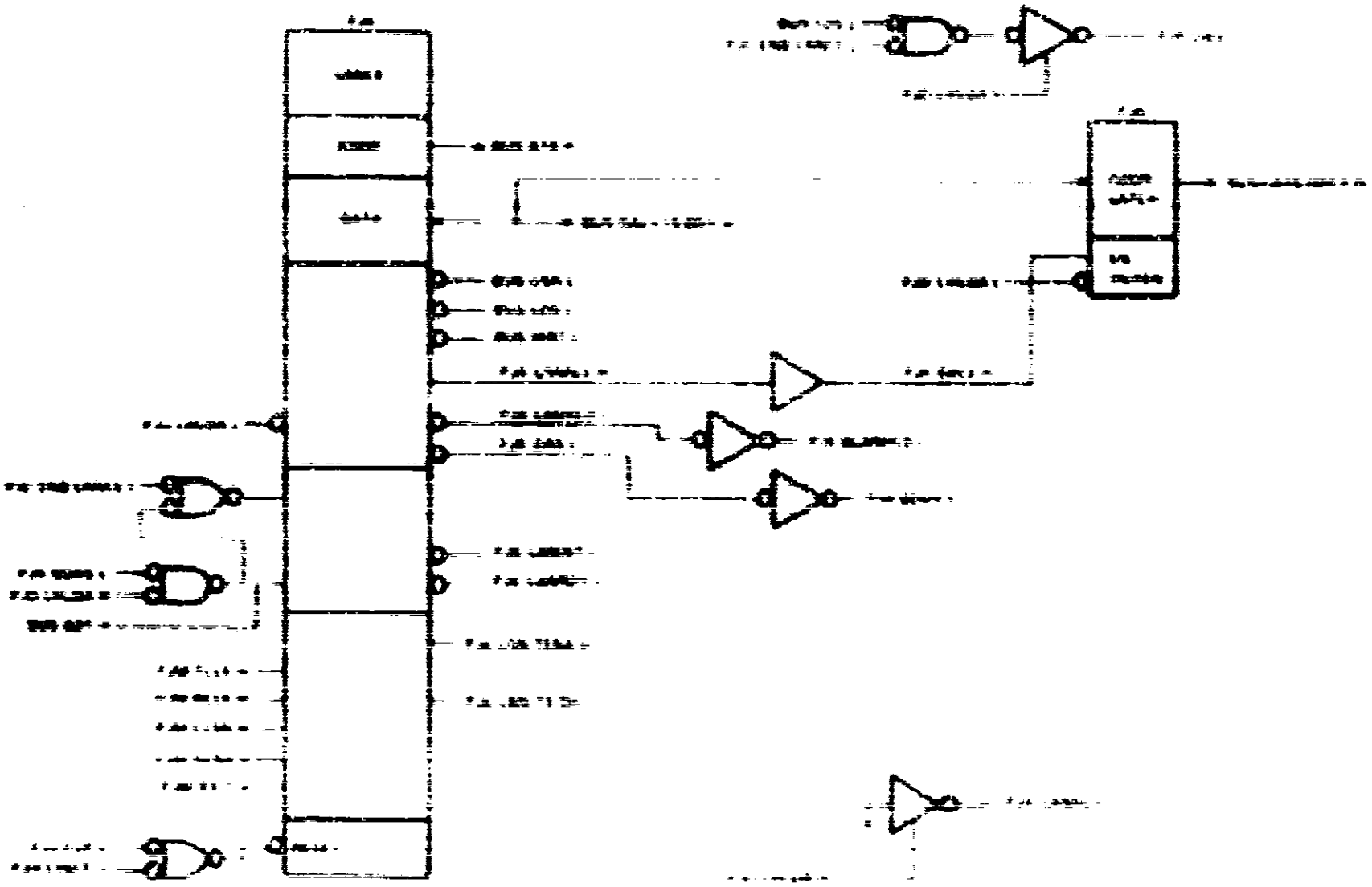


Figure 8-13 LANCE Block Diagram

Table 5-23 describes the LANCE data/address bus and operating signals. Tri-stated signals are not fully decoded and are in the high-impedance (Hi-Z) 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 24 tri-state address bus lines to access program RAM or bus master. There are three unused lines for the LANCE.
	The LANCE asserts BALE during the address cycle of a DMA transfer. It forces a bus address by opening the address latch, enabling the tri-state outputs, and asserting a memory address on BUS A16 and on the data bus. It then closes the latch, making the address at the end leaving the address asserted on BUS <A16:A01> H.
Data Bus BUS <DAL15:DAL00> H:	The 16-bit bidirectional tri-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 bus slave).
	As the bus master on a DMA cycle, the LANCE first asserts an address on these lines during the address cycle, and latches it to the address latch. It then uses the lines for the transfer of data on the data cycle.
Bus Arbitration Signals	
Bus Request LANCE Hold (OUT) BUSRQ H:	Asserted by the LANCE when requesting access to the data/address bus. BUSRQ should assert a bus request to the CPU but is not guaranteed for the entire transaction. When negated, it terminates the DMA cycle, releasing the bus.
Bus Grant Acknowledge (IN) BUSAK H:	Asserted by the bus arbitrator in response to bus grant from the CPU.
LANDLD Acknowledge (OUT) LANDLD H:	Grants bus access to the LANCE for a DMA transfer. LANDLD enables the tri-state outputs of the address latch, and asserts LANDLD to the LANCE to indicate that program RAM is ready to perform the transfer.

Table 8-23. LANCE Operating and Data/Address Bus Signals (Cont)

Signal Name	Description
Asynchronous Bus Control Signals	
LANCE Enable (PAL ENBLANCE L)	Asserted by the CPU to select the LANCE for a register transfer.
Buffered Address Latch Enable (PAL BALE H)	Asserted by the LANCE on the address cycle of a DMA transfer. BALE opens the address latch to capture bits <23:0> of the memory address asserted on BUS DAL<15:0>. When negated, BALE closes the latch which holds the address asserted on BUS A<23:0>.
Data Strobe (PAL BDAS L)	Bus Master: Asserted by the LANCE on the data cycle of a DMA transfer. Write data asserted by the LANCE is stable on the high-to-low transitions of BDAS. The LANCE latches read data on the low-to-high transitions of BDAS. Bus Slave: Received from the CPU on the data cycle of a CPU register. BDAS selects the LANCE register address port or register data port for the transfer, depending on the state of address bit BUS <A01>. BUS <A01> = 0 — Register data port BUS <A01> = 1 — Register address port
LANCE Reset (PAL LANRES L)	Bus Master: Asserted by LANCE on a DMA error. LANRES notifies the LANCE to stop operation. Write a register CSR.
	Bus Slave: Asserted by the LANCE to reset the register. LANRES notifies that the LANCE has asserted read data on the data bus or has received write data from the data bus.
Write (PAL WRITE L)	Indicates that the operation is a read or write transfer when responding to the bus master. Write — Asserted low Read — Negated high As bus master, the LANCE drives this signal for a DMA transfer. As bus slave, it receives the signal for a CPU register and LANCE register.
Synchronous Acquisition Data Strobe (PAL SYNDAS L)	As bus master, the LANCE asserts these signals on a DMA transfer to specify the valid interval for a read or write transfer.

