DECchip 21040 Ethernet LAN Controller for PCI

Hardware Reference Manual

Order Number: EC–N0752–72

Revision/Update Information: This is a new manual.

Digital Equipment Corporation
Maynard, Massachusetts

May 1994

This document was prepared using VAX DOCUMENT, Version 2.0.

Contents

4 Host Communication Area

5 Functional Description

6 Host Bus Operation

7 Network Interface

A Joint Test Action Group Test Logic

B DNA CSMA/CD Counters and Events Support

C Hash C Routine

D Technical Support, Ordering, and Associated Literature

Index

Examples

Figures

Tables

Preface

Purpose and Audience

The *DECchip 21040 Ethernet LAN Controller for PCI Hardware Reference Manual* describes the operation of the DECchip 21040 Ethernet LAN Controller for PCI (also referred to as the 21040). This manual is for system designers who use the 21040.

Manual Organization

This manual contains seven chapters, four appendices, and an index.

- Chapter 1, Introduction, includes a general description of the 21040. It also provides an overview of the 21040 hardware components.
- Chapter 2, Signal Descriptions and Bus Commands, provides the physical layout of the 21040 and describes each of the input and output signals.
- Chapter 3, Registers, provides a complete bit description of the 21040 command and status registers as well as the configuration registers.
- Chapter 4, Host Communication Area, describes how the 21040 communicates with the host using descriptor lists and data buffers.
- Chapter 5, Functional Description, describes reset commands, interrupt handling, and startup. It also describes the transmit and receive processes.
- Chapter 6, Host Bus Operation, provides a description of the read, write, and termination cycles.
- Chapter 7, Network Interface, describes the 10BASE-T and AUI interfaces. It includes a complete description of media access control operations. It also provides detailed transmitting and receiving operation information.
- Appendix A, Joint Test Action Group Test Logic, provides descriptions of the testing, observing, and modifying circuit activity during normal operation.

- Appendix B, DNA CSMA/CD Counters and Events Support, describes features that support the driver in implementing and reporting the specified counters and events.
- Appendix C, Hash C Routine, provides an example of a C routine that generates a hash index for a given Ethernet address.
- Appendix D, Technical Support, Ordering, and Associated Literature, contains information about technical support as well as ordering parts and related documentation.
- The index provides an alphabetical list of topics described in this manual. An entry with an f appended to the page number (for example, 21040 pinout diagram, 2-2f) indicates a figure reference. An entry with a t appended to the page number (for example, Twisted-pair compensation behavior, 3-60t) indicates a table reference.

Document Conventions

The values 1, 0, and X are used in some tables. X signifies a don't care (1 or 0) convention, which can be determined by the system designer.

1 Introduction

This chapter provides a general description of the DECchip 21040 Ethernet LAN Controller, its features and an overview of the hardware.

1.1 General Description

The DECchip 21040 is an Ethernet LAN controller that is based on the peripheral component interconnect (PCI) local bus. The 21040 provides a glueless connection to the PCI.

During host interface operation, the 21040 interfaces with the processor using on-chip command and status registers (CSRs) and a shared host memory area, set up mainly during initialization. This minimizes processor involvement in the 21040 operation during normal reception and transmission. Bus traffic is also minimized by filtering out received runt frames and by automatically transmitting collided frames again without a repeated fetch from host memory.

During communication interface operation, the 21040 provides both an attachment unit interface (AUI) and a twisted-pair interface, enabling a low chip count connection to the two most popular Ethernet interfaces. The 21040 can sustain transmission or reception of minimal-sized back-to-back packets at full line speed with a 9.6-microsecond interpacket gap. The 21040 can also function in a full-duplex environment.

1.2 Features

The 21040 has the following features:

- Offers a single-chip Ethernet controller for PCI local bus
  - Provides glueless connection to PCI bus
  - Contains on-chip integrated attachment unit interface (AUI) port and a 10BASE-T transceiver
- Supports full-duplex operation

- Provides clock speed up to 33 megahertz, with no wait states on PCI master operation
- Enables powerful on-chip DMA with programmable, unlimited burst size providing for low CPU utilization
- Implements unique, patent-pending, intelligent arbitration between DMA channels that prevent underflow or overflow and are optimized for full-duplex operation
- Contains two large (256-byte) independent receive and transmit FIFOs
- Supports either big or little endian byte ordering
- Implements joint test action group (JTAG) compatible test access port with boundary-scan pins
- Provides full support of IEEE 802.3, ANSI 8802-3, and Ethernet standards
- Offers a unique, patented solution to Ethernet capture-effect problem
- Contains a variety of flexible address filtering modes
  - 16 perfect addresses
  - 512 hash-filtered multicast addresses and one perfect address
  - 512 hash-filtered physical addresses and multicast addresses
  - Inverse perfect filtering
- Provides serial ROM interface for Ethernet ID address ROM
- Supports three LEDs: network activity, LinkPass, and AUI/10BASE-T
- Enables automatic detection and correction of 10BASE-T receive polarity
- Provides external and internal loopback capability
- Implements low power, 3.3-volts complimentary metal oxide semiconductor (CMOS) device; interfaces to 5.0-volt or 3.3-volt logic

1.3 Hardware Overview

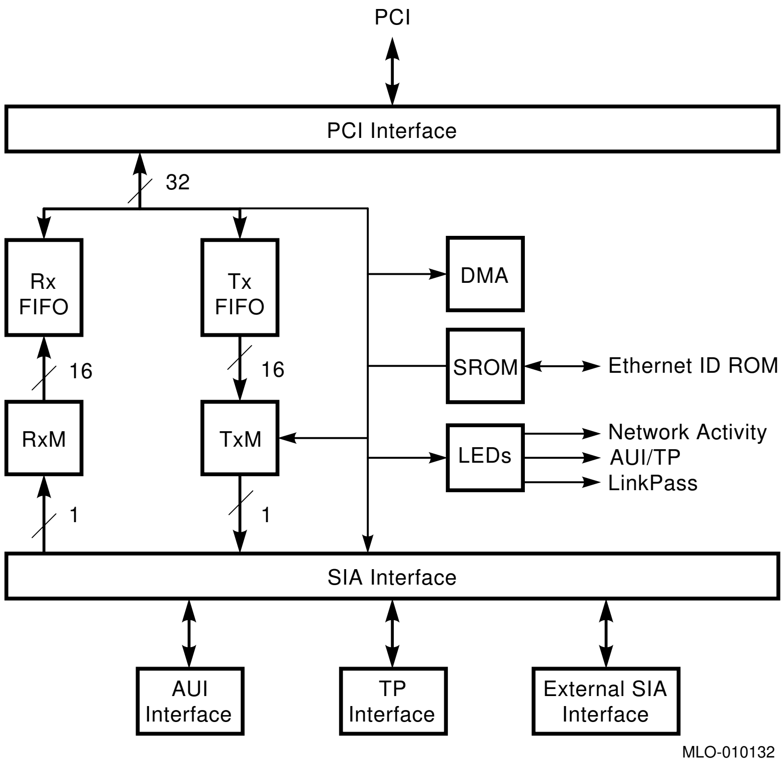
The following list describes the 21040 hardware components, and Figure 1–1 shows a block diagram of the 21040.

- PCI Interface—Includes all interface functions to the PCI bus; handles all interconnect control signals, and executes PCI direct memory access (DMA) and I/O transactions.
- DMA—Contains dual receive and transmit controller; supports bursts of up to 32 longwords; handles data transfers between PCI memory and on-chip memory.
- FIFOs—Contains dual 256-byte FIFOs for receive and transmit; supports automatic packet deletion (runt packets or after a collision) and packet re-transmission after a collision on transmit.
- TxM—Handles all CSMA/CD[1] MAC[2] transmit operations and transfers data from transmit FIFO to the serial interface attachment (SIA) for transmission.
- RxM—Handles all CSMA/CD receive operations and transfers the data from the SIA to the receive FIFO.
- SIA—Performs physical layer operations; implements the AUI and 10BASE-T functions, including the Manchester encoder and decoder functions.

---

1 Carrier-sense multiple access with collision detection
2 Media access control

Figure 1–1 DECchip 21040 Block Diagram



2

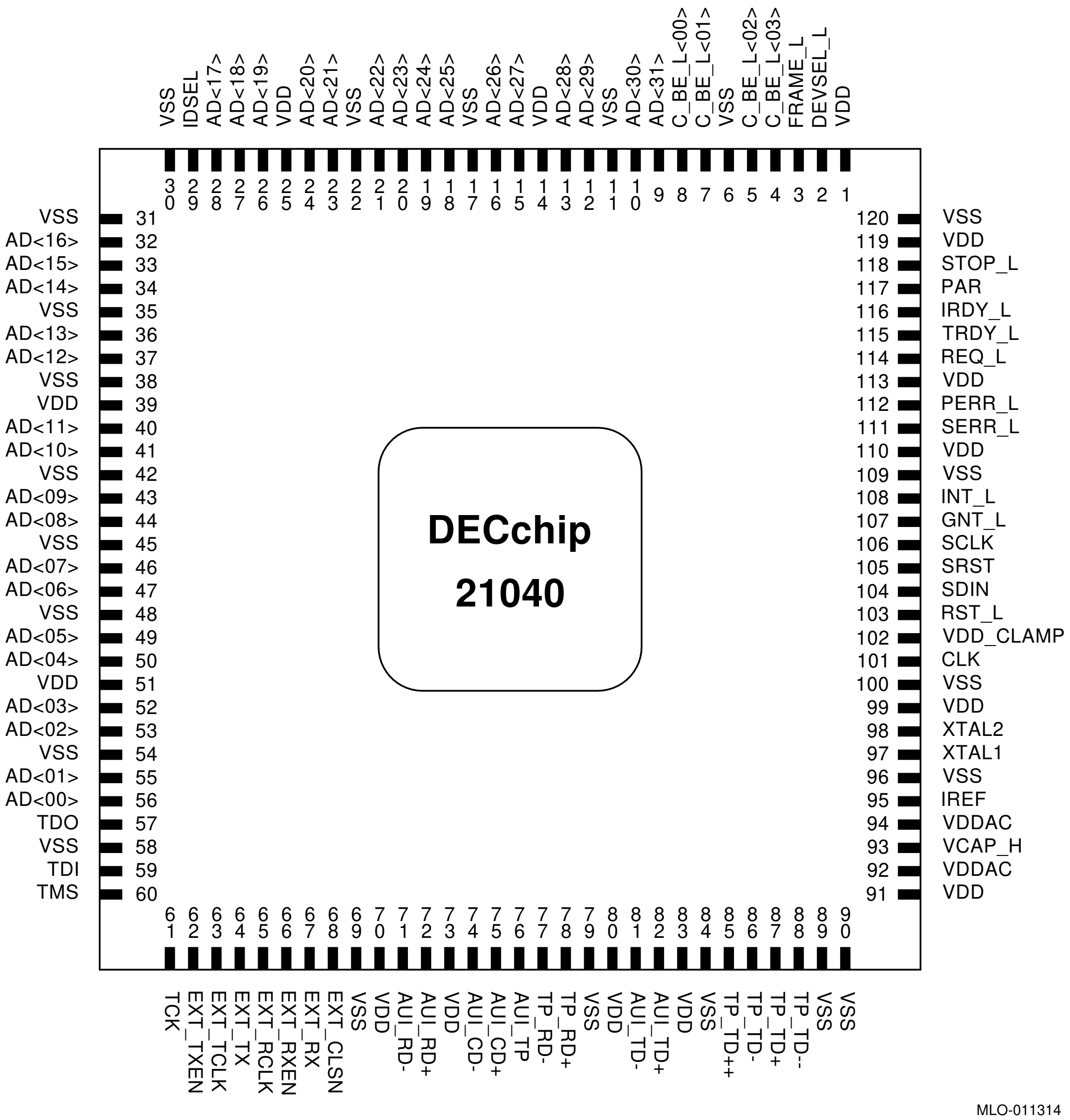
Signal Descriptions and Bus Commands

This chapter describes the 21040 signals and lists the bus commands.