Table 8-23. LANCE Operating and Data/Address [illegible]

Signal Name	Description
	Valid Data is on
BUS **BUS**	**BUS BUS** **BUS BUS**
UDS **LDS**	**<15:08>** **<07:00>**
[illegible]	[illegible]

[illegible] = [illegible]

[illegible]

There are six DMA signals and are shown in the LANCE module [illegible] at the latest times. They are buffered in the CPU logic providing the following outputs.

Inputs	Buffered Outputs
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

LANCE Interface
CPU / LANCE I/O

[illegible]

Reset

When a low is asserted on the reset signal the LANCE Data (DAT) bus logic and enters an idle state. Reset is asserted for either of the following reasons:

Power up (PWR UP) is made the power up logic

LANCE software (SFTWR LANCE) is under program control from the host CPU

6.5.2.1 Features – The LANCE is a 10 Mbit device that is optimized to perform the high-level Ethernet protocol. Its features include a 144-byte data buffer (SRAM) that allows it to perform burst transfers of up to eight data words with program RAM on a single DMA cycle.

After the CPU sets up and loads the LANCE control/status registers (CSRs), the LANCE performs a DMA burst to program RAM to retrieve the initialize block containing the LANCE operating parameters. During operation, it retrieves the receive or transmit descriptor rings. The descriptor rings contain the memory data buffer parameters for transfers that take place between the Ethernet and program RAM through the LANCE.

6.5.2.2 DMA Transfers with Program RAM – The LANCE is the bus driver rather than the CPU that gains access and uses the data/address bus as a bus master. It gains access to the bus through bus arbitration and performs a direct memory access (DMA) transfer with program RAM under the following procedure.

Bus arbitration cycle – The LANCE first arbitrates and gains access to the data/address bus through the following steps.

1. The LANCE asserts buffered LANCE hold (BUFFERED HLDA) to the bus arbitrator when it wishes to assert bus request (PJD BR L) to the CPU.
2. When the CPU can release the bus, it returns bus grant (PJD BG L) to the arbitrator.
3. When address strobe (PJA BAS L) is deasserted by the CPU, the arbitrator drives back grant acknowledge (PJD BGACK L) to the CPU, which in turn negates bus request (PJD BR L).
4. The arbitrator asserts LANCE hold acknowledge (BUFFERED HLDA) to the LANCE and negates bus request (PJD BR L) to the CPU.
5. When the LANCE completes the bus cycle, it negates BUFFERED HLDA L and PJD BGACK L, releasing the bus.

Address cycle – Having control of the bus, the LANCE performs the following events that select a program RAM location:

1. It enables the 3-state data control outputs (BUFFERED WRITE L, BUFFERED READ L, and BUFFERED LOW BYTE L) with the correct signals for the transfer.
2. The LANCE enables its 3-state outputs to the data bus and enables the 3-state outputs of its address latch to the address bus.

3. [illegible] on the data bus and on [illegible] address bus bit <A16> and opens the address latch to [illegible] address into [illegible]

4. [illegible] address latch [illegible] [illegible] the address onto the data bus, leaving the [illegible] on the address bus.

Data cycle – The LANCE then performs the following events that [illegible] with the selected [illegible]

1. The LANCE [illegible] data cycle by asserting buffered data strobe (PIN BUAS L) [illegible] also enables its [illegible] and asserts data on the data bus.

2. [illegible] buffered data strobe (PIN BUAS L) [illegible] data from the data bus.

3. [illegible] all of its [illegible] and returns them to the high impedance (Hi-Z) state.

4. [illegible] buffered LANCE hold (PIN BUAHLD L) to the bus arbitrator, which negates bus grant acknowledge (PIN BUAGN L) and LANCE hold acknowledge (PIN LHLDA L) [illegible] by the transition.

6.5.2.3 Ethernet Operating Modes – The LANCE [illegible] the following modes of operation on the Ethernet:

[illegible] Mode – In the normal mode, the LANCE [illegible] a DMA [illegible] a transmit descriptor ring from memory. Further DMA [illegible] data from a [illegible] buffer in memory and the LANCE [illegible] it out on the Ethernet. It [illegible] on error data from [illegible]

As described in Section [illegible], the LANCE [illegible] each data frame with a preamble pattern of alternating ones and zeros [illegible] and Manchester encoders to [illegible] the LANCE [illegible] a 32 bit cyclic redundancy check (CRC) on the [illegible] and [illegible] minimum frame size (64 bytes) [illegible] appends the [illegible] to the end of the data [illegible] the modified [illegible] it provides an interframe gap (9.6 µs after the CRC)

On a collision, the LANCE [illegible] the data [illegible] to the external [illegible] buffer [illegible] contained in the [illegible] [illegible]

Promiscuous mode – When a station is present on the Ethernet cable, the received preamble allows the SIA receive circuits to settle. The Manchester decoder synchronizes itself to the incoming data frame and locks onto the encoded signal carrying the incoming data frame.

The LANCE calculates the CRC on the received source address field, destination address fields, the data type and data fields, and the received CRC value. If the result of the calculation is not zero, an error has occurred in the LANCE, generating an interrupt to the CPU.

Received address modes – Each of the following modes compares the received destination address with a preselected value and determines whether the LANCE should accept or reject the incoming data frame.

- **Physical address mode** – The data frame is accepted when the destination address matches the station's Ethernet node address.

- **Broadcast address mode** – A destination address of all 1's specifies a broadcast address and the data frame is always accepted.

- **Logical address mode** – Accepts all data frames addressed to one type of device or set of devices. Addresses are selected for acceptance using a logical address filter as described in Section 5.5.5.

5.6.3 Register Address and Data Ports

The CPU selects the LANCE from the address bus as shown in Figure 5-4 (Section 5.6.1).

In the program address list, Addr must be a 1 to select the RDP/RAP for a word transfer. (Addr = 0 selects the LANCE CSR/BCR for word transfers.)

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

- Address [0000]$_{16}$ – Selects the register address port (RAP).

- Address [0000]$_{16}$ – Selects the register data port (RDP) and reads or writes the control status register (CSR) selected by the register address port (RAP).

Figure 5-14 shows the register address port and control status register bit formats. Except for bit 7 (STOP) the CSRs are read/write registers and are only accessible when the STOP bit is set to 1. Note 1: Note 2: other bit functions that allow it to be accessed at any time during normal operation.