2.1 21040 Pinout

The 21040 is housed in the 120-pin plastic quad flat pack. The 21040 uses all pins. Figure 2–1 shows the 21040 pinout.

Figure 2–1 DECchip 21040 Pinout Diagram (Top View)



2.2 Signal Descriptions

Table 2–1 provides a description of each of the signals used by the 21040. These signals are listed alphabetically.

The following terms describe the 21040 pinout.

- **Address phase**

  Address and appropriate bus command are driven during this cycle.

- **Data phase**

  Data and the appropriate byte enable code are driven during this cycle.

- **_L**

  All pin names with the _L suffix are only asserted low.

Note

The following abbreviations are used in the tables in this section.

I = Input
O = Output
I/O = Input/output
O/D = Open drain

Table 2–1 Signal Pin Reference

Signal	Type	Description
AD<31:00>	I/O	32-bit multiplexed PCI address and data lines. Address and data bits are multiplexed on the same pins. During the first clock cycle of a transaction, AD<31:00> contains a physical byte address (32 bits). During subsequent clock cycles, AD<31:00> contains data. A 21040 bus transaction consists of an address phase followed by one or more data phases. The 21040 supports both read and write bursts. Little and big endian byte ordering can be used.
AUI_CD–	I	Attachment unit interface receive collision differential negative data.
AUI_CD+	I	Attachment unit interface receive collision differential positive data.

(continued on next page)

Table 2–1 (Cont.) Signal Pin Reference

Signal	Type	Description
AUI_RD–	I	Attachment unit interface receive differential negative data.
AUI_RD+	I	Attachment unit interface receive differential positive data.
AUI_TD–	O	Attachment unit interface transmit differential negative data.
AUI_TD+	O	Attachment unit interface transmit differential positive data.
AUI_TP	I	Attachment unit interface and twisted-pair select line. When asserted high, the attachment unit interface is selected. When asserted low, the twisted-pair interface is selected. Software can override the pin selection (Section 3.2.2.3).
C_BE_L<03:00>	I/O	Bits 0 through 3 of the bus command and byte enable lines. Bus command and byte enable are multiplexed on the same PCI pins. Table 2–2 lists the bus commands. During the address phase of the transaction, C_BE_L<03:00> provide the bus command. During the data phase, C_BE_L<03:00> provide the byte enable. The byte enable determines which byte lines carry valid data. For example, C_BE_L<00> applies to byte 0, and C_BE_L<03> applies to byte 3. In all master and I/O operations, C_BE_L<03:00> contain a value equal to a longword hexadecimal value of 0. In configuration operations, C_BE_L<03:00> can contain any value; 21040 supports byte, word, and longword operations.
CLK	I	The clock provides the timing for the 21040–related bus transactions. All the other bus signals are sampled on the rising edge of CLK. The clock range is between 16 megahertz and 33 megahertz.
DEVSEL_L	I/O	Device select is asserted when it is the target of the current bus access. When the 21040 is the initiator of the current bus access, it expects the target to assert DEVSEL_L within 5 bus cycles, confirming the access. To accomplish this, the 21040 asserts this signal in a medium speed (within 2 bus cycles). If the target does not assert DEVSEL_L within the required bus cycles, the 21040 aborts the cycle.

(continued on next page)

Table 2–1 (Cont.) Signal Pin Reference

Signal	Type	Description
EXT_CLSN	I/O	Collision detect or test signals a collision occurrence on the Ethernet cable to the 21040. It may be asserted and deasserted asynchronously by the external SIA to the receive clock. This signal is an output to the AUI/TP LED. The LED is on when AUI is selected. This pin can be used for SIA testing features.
EXT_RCLK	I/O	Receive clock or test pin carries the recovered receive clock supplied by an external SIA. During idle periods, the RCLK pin may be inactive. This pin can be used for SIA testing features.
EXT_RX	I/O	Receive data or test pin carries the input receive data from the external SIA. The incoming data should be synchronous with the RCLK signal. This pin also outputs to the LinkPass LED. In 10BASE-T mode, when LinkPass is detected, the LED is asserted for a period of at least 300 milliseconds. In AUI mode, if CSR12 bit 1 is asserted indicating no carrier, the LED is deasserted for a period of 300 milliseconds. This pin can be used for SIA testing features.
EXT_RXEN	I/O	Receive enable or test pin signals activity on the Ethernet cable to the 21040. It is asserted when receive data is present on the Ethernet cable and deasserted at the end of a frame. It may be asserted and deasserted asynchronously to the receive clock (RCLK) by the external SIA. This pin also interfaces with the network activity LED. When any activity is detected in the network, the LED is asserted for a period of at least 300 milliseconds. This pin can be used for SIA testing features.
EXT_TCLK	I/O	Transmit clock or test pin carries the transmit clock supplied by an external SIA. The clock must always be active. This pin can be used for SIA testing features.
EXT_TX	I/O	Transmit data or test pin carries the serial output data from the 21040. This data is synchronized to the TCLK signal. This pin can be used for SIA testing features.

(continued on next page)

Table 2–1 (Cont.) Signal Pin Reference

Signal	Type	Description
EXT_TXEN	I/O	Transmit enable or test pin signals the 21040 transmit-in-progress to an external SIA. The pin is also used for SIA testing features.
FRAME_L	I/O	Cycle frame is driven by the 21040 (bus master) to indicate the beginning and duration of an access. FRAME_L asserts to indicate the beginning of a bus transaction. While FRAME_L is asserted, data transfers continue. FRAME_L deasserts to indicate that the next data phase is the final data phase transaction.
GNT_L	I	Bus grant asserts to indicate to the 21040 that access to the bus is granted.
IDSEL	I	Initialization device select asserts to act as a chip select during configuration read or write transactions.
INT_L	O/D	Interrupt request asserts when one of the appropriate bits of CSR5 sets and causes an interrupt, provided that the corresponding mask bit in CSR7 is not asserted. INT_L deasserts by writing a 1 into the appropriate CSR5 bit. If more than one interrupt bit is asserted in CSR5, the host clears only the interrupt bit that was acknowledged, INT_L deasserts for one cycle and then asserts again. This process continues until all interrupts are acknowledged. When deasserted, this pin should be pulled up by an external resistor.
IRDY_L	I/O	Initiator ready indicates the bus master's ability to complete the current data phase of the transaction. A data phase is completed on any clock when both IRDY_L and target ready (TRDY_L) are asserted. Wait cycles are inserted until both IRDY_L and TRDY_L are asserted together. When the 21040 is the bus master, IRDY_L is asserted during write operations to indicate that valid data is present on AD<31:00>. During read operations, the 21040 asserts IRDY_L to indicate that it is ready to accept data.
IREF	I	Current reference input for the analog phase lock loop logic.

(continued on next page)

Table 2–1 (Cont.) Signal Pin Reference

Signal	Type	Description
PAR	I/O	Parity is calculated by the 21040 as an even parity bit for the AD<31:00> and C_BE_L<03:00> lines. During address and data phases, parity is calculated on all the AD and C_BE_L lines whether or not any of these lines carry meaningful information.
PERR_L	I/O	Parity error asserts when a data parity error is detected. When the 21040 is the bus master and a parity error is detected, the 21040 asserts both CSR5 bit 13 (system error) and CFCS bit 8 (SERR_L enable) and completes the current data burst transaction, then stops its operation. After the host clears the system error, the 21040 continues its operation. When the 21040 is the bus target and a parity error is detected, the 21040 asserts PERR_L.
REQ_L	O	Bus request is asserted by the 21040 to indicate to the bus arbiter that it wants to use the bus.
RST_L	I	Resets the 21040 to its initial state. This signal must be asserted for at least 10 active PCI clock cycles. When in the reset state, all output pins are put into tristate and all open drain (O/D) signals are floated.
SCLK	O	Ethernet address ROM clock is used to clock data information into the 21040.
SDIN	I	Ethernet address ROM data in is used to serially shift the Ethernet identification address from the serial ROM device into the 21040.
SERR_L	O/D	If an address parity error is detected and CFCS bit 31 (detected parity error) is enabled, 21040 asserts both SERR_L (system error) and CFCS bit 30 (signal system error). When an address parity error is detected, system error asserts two clocks after the failing address. When deasserted, this pin should be pulled up by an external resistor.
SRST	O	Ethernet address ROM reset provides an asynchronous initialization of the serial ROM device.

(continued on next page)

Table 2–1 (Cont.) Signal Pin Reference

Signal	Type	Description
STOP_L	I/O	Stop indicator indicates that the current target is requesting the bus master to stop the current transaction. The 21040 responds to the assertion of STOP_L when it is the bus master, either to disconnect, retry, or abort.
TCK	I	JTAG clock shifts state information and test data into and out of the 21040 during JTAG test operations (Appendix A).
TDI	I	JTAG data in is used to serially shift test data and instructions into the 21040 during JTAG test operations (Appendix A).
TDO	O	JTAG data out is used to serially shift test data and instructions out of the 21040 during JTAG test operations (Appendix A).
TMS	I	JTAG test mode select controls the state operation of JTAG testing in the 21040 (Appendix A).
TP_RD–	I	Twisted-pair negative differential receive data from the twisted-pair lines.
TP_RD+	I	Twisted-pair positive differential receive data from the twisted-pair lines.
TP_TD– TP_TD– –	O	Twisted-pair negative differential transmit data. The positive and negative differential transmit data outputs are resistively combined outside the 21040 with equalization to compensate for intersymbol interference on the twisted-pair medium.
TP_TD+ TP_TD+ +	O	Twisted-pair positive differential transmit data. The positive and negative differential transmit data outputs are resistively combined outside the 21040 with equalization to compensate for intersymbol interference on the twisted-pair medium.

(continued on next page)

Table 2–1 (Cont.) Signal Pin Reference

Signal	Type	Description
TRDY_L	I/O	Target ready indicates the target agent's ability to complete the current data phase of the transaction. A data phase is completed on any clock when both TRDY_L and initiator ready (IRDY_L) are asserted. Wait cycles are inserted until both IRDY_L and TRDY_L are asserted together. When the 21040 is the bus master, TRDY_L is asserted by the bus slave on the read operation indicating that valid data is present on AD<31:00>. During a write cycle, it indicates that the target is prepared to accept data.
VCAP_H	I	Capacitor input for analog phase lock loop logic.
VDD	I	3.3-volt supply input voltage.
VDDAC	I	3.3-volt supply input for analog phase lock loop logic.
VDD_CLAMP	I	Supplies +5-volt or 3.3-volt reference for the clamp logic.
VSS[1]	–	Ground pin.
XTAL1	I	Crystal oscillator input.
XTAL2	O	Crystal feedback output pin used for crystal connections only. If this pin is unused, do not connect it.

[1]Device pins 90 and 96 are test pins used for Digital engineering evaluation of the 21040; they must be tied to VSS for normal chip operation.

2.3 Bus Commands

Table 2–2 lists the bus commands.

Table 2–2 Bus Commands

C_BE_L<3:0>	Command	Type of Support
0000	Interrupt acknowledge	Not supported
0001	Special cycle	Not supported
0010	I/O read	Supported as target
0011	I/O write	Supported as target

(continued on next page)

Table 2–2 (Cont.) Bus Commands

C_BE_L<3:0>	Command	Type of Support
0100	Reserved	–
0101	Reserved	–
0110	Memory read	Supported as initiator and target
0111	Memory write	Supported as initiator and target
1000	Reserved	–
1001	Reserved	–
1010	Configuration read	Supported as target
1011	Configuration write	Supported as target
1100	Reserved	–
1101	Memory write and invalidate	Not supported
1110	Memory read long	Not supported
1111	Postable memory write	Not supported

3 Registers

This chapter describes the 21040 configuration registers as well as command and status registers (CSRs). The 21040 uses eight configuration registers for initialization and configuration. Configuration registers are used to identify and query the 21040.