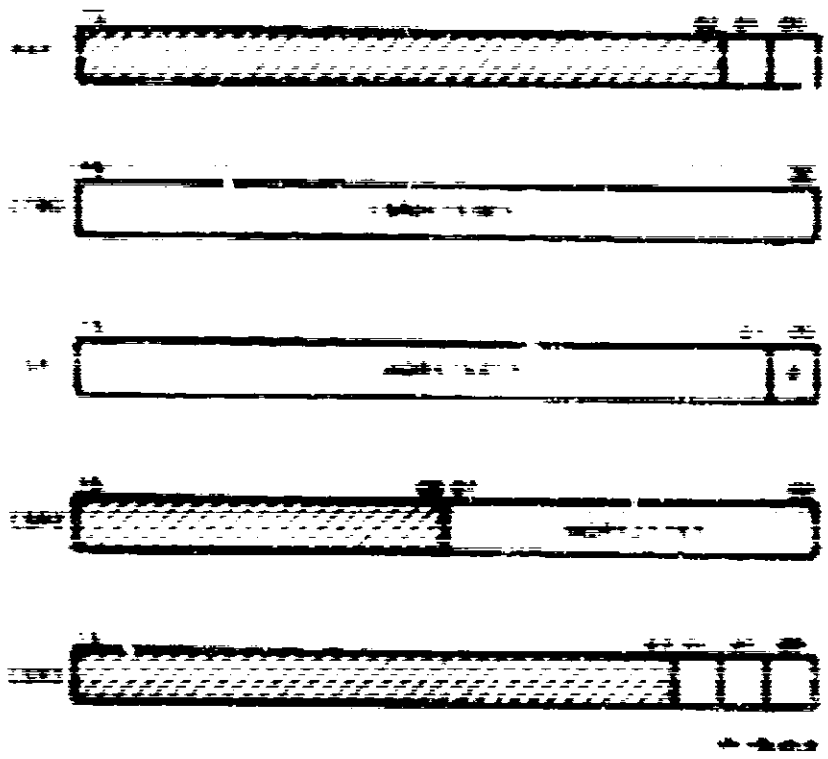


Figure 5-14 [illegible] Register Address Port (RAP) and Control/Status Register (CSR) Bit Formats

5.5.3.1 Register Address Port (RAP) and Latch – With BUS_ADR<4:1> = 0 and [illegible] the RAP register through the RAP.

The RAP is a latched input port that accepts the address asserted by the host over signals [illegible] when BUS_ADR<4:1> = 0. When a [illegible] latches and asserts the RAP<1:0> bits that select one of the four CSRs.

The RAP is a read/write register; all bits can be read or written by the program and is cleared by reset. Except for reserved bits in CSR0, the CSRs are also read/write registers.

5.3.2.2 Control/Status Register 0 (CSR0) – With address bit <A0:1> on a 0, the CPU reads 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.

CSR0 contains the control and status bits described in Table 5-24. The functions described for each bit is true when the bit is set and the bits are handled by the operating program according to the following access definitions.

- **Read/Write** – The bit can be read. It can also be set or cleared by writing a 1 or 0 to it.
- **Read-Only** – The bit can only be read. (Writing a 1 or 0 has no effect.)
- **Read/Set** – The bit is read-only but can be set by writing a 1 to it. (Writing a 0 has no effect.)
- **Read/Clear** – The bit is read-only but can be cleared by writing a 1 to it. (Writing a 0 has no effect.)
- **Write-Only** – Writing a 1 to the bit initiates a LANCE function.

Except for bit <02> (the STOP bit), all register bits are cleared by setting the STOP bit or by asserting the reset input to the chip.

Table 5-24: Control/Status Register 0 (CSR0) Bit Functions

Bits	Name	Description	Bit Type
<15>	Error (ERR)	Asserts an OR summary of error bits <14:11>. Reads as a 0 when these bits are clear.	Read-Only
<14>	Babble (BABL)	Indicates a transmit timeout error. (The transmitter was enabled longer than necessary to send a maximum data frame length of 1518 bytes.)	Read/Clear
<13>	Collision Error (CERR)	Indicates that the Ethernet transceiver did not return the collision test signal (heartbeat) within 4 μsec after completion of a normal transmission.	Read/Clear
<12>	Missed Frame (MISS)	Indicates that the receiver lost a data frame because it did not have receive buffer ownership, causing a silo overflow.	Read/Clear

Table 5-24: Control/Status Register 0 (CSR0) Bit Functions (Cont)

Bits	Name	Description	Bit Type
<11>	Memory Error (MERR)	Indicates that the LANCE did not receive LANRDY within 25.6 μsec after it asserted an address on the data bus	Read/Clear
<10>	Receive Interrupt (RINT)	Set after the LANCE updates an entry in the receive descriptor ring	Read/Clear
<09>	Transmit Interrupt (TINT)	Set after the LANCE updates an entry in the transmit descriptor ring	Read/Clear
<08>	Initialize Done (IDON)	Indicates that the LANCE has read the initialization block and stored the parameters	Read/Clear
<07>	Interrupt (INTR)	An OR summary of error and status bits <15:12,08:>	Read Only
<06>	Interrupt Enable (INEA)	Asserts LANINT when bit <07> is set	Read/Write
<05>	Receive On (RXON)	Indicates that the receiver is enabled. Reads as a 1 when bit <04> (IDON) is set if mode register bit <15> (DRX) is set. Cleared by bit <01> (STRT) and reads as a 0 until STRT is set Also cleared by the setting of bit <11> (MERR).	Read/Write
<04>	Transmit On (TXON)	Indicates that the transmitter is enabled. Set when bit <08> (IDON) is set if mode register bit <14> (DTX) is a 0. Cleared by bit <01> (STRT) and reads as a 0 until STRT is set Also cleared by the setting of bit <11> (MERR) or by the setting of bits <13> or <12> during initialization or a receive descriptor memory data error.	Read/Write
<03>	Transmit Demand (TDMD)	Setting this bit causes the LANCE to access the transmit descriptor ring from current without waiting for the polling interval	Write Only

Table 5-24: Control/Status Register 0 (CSR0) Bit Functions (Cont)

Bits	Name	Description	Bit Type
<07>	Stop (STOP)	Setting this bit causes the LANCE to halt all activity and clear the internal logic (same as reset). Also set by reset and cleared by the setting of bit <01> (STRT) or bit <00> (INIT)	Read/Set
<01>	Start (STRT)	Setting this bit enables the LANCE to perform DMA transfers.	Read/Set
<00>	Initialize (INIT)	Setting this bit causes the LANCE to access the initialize block and execute the initialization procedure.	Read/Set

5.5.3.3 Control/Status Register 1 (CSR1) – CSR1<15:01> store the lower bits of the first initialize block address as described in Table 5-25.

Table 5-25: Control/Status Register 1 (CSR1) Bit Functions

Bits	Name	Description
<15:01>	Initialize Address (IADR<15:01>)	At the beginning of a DMA address cycle, the LANCE asserts an address on IB<15:01> and stores it in the address latch.
<00>	Must Be Zero (MBZ)	Cleared and must be zero.