The 21040 contains 12 CSRs (CSR0 through CSR11) for communication with the driver to the host. It communicates with the serial interface attachment (SIA) using four additional command and status registers (CSR12 through CSR15).

CSRs are located in the 21040 and are mapped in the host I/O or memory address space. CSRs are used for the following:

- Initialization
- Pointers
- Commands
- Error reporting

3.1 21040 Configuration Operation

The 21040 enables a full software-driven initialization and configuration. This permits the software to identify and query the 21040.

The 21040 treats configuration space write operations to registers that are reserved as no-ops. That is, the access completes normally on the bus and the data is discarded. Read accesses, to reserved or non-implemented registers, complete normally and a data value of 0 is returned.

Software reset (CSR0<0>) has no effect on the configuration registers. Hardware reset clears the configuration registers.

21040 supports byte, word, and longword accesses to the configuration area.

3.1.1 Configuration Register Mapping

Table 3–1 lists the definitions and addresses for the configuration registers.

Table 3–1 Configuration Register Mapping

Configuration Register	Identifier	I/O Address
Identification	CFID	xxxxxx00H
Command and status	CFCS	xxxxxx04H
Revision	CFRV	xxxxxx08H
Latency timer	CFLT	xxxxxx0CH
Base I/O address	CBIO	xxxxxx10H
Base memory address	CBMA	xxxxxx14H
Reserved	–	xxxxxx18H - xxxxxx38H
Interrupt	CFIT	xxxxxx3CH
Driver area	CFDA	xxxxxx40H

3.1.2 Configuration Registers

The 21040 implements eight configuration registers. These registers are described in the following subsections.

3.1.2.1 Configuration ID Register (CFID)

The CFID register identifies the 21040. Figure 3–1 shows the CFID register bit fields, and Table 3–2 describes the bit fields.

Figure 3–1 CFID Configuration ID Register

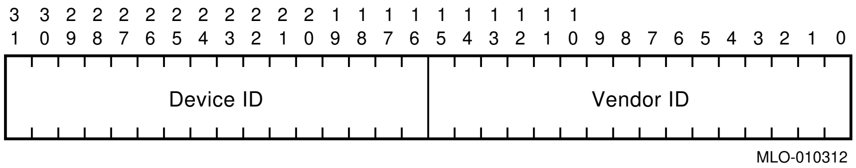


Table 3–2 CFID Configuration ID Register Description

Field	Description
31:16	**Device ID**
	Provides the unique 21040 ID number (0002H).
15:0	**Vendor ID**
	Specifies the manufacturer of the 21040 (1011H).

Table 3–3 lists the access rules for the CFID register.

Table 3–3 CFID Access Rules

Category	Description
Value after hardware or software reset	00021011H
Read access rules	–
Write access rules	Writing has no effect

3.1.2.2 Command and Status Configuration Register (CFCS)

The CFCS register is divided into two sections: a command register (CFCS<15:0>) and a status register (CFCS<31:16>).

The command register provides coarse control of the 21040's ability to generate and respond to PCI cycles. Writing 0 to this register, the 21040 logically disconnects from the PCI bus for all accesses except configuration accesses.

The status register records status information for the PCI bus-related events. The CFCS status bits do not clear when read. Writing 1 to these bits clears them; writing 0 has no effect.

Figure 3–2 shows the CFCS bit fields, and Table 3–4 describes the bit fields.

Figure 3–2 CFCS Command and Status Configuration Register

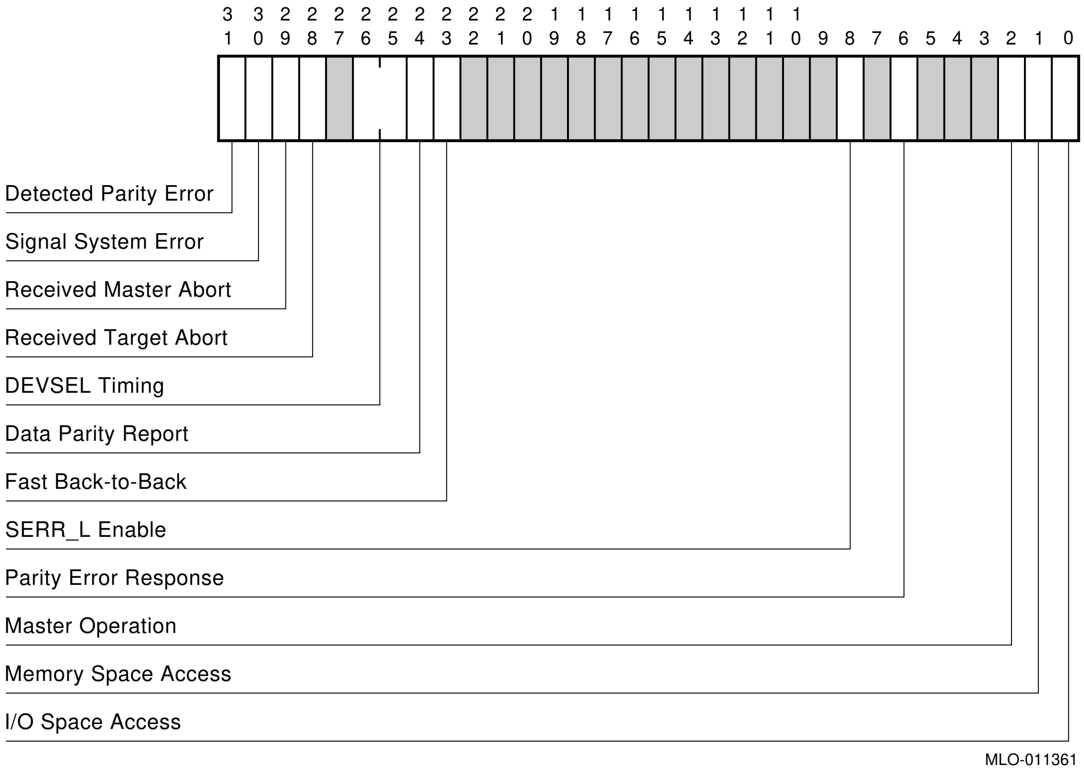


Table 3–4 CFCS Command and Status Configuration Register Description

Field	Description
31	**Detected Parity Error—status** When set, the 21040 detected a parity error, even if parity error handling is disabled in parity error response (CFCS<6>).
30	**Signal System Error—status** When set, the 21040 asserted the system error (SERR_L) pin.
29	**Received Master Abort—status** When set, the 21040 terminated a transaction with master abort.
28	**Received Target Abort—status** When set, the 21040 terminated a transaction with target abort.
26:25	**DEVSEL Timing—status** Indicates the timing of the assertion of device select (DEVSEL_L). These bits are set to 01 which indicates a medium assertion of DEVSEL_L.
24	**Data Parity Report—status** This bit sets when the following three conditions are met: • 21040 asserts parity error (PERR_L) or it senses the assertion of PERR_L by another device. • 21040 operates as a bus master for the operation that caused the error. • Parity error response (CFCS<6>) is set.
23	**Fast Back-to-Back—status** Always set by the 21040. This indicates that the 21040 is capable of accepting fast back-to-back transactions that are not sent to the same bus device.
8	**SERR_L Enable—command** When set, the 21040 asserts system error (SERR_L) when it detects a parity error on the address.
6	**Parity Error Response—command** When set, the 21040 asserts system error (CSR5<13>) after a parity error detection. When reset, any detected parity error is ignored and the 21040 continues normal operation. Parity checking is disabled after reset.

(continued on next page)

Table 3–4 (Cont.) CFCS Command and Status Configuration Register Description

Field	Description
2	**Master Operation—command**
	When set, the 21040 is capable of acting as a bus master.
	When reset, the 21040 capability to generate PCI accesses is disabled.
	For normal 21040 operation, this bit must be set.
1	**Memory Space Access—command**
	When set, the 21040 responds to memory space accesses.
	When reset, the 21040 does not respond to memory space accesses.
0	**I/O Space Access—command**
	When set, the 21040 responds to I/O space accesses.
	When reset, the 21040 does not respond to I/O space accesses.

Table 3–5 lists the access rules for the CFCS register.

Table 3–5 CFCS Access Rules

Category	Description
Value after hardware reset	All reserved bits are 0.
Read access rules	–
Write access rules	Written during configuration cycle.

3.1.2.3 Configuration Revision Register (CFRV)

The CFRV register contains the 21040 revision number. Figure 3–3 shows the CFRV bit fields, and Table 3–6 describes the bit fields.

Figure 3–3 CFRV Configuration Revision Register

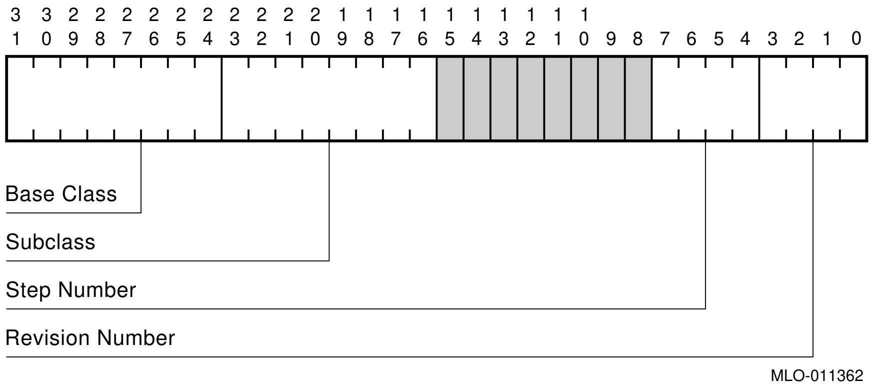


Table 3–6 CFRV Configuration Revision Register Description

Field	Description
31:24	**Base Class**
	Indicates the network controller and is equal to 2H.
23:16	**Subclass**
	Indicates the Ethernet controller and is equal to 0H.
7:4	**Step Number**
	Indicates the 21040 step number and is equal to 2H. This number is incremented for subsequent 21040 steps.
3:0	**Revision Number**
	Indicates the 21040 revision number and is equal to either 0H, 1H, 2H, or 3H. This number is incremented for subsequent 21040 revisions within the current step.

Table 3–7 lists the access rules for the CFRV register.

Table 3–7 CFRV Access Rules

Category	Description
Value after hardware or software reset	0200FF20H, 0200FF21H, 0200FF22H, or 0200FF23H
Read access rules	–
Write access rules	Writing has no effect

3.1.2.4 Configuration Latency Timer Register (CFLT)

This register configures the 21040 bus latency timer. Figure 3–4 shows the CFLT bit field, and Table 3–8 describes the bit field.

Figure 3–4 CFLT Configuration Latency Timer Register

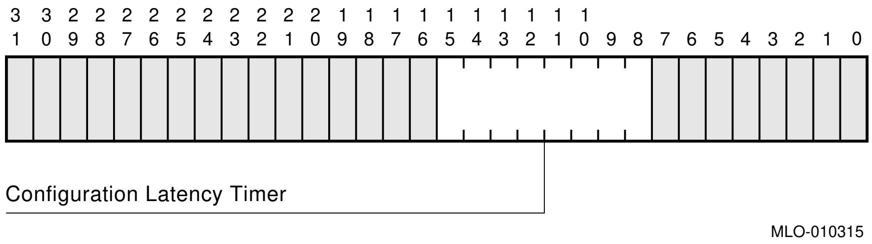


Table 3–8 CFLT Configuration Latency Timer Register Description

Field	Description
15:8	**Configuration Latency Timer** Specifies, in units of PCI bus clocks, the value of the latency timer of the 21040. When the 21040 asserts FRAME_L, it enables its latency timer to count. If the 21040 deasserts FRAME_L prior to count expiration, the content of the latency timer is not valid. Otherwise, after the count expires, the 21040 initiates transaction termination as soon as its GNT_L is deasserted.