5.6.3.4 Control/Status Register 2 (CSR2) – CSR2<15:0> store the upper bits of the first control byte block address as described in Table 5-26.

Table 5-26: Control/Status Register 2 (CSR2) Bit Functions

Bits	Name	Description
<15:08>		Reserved
<07:00>	Initiator Address (IAD[R<23:16>)	On a DMA address cycle, the LANCE holds an address asserted on BUS <A23:16> (Bus IAD[R<16>) as used in the DP8300 or 100.

5.6.3.5 Control/Status Register 3 (CSR3) – CSR3 controls the byte swap function and some chip output pin definitions as described in Table 5-27.

NOTE

CSR1, CSR2, and CSR3 can only be read when the STOP bit in CSR0 is set. They must also be initialized after a reset or stop operation.

Table 5-27: Control/Status Register 3 (CSR3) Bit Functions

Bits	Name	Description
<15:04>		Reserved
<03>	Byte Swap (BSWP)	Set for a data buffer that starts on an odd byte address. Data bytes <15:08> are swapped with bytes <07:00> on a DMA transfer.
<02>	ALE Control (ACON)	Defines the assertion state of LANCE Address Line Enable (LAN ALE) from pin 16.
		ACON = 1: ALE is asserted high (normal) ACON = 0: ALE is asserted low
<01>	Byte Control (BCON)	Defines the Hold, Lgnorbyte, and Lxmitbyte logic levels outputs from the following pins:

	Pin 17	Pin 18	Pin 19
BCON = 0	LANHLD	HLDRQ	HLDRQ (inverse)
BCON = 1	HLDRQ	BYTE	HLDRQ

6.9.4 Buffer Management Protocol

The LAT operating software maintains data buffer areas in memory for the temporary storage of Ethernet data that passes through the LANCE. It also maintains tables in memory which the LANCE retrieves (under DMA transfers) and uses to access the data buffers (under DMA transfers).

The LANCE buffer management protocol uses the following table structures in managing the data table and buffer areas in memory:

- Data buffers – The receive and transmit data buffers occupy contiguous memory space and may 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 word (two-byte) boundary.

- Descriptor rings – The receive and transmit descriptor rings are tables in memory that the LANCE uses to access the receive and transmit data buffers in memory. Each descriptor ring consists of an input one entry and starts on a quadword (eight-byte) boundary.

  The descriptor rings provide the information path between the LANCE and the operating program for filling or emptying a data buffer in memory.

- Message descriptors – Each descriptor ring entry consists of four words called message descriptors. These supply the following information to access one buffer:

  Points to the base address of the buffer

  Specifies the length of the buffer

  Specifies how the data frame is to be processed

  Specifies LANCE action on an error or change in status

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

Network operations begin when the operating program sets the LANCE register CSR0 bit 0 (INIT bit). The LANCE retrieves the initialize block from memory, starting at the address specified in CSR1 and CSR2, and stores the information in its internal parameter registers. It can then begin transfer operations when the operating program assembles the receive and transmit descriptor rings in memory.

[redacted]

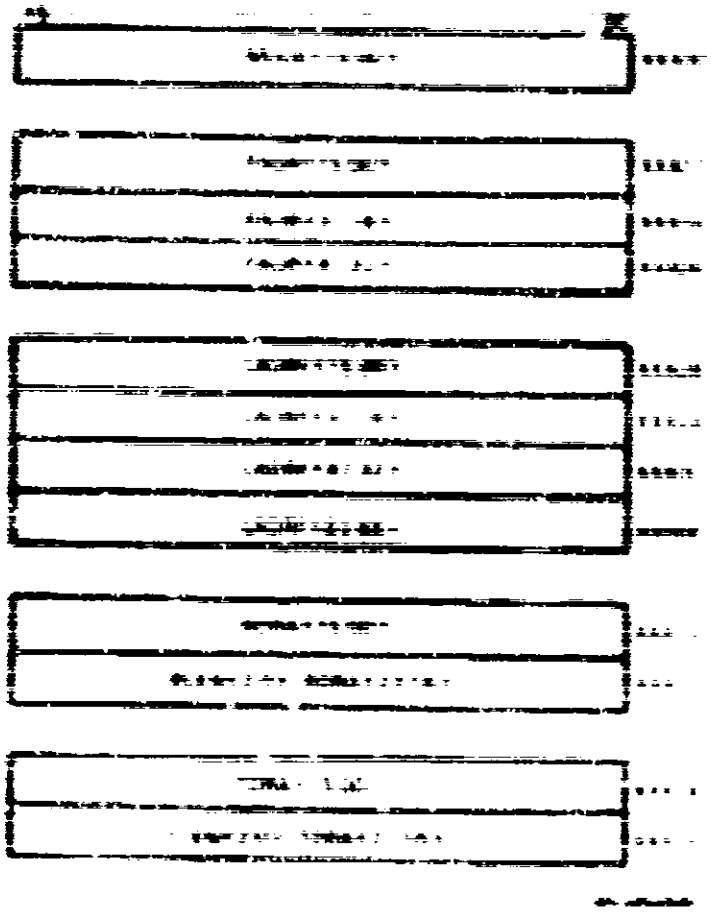


Figure 5-15 [redacted]

The server software assembles the setup block before initializing the LANCE. When it is entered, the LANCE performs a DMA cycle that retrieves the following initializing parameter from the initialize block and stores them in its internal registers.

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

Basic addressing - The LANCE uses the physical address (PADR) register and logical address filter (LADRF) register to determine whether to accept or reject incoming data frames.

- Physical address - If the first bit of the incoming destination address is a 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 a 1 for the broadcast address because it is also a logical address that must be enabled. A destination address of all 1s is the broadcast address and the data frame is always accepted.

- Logical address - For all other address modes with the first bit set to a 1, data frames are accepted according to the LADRF register. There is an algorithm filter that requires the operating software to do the hash filtering.

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

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

Minimum data frame length - An incoming message must have a byte count of at least 64 bytes to be considered a valid data frame. Otherwise, the LANCE does not update the receive message descriptor pointing to the buffer. The contents are saved for the next data frame.

[illegible] are clear during normal operation. They allow the LAN/E operating parameters to be changed for maintenance functions as shown in Table 5-28.

Table 5-28. [illegible] Register Bit Functions

Bits	Name	Description
[illegible]	[illegible] (FRAM)	With FRAM set, the LAN/E accepts all incoming data frames regardless of the destination address.
[illegible]	[illegible]	Reserved.
[illegible]	[illegible] (EXT)	Valid only if the [illegible] is set. EXT allows the LAN/E loopback path. EXT = 0 External loopback EXT = 1 Internal loopback With EXT set, the data buffer can be looped on the on-board [illegible]. The normal data [illegible] of the [illegible] is [illegible].
[illegible]	[illegible]	When set, the LAN/E accepts only one [illegible]. A [illegible] on the [illegible] sets error status (ERR), but [illegible] a subsequent one-shot [illegible].
[illegible]	[illegible]	Valid only in the external loopback mode (EXT is set) and [illegible] the collision logic. The LAN/E enters its transmission attempts then sets retry error (RTE), but [illegible] subsequent transmissions.
[illegible]	[illegible] (DTR)	With DTR clear, the transmitter generates the CRC and appends it to the data frame. (In a loopback test, with DTR clear, the receiver checks that the CRC is written to memory after it would be checked in the program. With DTR set, the CRC logic is used in the receiver (in a loopback test with DTR set, the program generates and appends the CRC. The received CRC [illegible] and [illegible] are stored.

Table 5-28 Mode (MODE) Register Bit Functions (Cont)

Bits	Name	Description
<0>	Loopback (LOOP)	Sets the LANCE operation in the diagnostic loopback mode (active high) parameter. Because of the 64-byte maximum-to-be-followed data frame, the maximum data frame size is disabled. The LANCE waits until the entire message is in the FIFO before transmitting it. The incoming data between the outgoing data and is not passed to memory until the comparisons is complete. (Data chaining is not possible in this mode.)
<01>	Disable Transmitter (DTX)	When set, disables LANCE. ... the transmit ... ring. (Same as <16> (TACK).) If the disabled and transmit are not attempted.
<02>	Disable Receiver (DRX)	When set, disables LANCE access to the receive descriptor ring. (Same as <05> (RACK).) If the disabled and all receive data frames are rejected.

5.5.5.2 Physical Address (PADR) Register – PADR <47:00> store the individual ... node address that was assembled from the PA PROM. PADR <00> must be 0.

5.5.5.3 Logical Address Filter (LADRF) Register – LADRF ... accepted corresponding logical addresses.

5.5.5.4 Receive Descriptor Ring Address (RDRA) Register – RDRA ... functions described in Table 5-29.

Table 6-23 Receive Descriptor Ring Address (RDRA) Register Bit Functions

Bits	Name	Description
<31:28>	Receive Ring Length (RLEN)	The three high-order bits specify the number of receive descriptor ring entries as a power of two: RLEN Bits <31:29> — Number of Entries 0 — 1 1 — 2 2 — 4 3 — 8 4 — 16 5 — 32 6 — 64 7 — 128
<28:24>		Reserved
<23:00>	Receive Ring Address	Specifies the receive descriptor ring base address.
<63:32>	Reserved	Reserved; ignored on write. Returns zeros on read.

8.5.5.2 Transmit Descriptor Ring Address (TDBA) Register – TDBA <31:0>

Table 8-35 Transmit Descriptor Ring Address (TDBA) Register DR Parameters

Size	Symbol	Description
<31:0>	Transmit Ring Length (TDBA<31:0>)	The value in this register specifies the number of bytes in the receive ring. Software writes to a packet of the:
		TDBA Bits <20> / Number of Entries
		0 / 1
		1 / 2
		2 / 4
		3 / 8
		4 / 16
		5 / 32
		6 / 64
		7 / 128
[illegible]		[illegible]
[illegible]	Transmit Ring Valid (TDBA<20>)	[illegible]
[illegible]	Must be Zeros (MBZ)	[illegible]

8.5.6 Receive Descriptor Ring

A receive descriptor ring consists of at least one entry as shown in the format shown in Figure [illegible]. Each entry points to a receive data buffer and defines how the data is to be operated.

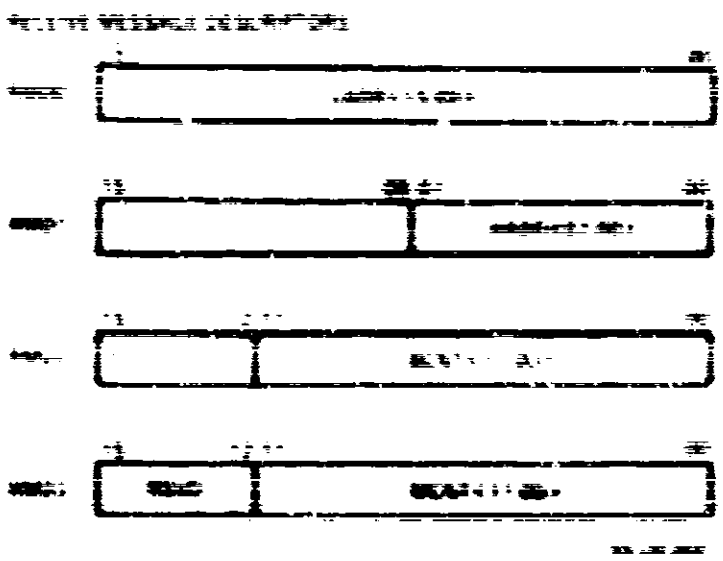


Figure 6-10 LANCE Receive Descriptor Ring Entry

Each entry consists of four words called receive message descriptors (RMDs) and is accessed on a quadword (eight byte) boundary. The following tables describe the bit functions for each descriptor.

- Receive Message Descriptor 0 (RMD0) is described in Table 6-31
- Receive Message Descriptor 1 (RMD1) is described in Table 6-32
- Receive Message Descriptor 2 (RMD2) is described in Table 6-33
- Receive Message Descriptor 3 (RMD3) is described in Table 6-34

Table 6-31 Receive Message Descriptor 0 (RMD0) Bit Functions

Bits	Name	Description
[illegible]	Low Address of Buffer (Write)	[illegible] the low order host address bits for the receive data buffer selected by this descriptor ring entry. Written by the program and is not changed by the LANCE

Table 8-32 Receive Message Descriptor 1 (RMD1) Bit Functions

Bits	Items	Description
<15>	Ownership (OWN)	Indicates the ownership of this entry. 0 = Owned by the server software 1 = Owned by the LANCE The LANCE clears the OWN bit after filling the receive data buffer. The program sets it after reading the data completing the buffer. Buffer ownership can only be given up, it can not be taken. **NOTE** The following status bits <14-08> are set by the LANCE and are cleared by the program.
<14>	Error (ERR)	An OR summary of error status bits <13-10>
<13>	Framing Error (FRAM)	Indicates that the received data frame contains both a [illegible] eight bits and a CRC error.
<12>	Overflow (OFLO)	Indicates that some part of the received data frame was lost because the LANCE was not able to access the buffer and store the data before the SILO buffer overflowed.
<11>	Cyclic Redundancy Check (CRC)	Indicates that a CRC error was detected on the received data frame.
<10>	Buffer Error (BUFF)	Indicates that the LANCE has used all allocated buffers or did not acquire the next entry status in time while chaining a received data frame. (OFLO) is also set because the LANCE cannot own it to acquire the next buffer until the SILO overflows.)