Table 3–9 lists the access rules for the CFLT register.

Table 3–9 CFLT Access Rules

Category	Description
Value after software reset	All reserved bits are 0.
Read access rules	–
Write access rules	Written once during configuration.

3.1.2.5 Configuration Base I/O Address Register (CBIO)

The CBIO register specifies the base I/O address for accessing the 21040 CSRs (CSR0 through CSR15). For example, if the CBIO register is programmed to 1000H, the I/O address of CSR15 is equal to CBIO + CSR15-offset for a value of 1078H (Table 3–18).

This register must be initialized prior to accessing any CSR.

Figure 3–5 shows the CBIO bit fields and Table 3–10 describes the bit fields.

Figure 3–5 CBIO Configuration Base I/O Address Register

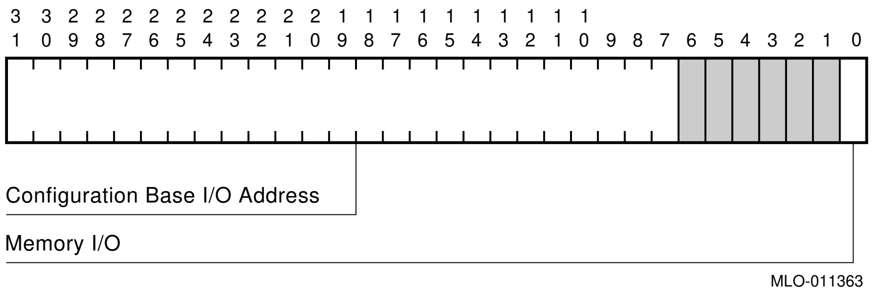


Table 3–10 CBIO Configuration Base I/O Address Register Description

Field	Description
31:7	**Configuration Base I/O Address** Defines the address assignment mapping of 21040 CSRs.
6:1	**This field value is 0 when read.**
0	**I/O Space Indicator** Determines that the register maps into the I/O space. The value in this field is 1. This is a read-only field.

Table 3–11 lists the access rules for the CBIO register.

Table 3–11 CBIO Access Rules

Category	Description
Value after reset	Software reset has no effect.
Read access rules	–
Write access rules	Written once during configuration.

3.1.2.6 Configuration Base Memory Address Register (CBMA)

The CBMA register specifies the base memory address for memory accesses to the 21040 CSRs (CSR0 through CSR15).

This register must be initialized prior to accessing any CSR0 register.

Figure 3–6 shows the CBMA bit fields, and Table 3–12 describes the bit fields.

Figure 3–6 CBMA Configuration Base Memory Address Register

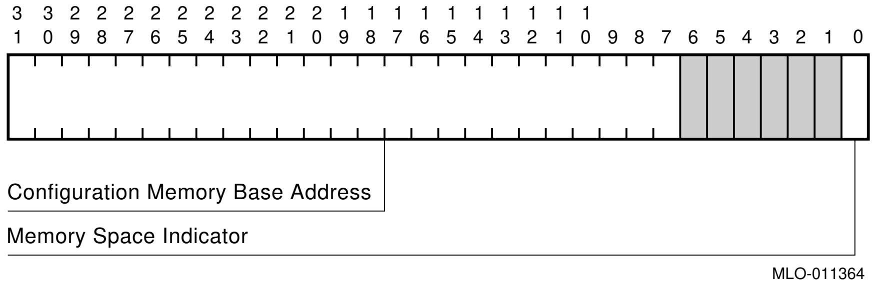


Table 3–12 CBMA Configuration Base Memory Address Register Description

Field	Description
31:7	**Configuration Base Memory Address** Defines the address assignment mapping of the 21040 CSRs.
6:1	**This field value is 0 when read.**
0	**Memory Space Indicator** Determines that the register maps into the memory space. The value in this field is 0. This is a read-only field.

Table 3–13 lists the access rules for the CBMA register.

Table 3–13 CBMA Access Rules

Category	Description
Value after reset	Software reset has no effect.
Read access rules	–
Write access rules	Written once during configuration.

3.1.2.7 Configuration Interrupt Register (CFIT)

The CFIT register is divided into two sections: the interrupt line and the interrupt pin. CFIT configures both the system's interrupt line and the 21040 interrupt pin connection.

Figure 3–7 shows the CFIT bit fields, and Table 3–14 describes the bit fields.

Figure 3–7 CFIT Configuration Interrupt Register

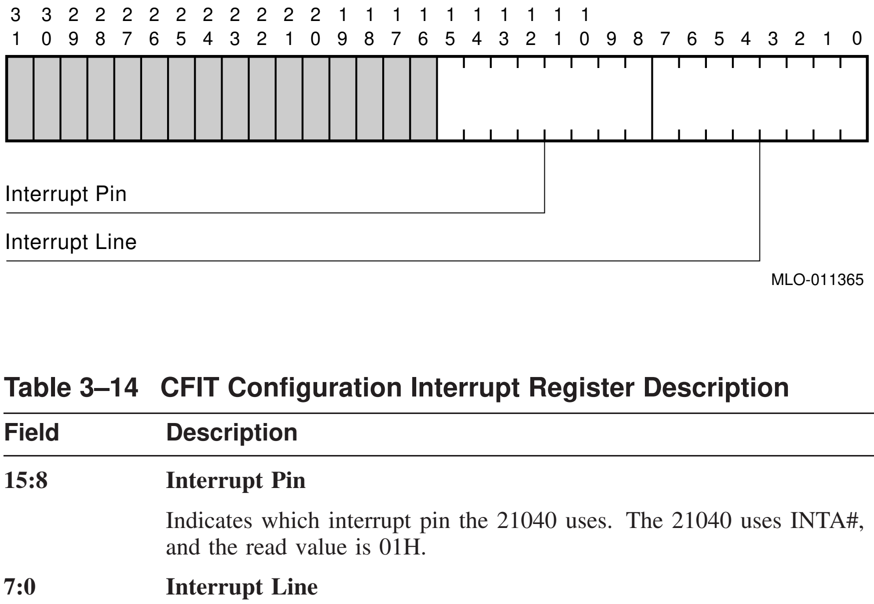


Table 3–14 CFIT Configuration Interrupt Register Description

Field	Description
15:8	**Interrupt Pin** Indicates which interrupt pin the 21040 uses. The 21040 uses INTA#, and the read value is 01H.
7:0	**Interrupt Line** Provides interrupt line routing information. The BIOS writes the routing information into this field when it initializes and configures the system. The value in this field indicates which input of the system interrupt controller the 21040's interrupt pin is connected to. The driver can use this information to determine priority and vector information. Values in this field are architecture-specific.

Table 3–15 lists the access rules for the CFIT register.

Table 3–15 CFIT Access Rules

Category	Description
Value after reset	Software reset has no effect.
Read access rules	–
Write access rules	–

3.1.2.8 Configuration Driver Area Register (CFDA)

The CFDA register can be used to store driver-specific information during initialization. It has no effect on the 21040 operation.

Figure 3–8 shows the CFDA bit field, and Table 3–16 describes the bit field.

Figure 3–8 CFDA Configuration Driver Area Register

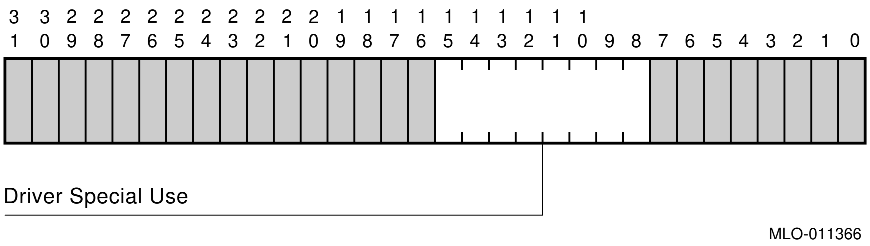


Table 3–16 CFDA Configuration Driver Area Register Description

Field	Description
15:8	**Driver Special Use** Read and write fields for the driver's special use.

Table 3–17 lists the access rules for the CFDA register.

Table 3–17 CFDA Access Rules

Category	Description
Value after reset	Software reset has no effect.
Read access rules	–
Write access rules	–

3.2 Command and Status Registers

The 21040 contains 16 command and status registers, which can be accessed by the host. Table 3–18 lists the CSR registers.

Table 3–18 CSR Mapping

Register	Meaning	Offset from CSR Base Address (CBIO, CBMA)
CSR0	Bus mode register	00H
CSR1	Transmit poll demand	08H
CSR2	Receive poll demand	10H
CSR3	Receive list base address	18H
CSR4	Transmit list base address	20H
CSR5	Status register	28H
CSR6	Operation mode register	30H
CSR7	Interrupt mask register	38H
CSR8	Missed frame counter	40H
CSR9	Ethernet ROM register	48H
CSR10	Reserved	50H
CSR11	Full-duplex register	58H
CSR12	SIA status register	60H
CSR13	SIA connectivity register	68H
CSR14	SIA transmit receive register	70H
CSR15	SIA general register	78H

The 21040 CSRs are located in the host I/O or memory address space. The CSRs are *quadword*-aligned and can only be accessed using *longword* instructions.

Note

- Register access is only longword access; byte accesses to CSR0–CSR15 are not supported. Accessing a non-longword address register causes UNPREDICTABLE data results.
- Reserved bits must be written with 0. Reserved bits are UNPREDICTABLE on read accesses.
- Retries on second data transactions occur in response to burst I/O accesses.

CSRs are physically located in the chip. The host uses a single instruction to access to a CSR. Most commonly used 21040 features are contained in the CSRs.

3.2.1 Host CSRs

There are 12 CSRs (CSR0 through CSR11) used to communicate with the host.

3.2.1.1 Bus Mode Register (CSR0)

Figure 3–9 shows the CSR0 bit fields, and Table 3–19 describes the bit fields. CSR0 establishes the bus operating modes.

Figure 3–9 CSR0 Bus Mode Register

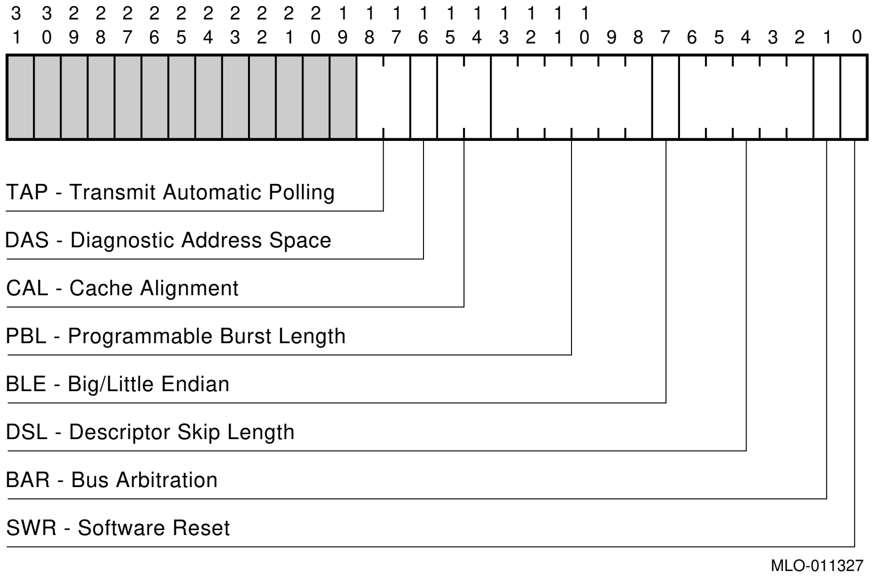


Table 3–19 CSR0 Bus Mode Register Description

Field	Description
18:17	**TAP—Transmit Automatic Polling (read, write)** When set and the 21040 is in a suspended state because of a transmit buffer unavailable, the 21040 performs a transmit automatic poll demand (Table 3–20).
16	**DAS—Diagnostic Address Space (read, write)** When reset, CSR0 through CSR15 are mapped on I/O space and memory space (21040 address space becomes 128 bytes). When set, all 16 CSRs and all diagnostic registers are mapped on I/O and memory space.
15:14	**CAL—Cache Alignment (read, write)** Programmable address boundaries for data burst stop (Table 3–22). If the buffer is not aligned, the 21040 executes the first transfer up to the address boundary, then all transfers are aligned to the specified boundary.
13:8	**PBL—Programmable Burst Length (read, write)** Indicates the maximum number of longwords to be transferred in one DMA transaction. If PBL = 0, the 21040 burst is limited only by the amount of data stored in the receive FIFO (at least 16 longwords) or by the amount of free space in the transmit FIFO (at least 16 longwords) before issuing a bus request. The PBL can be programmed with permissible values 0, 1, 2, 4, 8, 16, or 32. After reset, the PBL default value is 0.
7	**BLE—Big/Little Endian (read, write)** When set, the 21040 operates in big endian byte ordering mode. When reset, the 21040 operates in little endian byte ordering mode. Big endian is applicable only for data buffers. For example, the byte order in little endian of a data buffer is 12345678H, with each digit representing a nibble. In big endian, the byte orientation is 78563412H.
6:2	**DSL—Descriptor Skip Length (read, write)** Specifies the number of longwords to skip between two descriptors. To improve performance, descriptors can be placed in a separate cache line and should not have to be contiguous.
1	**BAR—Bus Arbitration (read, write)**