Table 6-32 Receive Message Descriptor 1 (RMD ., Bit Functions (Cont)

Bits	Name	Description
<09>	Start of Frame (STP)	Indicates that this entry contains the pointer to the first data buffer for the frame. (Used when chaining the data buffers.)
<08>	End of Frame (ENP)	Indicates that this entry contains the pointer to the last data buffer for the frame. (Used when chaining the data buffers.) With STP and ENP both set in the same entry, the data frame fits into one buffer and no chaining is in effect.
<07:00>	High Address (HADR <07-00>)	Contains the high-order base address bits for the receive data buffer area selected by this entry. It is written by the program and is not changed by the LANCE.

Table 6-33 Receive Message Descriptor 2 (RMD2) Bit Functions

Bits	Name	Description
<15:12>	Must be Ones (MBO)	This field is written by the program and is not changed by the LANCE.
<11:00>	Buffer Byte Count (BCNT <11:00>)	Receive buffer length in bytes. BCNT is written by the program and is not changed by the LANCE.

Table 6-34 Receive Message Descriptor 3 (RMD3) Bit Functions

Bits	Name	Description
<15:12>		Reserved and read as zeros.
<11:00>	Message Byte Count (MCNT <11:00>)	Receive message length in bytes. MCNT is written by the LANCE and is used by the program. It is valid only when RMD1 bit <7> (OWN) is clear and bit <6> (ERR) is not set.

5.5.7 Transmit Descriptor Ring

A transmit descriptor ring consists of at least one entry in memory in the format shown in Figure 5-17. Each entry points to a transmit data buffer and defines how the data is to be handled.

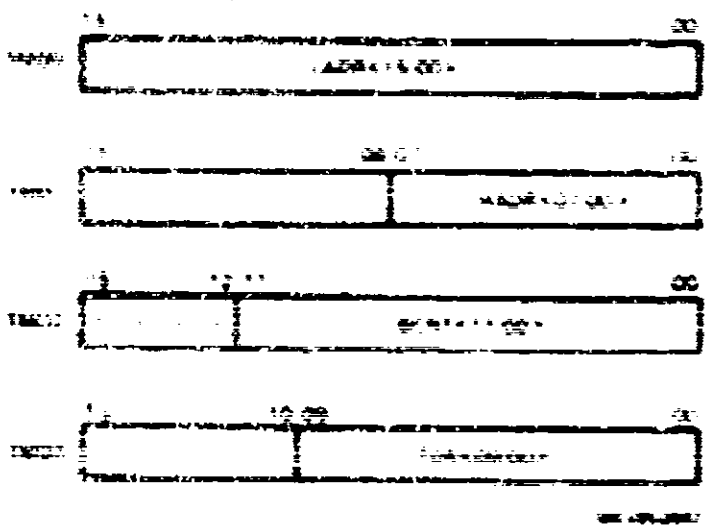


Figure 5-17 LANCE Transmit Descriptor Ring Entry

Each entry consists of four words called transmit message descriptors (TMDs) and is accessed on a quadword (eight byte) boundary. The following tables describe the bit functions for each descriptor.

- Transmit Message Descriptor 0 (TMD0) is described in Table 5-15.
- Transmit Message Descriptor 1 (TMD1) is described in Table 5-16.
- Transmit Message Descriptor 2 (TMD2) is described in Table 5-17.
- Transmit Message Descriptor 3 (TMD3) is described in Table 5-18.

Table 5-25: Transmit Message Descriptor 0 (TMD0) Bit Functions

Bits	Name	Description
<15:00>	Low Address (LADR<15:00>)	Contains the low-order byte address bits for the transmit data buffer area selected by this entry. (Written by the program and not changed by the LANCE.)

Table 5-26: Transmit Message Descriptor 1 (TMD1) Bit Functions

Bits	Name	Description
<15>	Ownership (OWN)	Indicates the current ownership of the entry. 0 = Owned by the server software 1 = Owned by the LANCE The program sets the OWN bit after filling the transmit data buffer. The LANCE clears it after transmitting the data (emptying the buffer). Buffer ownership can only be given up and cannot be taken. **NOTE** The following status bits (ERR, LCOL, LCAR, and RTRY) are set by the LANCE and are cleared by the program.
<14>	Error (ERR)	An OR summary of TMD2 status bits (BUFF, UFLO, LCOL, LCAR, and RTRY) set by the LANCE and cleared by the program.
<13>	Reserved	The LANCE writes this bit as a zero.
<12>	More Retries (MORE)	Indicates that more than one retry was necessary to transmit the data frame.
<11>	One Retry (ONE)	Indicates that the transmission completed successfully after one retry.
<10>	Deferred (DEF)	Indicates that the LANCE has deferred its transmission because of existing Ethernet traffic. (The carrier sense input was asserted.)

Table 5-36 Transmit Message Descriptor 1 (TMD1) Bit Functions (Cont)

Bits	Name	Description
<31>	Start of Frame (STP)	Indicates that this entry contains the pointer to the first data buffer for the frame used when chaining the buffers.
<30>	End of Frame (ENP)	Indicates that this entry contains the pointer to the last data buffer for the frame used when chaining the buffers. With STP and ENP both set in the same entry, the data fits into one buffer and no chaining is in effect.
<23:16>	High Address (HADR <23:16>)	Contains the high-order byte address bits for the transmit data buffer area selected by this entry (written by the program and not changed by the LANCE).

Table 5-37 Transmit Message Descriptor 2 (TMD2) Bit Functions

Bits	Name	Description
<31:16>	Must be Zero (BCNT)	This field is written by the program and is not changed by the LANCE.
<15:00>	Buffer Byte Count (BCNT <15:00>)	Transmit buffer length in bytes. The two's complement of the BCNT value is written by the program and is not changed by the LANCE.

Table 6-20 Transmit Message Descriptor 3 (TMD3) Bit Functions

Bits	Name	Description
<15>	Buffer Error (BUFF)	Set by a LANCE when the ENP flag (TMD1 bit <8>) is not set for the current buffer and the LANCE does not own the next buffer (TMD1 bit <15> is set) because the LANCE owned the next buffer before data from the CPU was empty.
<14>	Collision Error (CERR)	Indicates that data was lost because the silo underflowed before the end of the data frame and further data was not requested in time.
<13>	Reserved	The LANCE writes this bit as a zero. **NOTE** Bits TMD3 <12:0> are set by the LANCE and are cleared by the program.
<12>	Late Collision (LCOL)	Indicates that a collision occurred after the minimum slot time. A retry is not attempted.
<11>	Loss of Carrier (LCAR)	Indicates that the carrier presence was lost during a LANCE unaided transmission. A retry is not attempted.
<10>	Retry Error (RTRY)	Indicates that the LANCE made 16 unsuccessful attempts to transmit because of repeated collisions. With RTRY set, the register bits TMD3 <9:0> are the same as TMD3 <15>. RTRY is set after the sixteenth attempt.
<9:0>	Time Domain Reflectometry (TDR)	Counts the LANCE transmit clocks from the start of a transmission to a collision detection. The resulting value is useful in determining the approximate distance to a fault on the Ethernet coaxial cable but is not valid if RTRY is set.