(continued on next page)

Table 3–19 (Cont.) CSR0 Bus Mode Register Description

Field	Description
	Selects the internal bus arbitration between the receive and transmit processes. When set, a round robin arbitration scheme is applied resulting in equal sharing between processes. When reset to 0, the receive process has priority over the transmit process, unless the 21040 is currently transmitting (Section 5.2).
0	**SWR—Software Reset (read, write)** When set, the 21040 resets all internal hardware. When reset, duration should be at least 10 PCI clock cycles. After reset deassertion, the first bus transaction to the 21040 should not be initiated before at least 50 more PCI cycles elapse. Software reset does not affect the configuration area.

Table 3–20 defines the transmit automatic polling bits.

Table 3–20 Transmit Automatic Polling Bits

CSR0<18:17>	Time Intervals
00	No transmit automatic polling; CSR1 access should be used to poll the transmit descriptor list.
01	Transmit automatic polling every 200 microseconds.
10	Transmit automatic polling every 800 microseconds.
11	Transmit automatic polling every 1.6 milliseconds.

Table 3–21 lists the CSR0 access rules.

Table 3–21 CSR0 Access Rules

Category	Description
Value after reset	FFF80000H.
Read access rules	–
Write access rules	To write, the transmit and receive processes must be stopped. If one or both of the processes is not stopped, the result is UNPREDICTABLE.

Table 3–22 defines the cache address alignment bits.

Table 3–22 Cache Address Alignment Bits

CSR0<15:14>	Address Alignment
00	Not used
01	8-longword boundary alignment
10	16-longword boundary alignment
11	32-longword boundary alignment

3.2.1.2 Transmit Poll Demand (CSR1)

Figure 3–10 shows the CSR1 bit field, and Table 3–23 describes the bit field.

Figure 3–10 CSR1 Transmit Poll Demand

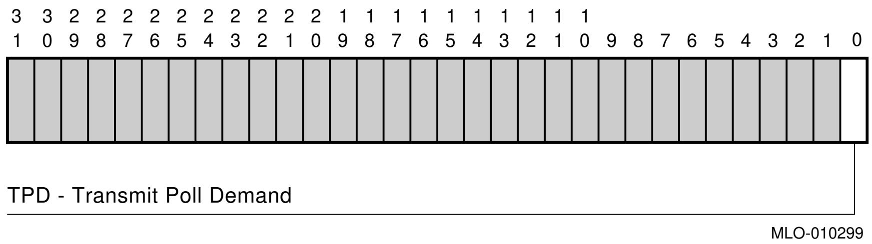


Table 3–23 CSR1 Transmit Poll Demand Description

Field	Description
0	**TPD—Transmit Poll Demand (write)** When written with any value, the 21040 checks for frames to be transmitted. If no descriptor is available, the transmit process returns to the suspended state and CSR5<2> is not asserted. If the descriptor is available, the transmit process resumes.

Table 3–24 lists the CSR1 access rules.

Table 3–24 CSR1 Access Rules

Category	Description
Value after reset	FFFFFFFFH
Read access rules	–
Write access rules	Not effective if the transmit process is not in the suspended state.

3.2.1.3 Receive Poll Demand (CSR2)

Figure 3–11 shows the CSR2 bit field, and Table 3–25 describes the bit field.

Figure 3–11 CSR2 Receive Poll Demand

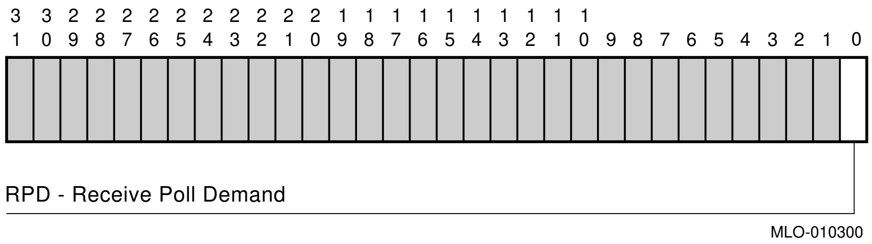


Table 3–25 CSR2 Receive Poll Demand Description

0	**RPD—Receive Poll Demand (write)** When written with any value, the 21040 checks for receive descriptors to be acquired. If no descriptor is available, the receive process returns to the suspended state and CSR5<7> is not asserted. If the descriptor is available, the receive process resumes.

Table 3–26 lists the access rules for CSR2.

Table 3–26 CSR2 Access Rules

Category	Description
Value after reset	FFFFFFFFH
Read access rules	–
Write access rules	Effective only if the receive process is in the suspended state.

3.2.1.4 Descriptor List Addresses (CSR3, CSR4)

The CSR3 descriptor list address register is used for receive buffer descriptors, and the CSR4 descriptor list address register is used for transmit buffer descriptors. In both cases, the registers are used to point the 21040 to the start of the appropriate descriptor list.

Figure 3–12 shows the CSR3 bit field, and Table 3–27 describes the bit field.

Note

The descriptor lists reside in *physical* memory space and must be *longword*-aligned. The 21040 behaves unpredictably when the lists are not longword-aligned.

Writing to either CSR3 or CSR4 is permitted only when its respective process is in the stopped state. When stopped, the CSR3 and CSR4 registers must be written *before* the respective START command is given (Section 3.2.1.6).

Figure 3–12 CSR3 Receive List Base Address

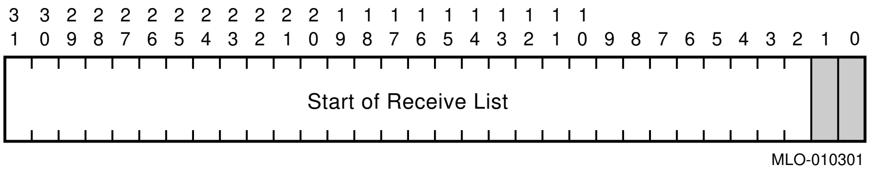


Table 3–27 CSR3 Receive List Base Address Description

Field	Description
31:2	**Start of receive list (read, write)**
1:0	**Must be 00 (read, write)**

Figure 3–13 shows the CSR4 bit field, and Table 3–28 describes the bit field.

Figure 3–13 CSR4 Transmit List Base Address

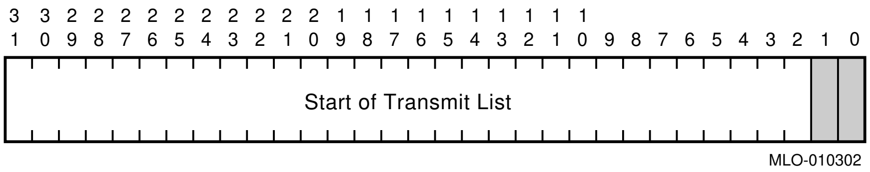


Table 3–28 CSR4 Transmit List Base Address Description

Field	Description
31:2	**Start of transmit list (read, write)**
1:0	**Must be 00 (read, write)**

Table 3–29 lists the access rules for CSR3, and Table 3–30 lists the access rules for CSR4.

Table 3–29 CSR3 Access Rules

Category	Description
Value after reset	UNPREDICTABLE
Read access rules	–
Write access rules	Receive process stopped

Table 3–30 CSR4 Access Rules

Category	Description
Value after reset	UNPREDICTABLE
Read access rules	–
Write access rules	Transmit process stopped

3.2.1.5 Status Register (CSR5)

The status register CSR5 contains all the status bits that the 21040 reports to the host. CSR5 is usually read by the driver during interrupt service routine or polling. Most of the fields in this register cause the host to be interrupted. CSR5 bits are not cleared when read. Writing 1 to these bits clears them; writing 0 has no effect. Each field can be masked (Section 3.2.1.7).

Figure 3–14 shows the CSR5 bit fields, and Table 3–31 describes the bit fields.

Figure 3–14 CSR5 Status Register

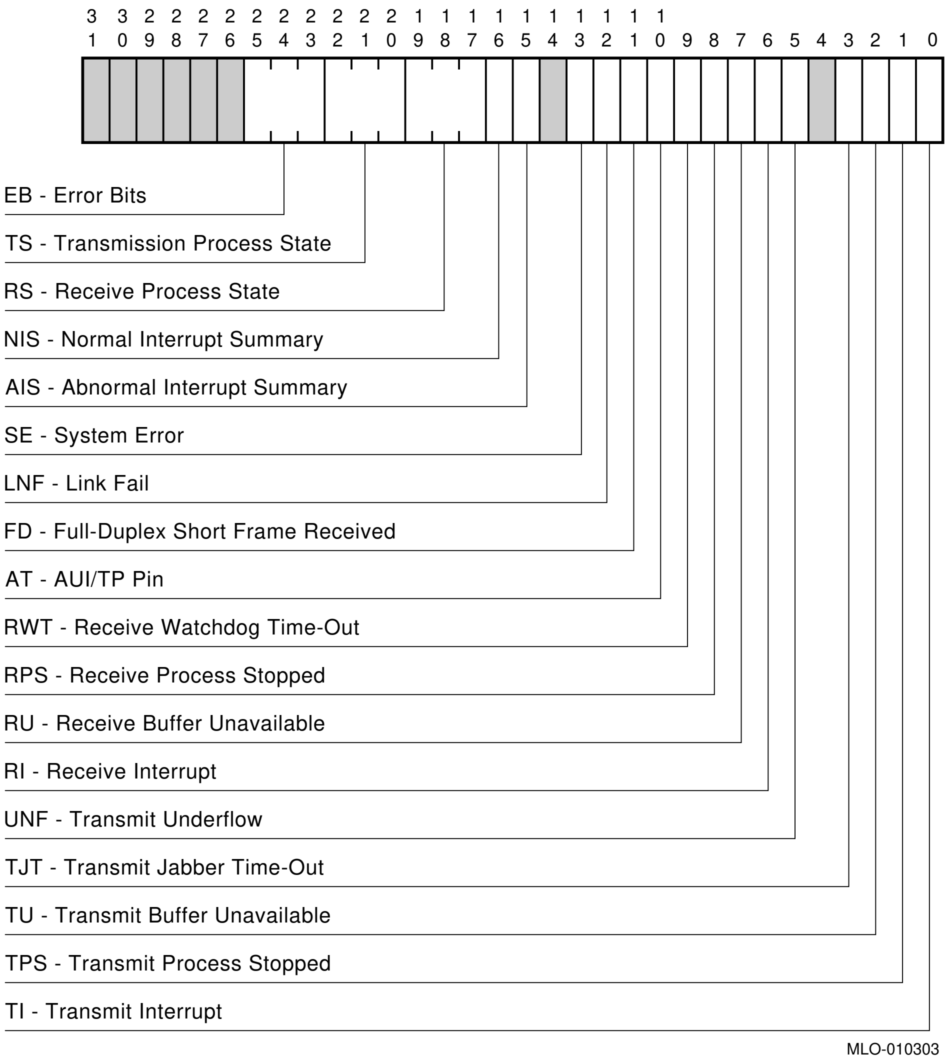


Table 3–31 CSR5 Status Register Description

Field	Description
25:23	**EB—Error Bits (read)** Indicates the type of error that caused system error. Valid only when system error CSR5<13> is set (Table 3–32). This field does not generate an interrupt.
22:20	**TS—Transmit Process State (read)** Indicates the state of the transmit process (Table 3–33). This field does not generate an interrupt.
19:17	**RS—Receive Process State (read)** Indicates the state of the receive process (Table 3–34). This field does not generate an interrupt.
16	**NIS—Normal Interrupt Summary (read, write)** Normal interrupt summary bit. Its value is the logical OR of CSR5<0>—Transmit interrupt CSR5<2>—Transmit buffer unavailable CSR5<6>—Receive interrupt Unmasked bits affect only the normal interrupt summary CSR5<16> bit.
15	**AIS—Abnormal Interrupt Summary (read, write)** Abnormal interrupt summary bits. Its value is the logical OR of CSR5<1>—Transmit process stopped CSR5<3>—Transmit jabber time-out CSR5<5>—Transmit underflow CSR5<7>—Receive buffer unavailable CSR5<8>—Receive process stopped CSR5<9>—Receive watchdog time-out CSR5<10>—AUI/TP pin CSR5<11>—Full-duplex short frame received CSR5<12>—Link fail CSR5<13>—System error Unmasked bits affect only the abnormal interrupt summary CSR5<15> bit.
13	**SE—System Error (read, write)** Indicates that a system error occurred (Table 3–32).
12	**LNF—Link Fail (read, write)** Indicates that a link fail occurred in the twisted-pair lines. See link fail status CSR12<2>.

(continued on next page)

Table 3–31 (Cont.) CSR5 Status Register Description

Field	Description
11	**FD—Full-Duplex Short Frame Received (read, write)**
	Indicates that the first full-duplex short frame was received. The driver should wait for the second 64-byte full-duplex packet (Section 5.8).
	The full-duplex auto configuration short packet is treated as any other runt frame except that full-duplex short frame received CSR5<11> is asserted. (In pass bad frame or promiscuous filtering modes, this packet is transferred to the host.)
10	**AT—AUI/TP Pin (read, write)**
	Indicates that the SIA AUI/TP pin has changed position.
9	**RWT—Receive Watchdog Time-Out (read, write)**
	Indicates that the receive watchdog timer expired, and another node is babbling on the network. Current frame reception aborts while length error RDES0<14> and last descriptor RDES0<8> assert. Receive interrupt CSR5<6> also asserts, and the receive process remains in the running state.
8	**RPS—Receive Process Stopped (read, write)**
	Indicates that the receive process is stopped. Table 5–2 explains the receive process state transitions.
7	**RU—Receive Buffer Unavailable (read, write)**
	Indicates that the next descriptor in the receive list is owned by the host and cannot be acquired by the 21040. The reception process is suspended. To resume processing receive descriptors, the host should change the ownership of the descriptor and might issue a receive poll demand command. If no receive poll demand is issued, the reception process resumes when the next recognized incoming frame is received.
	After the first assertion, CSR5<7> does not assert for any subsequent not owned receive descriptors fetches. CSR5<R7> asserts only when the previous receive descriptor was owned by the 21040.
6	**RI—Receive Interrupt (read, write)**
	Indicates the completion of a frame reception. Specific frame status information has been posted in the descriptor. The reception process remains in the running state.

(continued on next page)

Table 3–31 (Cont.) CSR5 Status Register Description

Field	Description
5	**UNF—Transmit Underflow (read, write)** Indicates that the transmit FIFO had an underflow condition during the packet transmission. The transmit process is placed in the suspended state, and underflow error TDES0<1> is set.
3	**TJT—Transmit Jabber Time-Out (read, write)** Indicates that the transmit jabber timer expired, meaning that the 21040 transmitter was babbling. The transmission process is aborted and placed in the stopped state. This event causes the transmit jabber time-out TDES0<14> flag to assert.
2	**TU—Transmit Buffer Unavailable (read, write)** Indicates that the next descriptor on the transmit list is owned by the host and cannot be acquired by the 21040. The transmission process is suspended. Table 5–3 explains the transmit process state transitions. To resume processing transmit descriptors, the host should change the ownership bit of the descriptor, then issue a transmit poll demand command, unless transmit automatic polling (Table 3–20) is enabled.
1	**TPS—Transmit Process Stopped (read, write)** Asserts when the transmit process enters the stopped state.
0	**TI—Transmit Interrupt (read, write)** Indicates that a frame transmission was completed, while TDES1<31> is asserted in the first descriptor of the frame.

Table 3–32 lists the bit codes for the bus error bits.

Table 3–32 Bus Error Bits

CSR5<25:23>	Process State	Recover Mechanism
000	Parity error	21040 software reset CSR0<0> = 1
001	Master abort	System error CSR5<13> = 1 (Section 6.4.2.1.3)
010	Target abort	System error CSR5<13> = 1 (Section 6.4.2.2.1)
011	Reserved	–
1xx	Reserved	–

Table 3–33 lists the bit codes for the transmit process state.

Table 3–33 Transmit Process State

CSR5<22:20>	Process State
000	Stopped—RESET command or transmit jabber expired
001	Running—Fetch transmit descriptor
010	Running—Wait for end of transmission
011	Running—Read buffer from memory, and queue the data into the transmit FIFO
100	Reserved
101	Running—Setup packet
110	Suspended—Transmit FIFO underflow or an unavailable transmit descriptor
111	Running—Close transmit descriptor

Table 3–34 lists the bit codes for the receive process state.

Table 3–34 Receive Process State

CSR5<19:17>	Process State
000	Stopped—RESET or STOP RECEIVE command
001	Running—Fetch receive descriptor
010	Running—Check for end-of-receive packet before prefetch of next descriptor
011	Running—Wait for receive packet
100	Suspended—Unavailable receive buffer
101	Running—Close receive descriptor
110	Running—Flush the current frame from the receive FIFO because of unavailable receive buffer
111	Running—Queue the receive frame from the receive FIFO into the receive buffer

Table 3–35 lists the access rules for CSR5.

Table 3–35 CSR5 Access Rules

Category	Description
Value after reset	FC000000H
Read access rules	–
Write access rules	CSR5 bits 0 through 16 are cleared by writing 1. Writing 0 to these bits has no effect. Writing to CSR5 bits 17 through 25 has no effect.

3.2.1.6 Operation Mode Register (CSR6)

CSR6 establishes the receive and transmit operating modes and commands. CSR6 should be the last CSR to be written as part of initialization. Figure 3–15 shows the CSR6 bit fields, and Table 3–36 describes the bit fields.

Figure 3–15 CSR6 Operating Mode Register

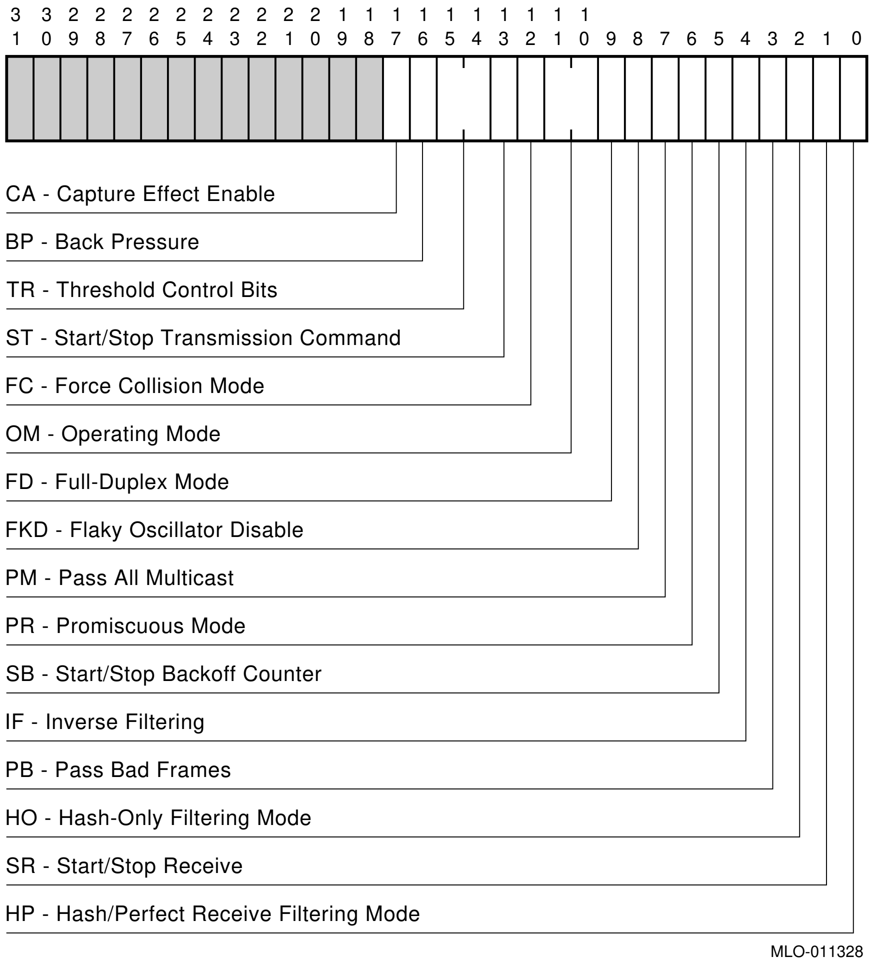


Table 3–36 CSR6 Operating Mode Register Description

Field	Description
17	**CA—Capture Effect Enable (read, write)**
	When set, enables the resolution of the capture effect on the network (Section 7.4.7).
	When reset, the 21040 disables the resolution of the capture effect on the network.
	This feature is not part of the IEEE 802.3 and Ethernet standards.
16	**BP—Back Pressure (read, write)**
	When set, enables the transmit back pressure logic. When receive data buffers are exhausted, the 21040 asserts the transmit carrier for a maximum period of 500 milliseconds. Upon back pressure, if a 21040 receive descriptor becomes available (a receive poll demand was issued), the 21040 stops back pressure and fetches the descriptor.
	When reset, disables the transmit back pressure logic.
	This feature is not part of the IEEE 802.3 and Ethernet standards.
15:14	**TR—Threshold Control Bits (read, write)**
	Controls the selected threshold level for the 21040 transmit FIFO. Four threshold levels are allowed (Table 3–37).
	The threshold value has a direct impact on the 21040 bus arbitration scheme (Section 5.2).
	Transmission starts when the frame size within the transmit FIFO is larger than the threshold. Full frames with a length less than the threshold are also transmitted.
	The transmit process must be in the stopped state to change these bits (CSR6<15:14>).
13	**ST—Start/Stop Transmission Command (read, write)**

(continued on next page)