CHAPTER 6

6
Hardware Description

6.1 General

The DEC server 100 Terminal Server is a stand-alone communications device. Table top mounted, it requires a minimum of 6 inches clearance on all sides for proper ventilation, with sufficient clearance in the back for cable and service access.

For site preparation and installation information, refer to the *DEC server 100 Terminal Server Site Preparation/Hardware Installation Guide*.

6.2 Features

6.2.1 Hardware Options

The DEC server 100 and its supporting hardware are available under the options designations listed in Table 6–1.

Table 6-1 DECserver 100 Hardware Options

Option	Designation
DECserver 100 Terminal Server (100 - 120 Vac)	DSRVA-AA
DECserver 100 Terminal Server (220 - 240 Vac)	DSRVA-AP
DECserver 100 Country Kits:	
U.S.A. and Canada (English) Canada (French)	DSRVK-AA DSRVK-AC
United Kingdom	DSRVK-AE
Germany	DSRVK-AG
Japan	DSRVK-AJ
France	DSRVK-AF
Spain	DSRVK-AS
Australia and New Zealand	DSRVK-AZ
Ethernet Connector	H8571-A
Ethernet Loopback Connector	H8571-J
Terminal Loopback Connector	[illegible]

6.2.2 Approved Terminals

The Digital Equipment Corporation terminals listed in Table 6-2 are supported by the terminal server at transfer speeds up to 19200 baud in the interactive terminal mode and up to 38400 baud in flow controlled block mode or the transfer mode.

Table 6-2: Approved DIGITAL Terminals

Hardcopy Terminals

LA12, LA34, LA38
LA120, LA180
LA34, LA36, LA38

Video Display Terminals

VT50, VT52, VT5x-VTxx in VT52 mode
VT100, VT101, VT102, VT125, VT131
VTxxx, VTxxx, VTxxx

Intelligent Terminals/Terminal Controllers

[illegible] or VTxxx in VTxxx mode
RAINBOW 100, DECmate II
PROFESSIONAL 300 in VT100 emulation mode

6.2.3 Rear Panel Elements

Table 6-3 describes the functions of the LED indicators, cable connectors and switches located on the rear panel. Refer to Chapter 3 for information on the possible error conditions signalled by the LED indicators.

Table 6-3 Rear Panel Element Functions

Element	Function
LED indicator state	
ON	The self test sets the indicator to the on state if it successfully passed all diagnostic tests. It then calls the initialize program to request a download load of the LAT operating software from a Phase IV DECserver
	If the load is successful, the LAT operating software then verifies its checksum and the server is operational
Blinking	If the self test detects one or more nonfatal diagnostic errors, it sets the indicator to the blinking state. It then calls the initialize program and attempts a network load operation. The load of the server image may start automatically or by user operation
OFF	If power is applied and the self test detects one fatal error, it leaves the indicator in the off state and halts. The server does not attempt to load
Network [illegible] Switch	Sets [illegible] as described in [illegible]
AUI Power Fault Recovery	Accepts the AUI recovery. A current sensing power circuit is provided on the [illegible] AUI port
AUI Voltage Input Selection Switch	[illegible]
AUI Low Power Mode	The [illegible]
Transceiver Cable Connector	Accepts the transceiver cable connector. The other end of the cable plugs into an H4000-[illegible] AUI transceiver. The [illegible] must [illegible]
Terminal Connection [illegible]	Port [illegible]

6.3 Specifications

The terminal server module and power supply are mounted inside a metal case that provides electromagnetic interference (EMI) shielding. The case is enclosed in a plastic case

6.3.1 [illegible]

[illegible] shows the terminal server's dimensions. Table 6–4 lists the dimensions and weight.

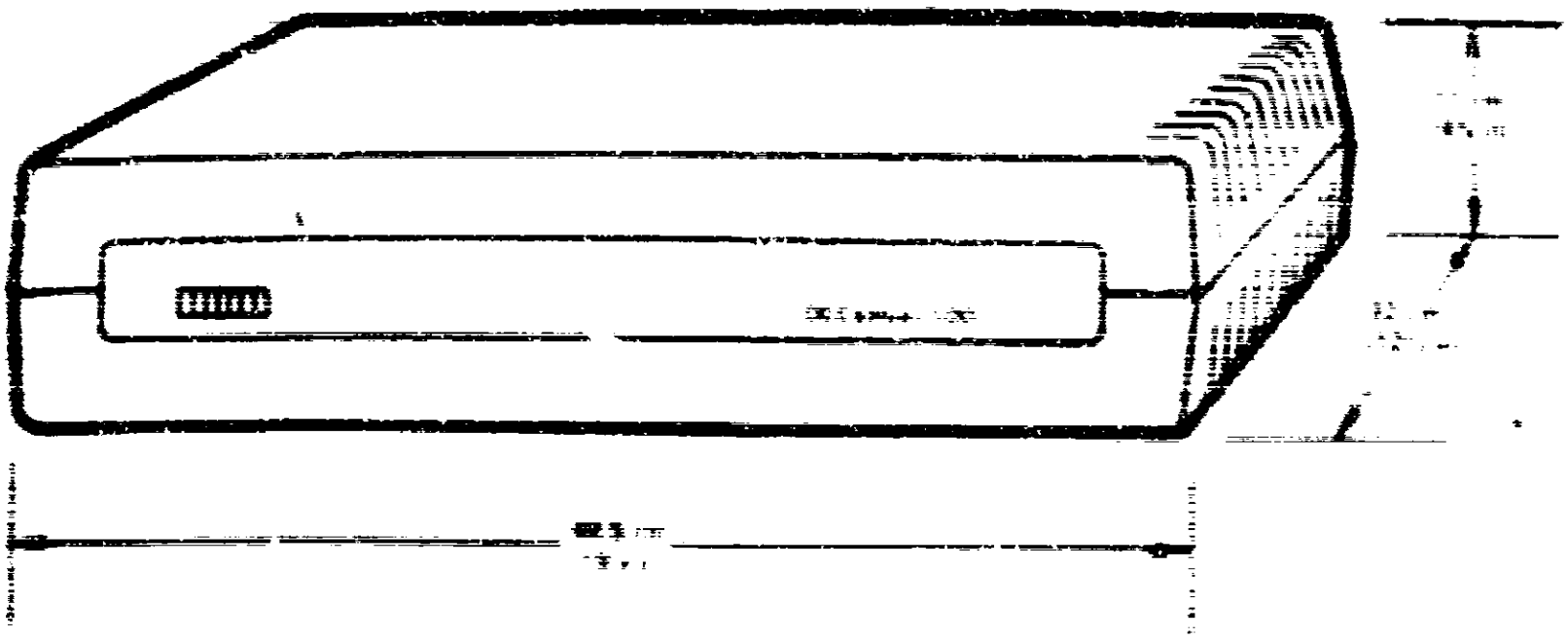


Figure 6–1 DECserver 100 Terminal Server Dimensions

Table 6–4 Physical Specifications

Item	Dimensions and Weight
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

6.3.2 [illegible]

[illegible]

Table 6-5. Electrical Specifications

Item	Specification
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

6.4 Data Cable Connectors

6.4.1 Terminal Connector Pin Assignments

Table 6-6 lists the pin assignments for each of the 25-pin D-subminiature (D-Sub) terminal connectors on the back of the terminal server.