Table 3–36 (Cont.) CSR6 Operating Mode Register Description

Field	Description
	When set, the transmission process is placed in the running state, and the 21040 checks the transmit list at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the transmit list base address set by CSR4, or from the position retained when the transmit process was previously stopped. If no descriptor can be acquired, the transmit process enters the suspended state. If the current descriptor is not owned by the 21040, the transmission process enters the suspended state, and transmit buffer unavailable CSR5<2> is set. The start transmission command is honored only when the transmission process is stopped. If the command is issued before setting CSR4, the 21040 will behave unpredictably. When reset, the transmission process is placed in the stopped state after completing the transmission of the current frame. The next descriptor position in the transmit list is saved and becomes the current position when transmission is restarted. The stop transmission command is honored only when the transmission process is in either the running or suspended state (Table 5–3).
12	**FC—Force Collision Mode (read, write)** Allows the collision logic to be tested. Meaningful only in internal loopback mode. When set, a collision is forced during the next transmission attempt. This results in 16 transmission attempts with excessive collision reported in the transmit descriptor (TDES0<8>).
11:10	**OM—Operating Mode (read, write)** Selects the 21040 main mode of operation (Table 3–59).
9	**FD—Full-Duplex Mode (read, write)** When set, the 21040 operates in a full-duplex mode (Section 5.8). The 21040 transmits and receives functions simultaneously (Table 3–59). Setting the 21040 to operate in full-duplex mode is allowed only if the transmit and receive processes are in the stopped state, and the start/stop receive (CSR6<1>) and start/stop transmission commands (CSR6<13>) are both set to 0. While in full-duplex mode, heartbeat check is disabled, heartbeat fail TDES0<7> should be ignored, and internal loopback is not allowed.
8	**FKD—Flaky Oscillator Disable (read, write)** When set, indicates that the internal flaky oscillator is disabled; pseudo random numbers are chosen instead of fully random numbers. This bit is set only for diagnostic purposes.
7	**PM—Pass All Multicast (read, write)**

(continued on next page)

Table 3–36 (Cont.) CSR6 Operating Mode Register Description

Field	Description
	When set, indicates that all the incoming frames with a multicast destination address (first bit in the destination address field is 1) are received. Incoming frames with physical address destinations are filtered according to the CSR6<0> bit.
6	**PR—Promiscuous Mode (read, write)**
	When set, indicates that any incoming valid frame is received, regardless of its destination address.
	After reset, the 21040 wakes up in promiscuous mode.
5	**SB—Start/Stop Backoff Counter (read, write)**
	When set, indicates that the internal backoff counter stops counting when any carrier activity is detected. The 21040 backoff counter resumes when the carrier drops. The earliest the 21040 starts its transmission is 9.6 microseconds after carrier deassertion.
	When reset, the internal backoff counter is not affected by the carrier activity.
	This feature violates IEEE 802.3 and Ethernet standards.
4	**IF—Inverse Filtering (read)**
	When set, the 21040 operates in an inverse filtering mode (Table 4–8).
3	**PB—Pass Bad Frames (read, write)**
	When set, the 21040 operates in pass bad frame mode. All incoming frames that passed the address filtering are received, including runt frames, collided fragments, or truncated frames caused by FIFO overflow.
	If any received bad frames are required, promiscuous mode (CSR6<6>) should be set to 1.
2	**HO—Hash-Only Filtering Mode (read)**
	When set, the 21040 operates in an imperfect address filtering mode for both physical and multicast addresses (Table 4–8).
1	**SR—Start/Stop Receive (read, write)**

(continued on next page)

Table 3–36 (Cont.) CSR6 Operating Mode Register Description

Field	Description
	When set, the receive process is placed in the running state. The 21040 attempts to acquire a descriptor from the receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by CSR3 or the position retained when the receive process was previously stopped. If no descriptor is owned by the 21040, the receive process enters the suspended state and receive buffer unavailable (CSR5<7>) sets. The start reception command is honored only when the reception process has stopped. If the command was issued before setting CSR3, the 21040 behaves unpredictably. When cleared, the receive process enters the stopped state after completing the reception of the current frame. The next descriptor position in the receive list is saved, and becomes the current position after the receive process is restarted. The stop reception command is honored only when the receive process is in running or suspended state (Section 5.5.4).
0	**HP—Hash/Perfect Receive Filtering Mode (read)** When reset, the 21040 does a perfect address filter of incoming frames according to the addresses specified in the setup frame (Table 4–8). When set, the 21040 does imperfect address filtering of the incoming frame according to the hash table specified in the setup frame.

Table 3–37 lists the threshold values in bytes.

Table 3–37 Transmit Threshold

CSR6<15:14>	Threshold (Bytes)
00	72
01	96
10	128
11	160

Table 3–38 lists the codes to determine the filtering mode.

Table 3–38 Filtering Mode

CSR6<7>	CSR6<6>	CSR6<4>	CSR6<2>	CSR6<0>	Filtering Mode
0	0	0	0	0	16 perfect filtering
0	0	0	0	1	512-bit hash + 1 perfect filtering
0	0	0	1	0	512-bit hash for multicast and physical addresses
0	0	0	1	1	Not applicable
0	0	1	0	0	Inverse filtering
0	0	1	0	1	Not applicable
0	0	1	1	0	Not applicable
0	0	1	1	1	Not applicable
0	1	0	0	0	Promiscuous
0	1	0	0	1	Promiscuous
0	1	0	1	0	Promiscuous
0	1	0	1	1	Not applicable
0	1	1	0	0	Not used
0	1	1	0	1	Not used
0	1	1	1	0	Not used
0	1	1	1	1	Not used
1	0	0	0	0	Pass all multicast
1	0	0	0	1	Pass all multicast
1	0	0	1	0	Pass all multicast
1	0	0	1	1	Not applicable
1	0	1	0	0	Not used
1	0	1	0	1	Not applicable
1	0	1	1	0	Not applicable
1	0	1	1	1	Not applicable
1	1	0	0	0	Promiscuous
1	1	0	0	1	Promiscuous
1	1	0	1	0	Promiscuous

(continued on next page)

Table 3–38 (Cont.) Filtering Mode

CSR6<7>	CSR6<6>	CSR6<4>	CSR6<2>	CSR6<0>	Filtering Mode
1	1	0	1	1	Not applicable
1	1	1	0	0	Not used
1	1	1	0	1	Not applicable
1	1	1	1	0	Not applicable
1	1	1	1	1	Not applicable

Table 3–39 describes the only conditions that permit change to a field when modifying values to CSR6.

Table 3–39 CSR6 Access Rules

Category	Description
Value after reset	FFFC0040H
Read access rules	–
Write access rules	
* CSR6<11:10>	Receive and transmit processes stopped
* CSR6<12>	Receive and transmit processes stopped, internal_loopback mode
* CSR6<3>	Receive process stopped
* CSR6<15:14>	Transmit process stopped
* CSR6<8>	Transmit process stopped
* CSR6<9>	Transmit process stopped
* CSR6<5>	Receive and transmit processes stopped
* CSR6<17>	Receive and transmit processes stopped
* CSR6<16>	Receive and transmit processes stopped
* Start_Receive CSR6<1>=1	CSR3 initialized
* Start_Transmit CSR6<13>=1	CSR4 initialized
* Stop_Receive CSR6<1>=0	Receive running or suspended
* Stop_Transmit CSR6<13>=0	Transmit running or suspended

3.2.1.7 Interrupt Mask Register (CSR7)

The Interrupt Mask register (CSR7) masks the interrupts reported by CSR5 (Section 3.2.1.5). Setting a bit to 1 enables a corresponding interrupt. After a hardware or software reset, all interrupts are disabled. Figure 3–16 shows the CSR7 bit fields, and Table 3–40 describes the bit fields.

Figure 3–16 CSR7 Interrupt Mask Register

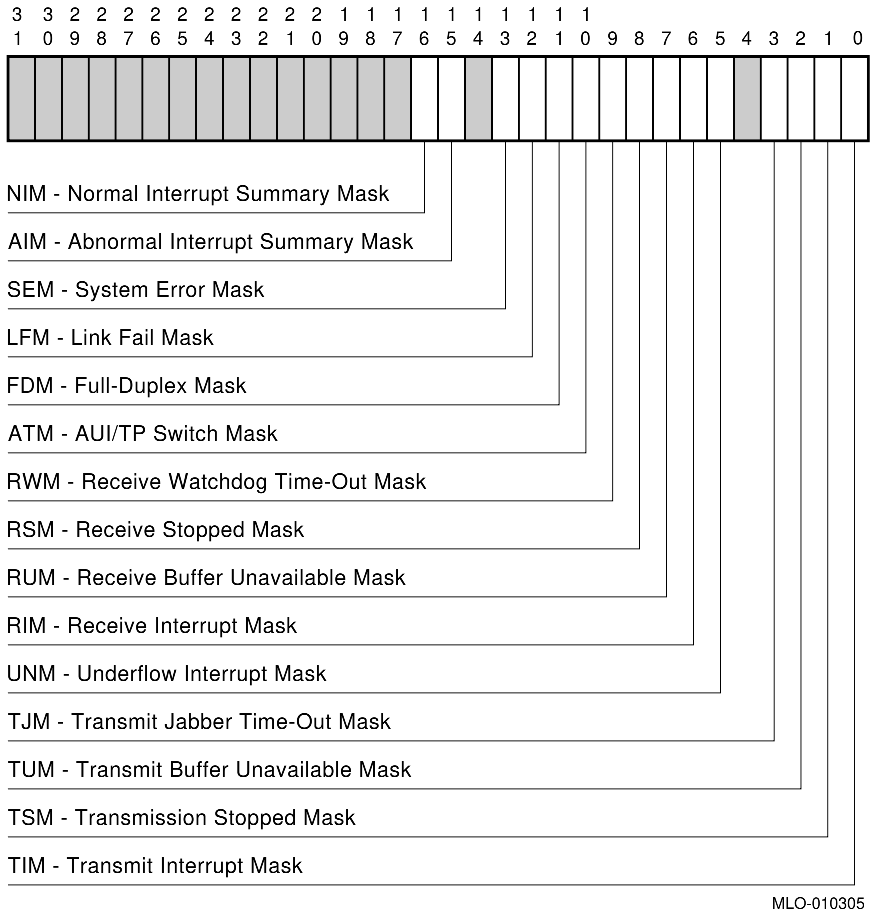


Table 3–40 CSR7 Interrupt Mask Register Description

Field	Description
16	**NIM—Normal Interrupt Summary Mask (read, write)** When set, normal interrupt is posted. When reset, no normal interrupt is posted. This bit (CSR7<16>) masks the following bits: CSR5<0>—Transmit interrupt CSR5<2>—Transmit buffer unavailable CSR5<6>—Receive interrupt
15	**AIM—Abnormal Interrupt Summary Mask (read, write)** When set, abnormal interrupt is posted. When reset, no abnormal interrupt is posted. This bit CSR7<15> masks the following bits: CSR5<1>—Transmit process stopped CSR5<3>—Transmit jabber time-out CSR5<5>—Transmit underflow CSR5<7>—Receive buffer unavailable CSR5<8>—Receive process stopped CSR5<9>—Receive watchdog time-out CSR5<10>—AUI/TP pin CSR5<11>—Full-duplex short frame received CSR5<12>—Link fail CSR5<13>—System error
13	**SEM—System Error Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and system error (CSR5<13>), the interrupt is posted. When reset and system error (CSR5<13>) is set, the interrupt posting is disabled.
12	**LFM—Link Fail Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and link fail (CSR5<12>), the interrupt is posted. When reset and link fail (CSR5<12>) is set, the interrupt posting is disabled.
11	**FDM—Full-Duplex Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and full-duplex short frame received (CSR5<11>), the interrupt is posted. When reset and full-duplex CSR5<11> is set, the interrupt posting is disabled.