Table 6-6 Terminal Connector Pin Assignments

Pin No.	Signal	Function
1	[illegible]	[illegible]
2	[illegible]	[illegible]
3	[illegible]	[illegible]

6.4.2 Ethernet Transceiver Connector Pin Assignments

The Ethernet transceiver cable contains four twisted pairs (three differential signal pairs and one power pair) that are separately shielded inside a shielded outer cover.

The pin assignments for the 15-pin Ethernet transceiver cable connector are listed in Table 6-7.

Table 6-7 Ethernet Transceiver Connector Pin Assignments

Pin No.	Signal	Function
1	PEM XCVR CABLE SHIELD	Outer cable shield — signal ground
2	PEM COLL L	Collision presence (–)
3	PEM COLL H	Collision presence (+)
4	PEM TX DATA L	Transmit data (–)
5	PEM TX DATA H	Transmit data (+)
6	PEM RX DATA L	Receive data (–)
7	PEM RX DATA H	Receive data (+)
10	+12V	+12 V supply to transceiver
11	+12V RET	+12 V return to power ground

6.5 Data Cables

6.5.1 Transceiver Connector Cables

Table 6-8 lists the available types and lengths of transceiver connector cables. There are shielded cables with straight (through BC22E) and connectors.

- The BC21E-xx transceiver cable is used for direct connection to a host interface.
- The BC22E-xx transceiver cable connects the terminal server to a multiport cable from an existing transceiver.

Table 6-8 Yamaichi Connector Cables

Description	Part Number
Multi-Mode Cables	
3.3 m (10 ft)	[illegible]
7.6 m (25 ft)	[illegible]
15.2 m (50 ft)	[illegible]
30.5 m (100 ft)	[illegible]
61.0 m (200 ft)	[illegible]
91.5 m (300 ft)	[illegible]
Attachment Cables	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

6.4.2 Ethernet Transceiver Connector Cables

Table 6-9 lists the available types and lengths of Ethernet Transceiver connector cables.

- Cables are available with PVC or Teflon™ insulation and with straight through or right angle end connectors.
- Teflon insulated cables must be used in environmental air spaces such as air (plenum) ceilings.
- UL94 cables are PVC insulated but are smaller in diameter and more flexible for use in office areas.

Teflon is a registered trademark of [illegible]

Table 6-9 Ethernet Transceiver Connector Cables

Description	Part Number
Straight through connectors, [illegible]	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
Right angle [illegible]	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
Straight through [illegible]	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
Right angle [illegible]	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	
[illegible]	[illegible]
[illegible]	[illegible]

INDEX

A

B

C

D

E

F

H

I

U

W

X

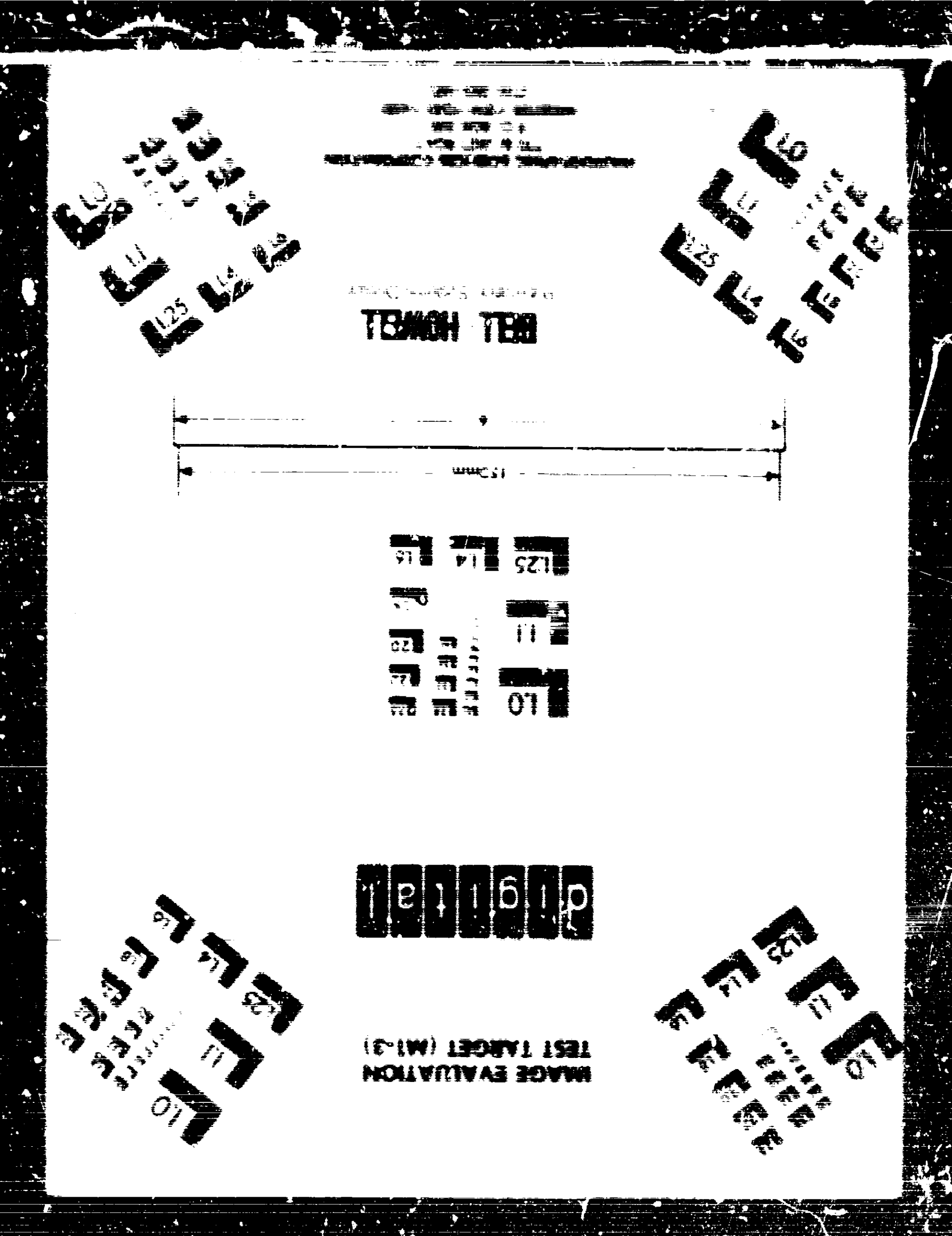


IMAGE EVALUATION
TEST TARGET (MT-3)