(continued on next page)

Table 3–40 (Cont.) CSR7 Interrupt Mask Register Description

Field	Description
10	**ATM—AUI/TP Switch Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and AUI/TP pin (CSR5<10>), the interrupt is posted. When reset and AUI/TP (CSR5<10>) is set, the interrupt posting is disabled.
9	**RWM—Receive Watchdog Time-Out Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and receive watchdog time-out (CSR5<9>), the interrupt is posted. When reset and receive watchdog time-out (CSR5<9>) is set, the interrupt posting is disabled.
8	**RSM—Receive Stopped Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and receive stopped (CSR5<8>), the interrupt is posted. When reset and receive stopped (CSR5<8>) is set, the interrupt posting is disabled.
7	**RUM—Receive Buffer Unavailable Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and receive buffer unavailable (CSR5<7>), the interrupt is posted. When reset and receive buffer unavailable (CSR5<7>) is set, the interrupt posting is disabled.
6	**RIM—Receive Interrupt Mask (read, write)** When set together with normal interrupt summary mask (CSR7<16>) and receive interrupt bit (CSR5<6>), the interrupt is posted. When reset and receive interrupt (CSR5<6>) is set, the interrupt posting is disabled.
5	**UNM—Underflow Interrupt Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and transmit underflow (CSR5<5>), the interrupt is posted. When reset and transmit underflow (CSR5<5>) is set, the interrupt posting is disabled.
3	**TJM—Transmit Jabber Time-Out Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and transmit jabber time-out (CSR5<3>), the interrupt is posted. When reset and transmit jabber time-out (CSR5<3>) is set, the interrupt posting is disabled.

(continued on next page)

Table 3–40 (Cont.) CSR7 Interrupt Mask Register Description

Field	Description
2	**TUM—Transmit Buffer Unavailable Mask (read, write)** When set together with normal interrupt summary mask (CSR7<16>) and transmit buffer unavailable (CSR5<2>), the interrupt is posted. When reset and transmit buffer unavailable (CSR5<2>) is set, the interrupt posting is disabled.
1	**TSM—Transmission Stopped Mask (read, write)** When set together with abnormal interrupt summary mask (CSR7<15>) and transmission stopped (CSR5<1>), the interrupt is posted. When reset and transmission stopped (CSR5<1>) is set, the interrupt posting is disabled.
0	**TIM—Transmit Interrupt Mask (read, write)** When set together with normal interrupt summary mask (CSR7<16>) and transmit interrupt (CSR5<0>), the interrupt is posted. When reset and transmit interrupt (CSR5<0>) is set, the interrupt posting is disabled.

Table 3–41 lists the access rules for CSR7.

Table 3–41 CSR7 Access Rules

Category	Description
Value after reset	FFFE0000H
Read access rules	–
Write access rules	–

3.2.1.8 Missed Frame Counter (CSR8)

Figure 3–17 shows the CSR8 bit fields, and Table 3–42 describes the bit fields.

Figure 3–17 CSR8 Missed Frame Counter

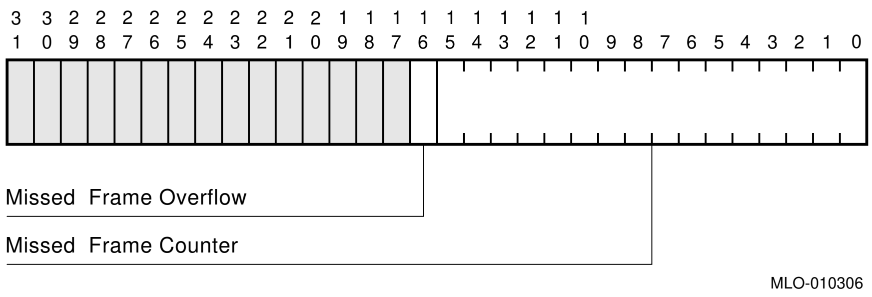


Table 3–42 CSR8 Missed Frame Counter Description

Field	Description
16	**Missed Frame Overflow (read)** Sets when the missed frame counter overflows; resets when CSR8 is read.
15:0	**Missed Frame Counter (read)** Indicates the number of frames discarded because no host receive descriptors were available. The counter clears when read.

Table 3–43 lists the access rules for CSR8.

Table 3–43 CSR8 Access Rules

Category	Description
Value after reset	FFFE0000H
Read access rules	–
Write access rules	Not possible

3.2.1.9 Ethernet Address ROM Register (CSR9)

This register provides an interface to the external Ethernet address ROM. It contains a data byte that is serially read from the ROM. Each read access causes 8-bit, serial, read cycles from the Ethernet address ROM. Writing to this register resets the pointer of the Ethernet address ROM to its first location.

Figure 3–18 shows the Ethernet address ROM register, and Table 3–44 describes the register bit fields.

Figure 3–18 CSR9 Ethernet Address ROM Register

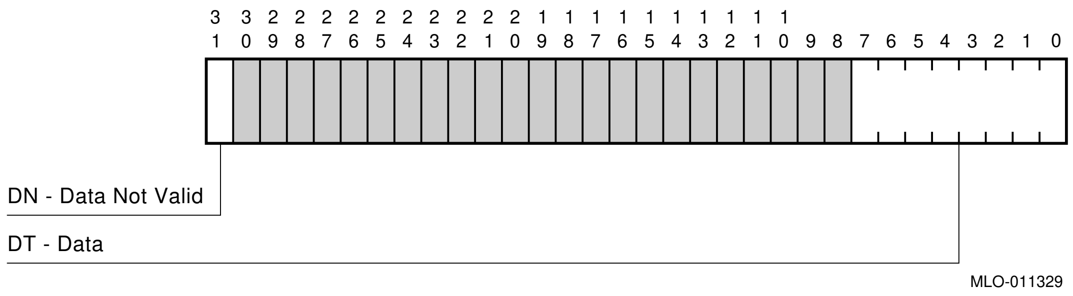


Table 3–44 CSR9 Ethernet Address ROM Register Description

Field	Description
31	**DN—Data Not Valid** When set, indicates that the byte transfer from the ROM is not completed. Subsequent reads must be performed until this bit returns 0, indicating that the data byte field is valid in CSR9<7:0>. Also, the ROM pointer indicates the location of the next byte in ROM.
7:0	**DT—Data** Contains the data byte read from the Ethernet address ROM.

Table 3–45 lists the access rules for CSR9.

Table 3–45 CSR9 Access Rules

Category	Description
Value after reset	UNPREDICTABLE
Read access rules	–
Write access rules	ROM pointer reset

3.2.1.10 Full-Duplex Register (CSR11)

This register contains a 16-bit value for received full-duplex auto configuration support. Figure 3–19 shows the CSR11 bit fields, and Table 3–46 describes the bit fields.

Figure 3–19 CSR11 Full-Duplex Register

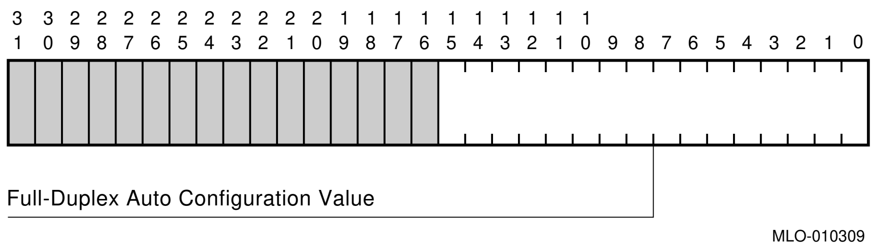


Table 3–46 CSR11 Full-Duplex Register Description

Field	Description
15:0	**Full-Duplex Auto Configuration Value (read, write)** Contains the full-duplex auto configuration value. When this field is set, the 21040 monitors received short frames with a maximum length of 80 bits: 64 bits for preamble and 16 bits of data. If the 16 bits of data match this field, the full-duplex short frame CSR5<11> is set.

Table 3–47 lists the access rules for CSR11.

Table 3–47 CSR11 Access Rules

Category	Description
Value after reset	FFFF0000H
Read access rules	Not possible
Write access rules	Receive process stopped

3.2.2 Serial Interface Attachment CSRs

This section describes the four serial interface attachment (SIA) registers: CSR12, CSR13, CSR14, and CSR15. This description includes different SIA configurations and diagnostic programming. SIA status is maintained in CSR12.

The SIA registers control the functionality and connectivity of the SIA features, enabling various configurations and options. Some of the configurations are used only for diagnostic and testing purposes. The AUI or 10BASE-T selection is done in one of the following ways:

- SIA Auto Configuration—The SIA automatically configures to AUI or 10BASE-T according to the setup described in Table 3–52.
- SIA Pin Configuration—The SIA automatically configures to AUI or 10BASE-T according to the setup described in Table 3–52.
- SIA Full Programming—All three SIA registers (CSR13, CSR14, and CSR15) are programmed with the values required to achieve functionality for special configurations such as full-duplex, loopback, and diagnostic.

Note

Before changing any value in CSR13, CSR14, or CSR15, first perform an SIA software reset by writing CSR13 with all zeros (CSR13 = 00000000H).

Any mode change from SIA_full_programming to SIA_auto_configuration or SIA_pin_configuration must be preceded by setting all SIA registers to their reset values. These values are as follows:

CSR13 = FFFF0000H
CSR14 = FFFFFFFFH
CSR15 = FFFF0000H

3.2.2.1 SIA Status Register (CSR12)

The SIA status register reads SIA pins and internal states. Figure 3–20 shows the CSR12 bit fields, and Table 3–48 describes the bit fields.

Figure 3–20 CSR12 SIA Status Register

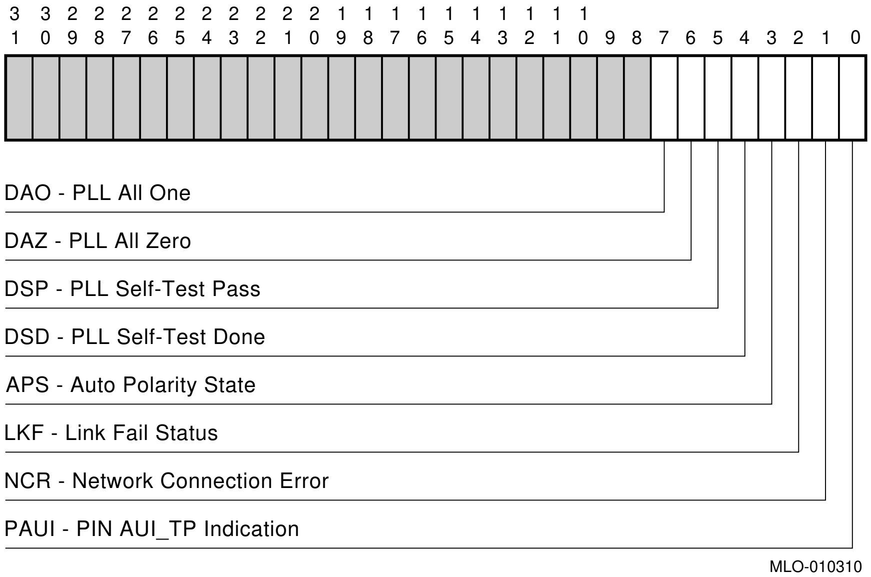


Table 3–48 CSR12 SIA Status Register Description

Field	Description
7	**DAO—PLL All One**
	Diagnostic bit. When set, indicates that all phase lock loop (PLL) sampler synchronizers are asserted high.
6	**DAZ—PLL All Zero**
	Diagnostic bit. When set, indicates that all PLL sampler synchronizers are asserted low.
5	**DSP—PLL Self-Test Pass**
	PLL built-in integrity self-test status indicator (self-test start CSR15<12>).

(continued on next page)

Table 3–48 (Cont.) CSR12 SIA Status Register Description

Field	Description
	PLL self-test pass (CSR12<5>) is valid only if PLL self-test done (CSR12<4>) is read as 1. If PLL self-test done (CSR12<4>) is 1 and PLL self-test pass (CSR12<5>) is 1, the self-test is successful; otherwise, the self-test fails.
4	**DSD—PLL Self-Test Done** Reset when PLL self-test is initiated. Set after self-test completes.
3	**APS—Auto Polarity State** When set, the 10BASE-T polarity is positive. When reset, the 10BASE-T polarity is negative. The received bit stream is inverted by the receiver. (Refer to auto polarity enable CSR14<13> and set polarity plus CSR14<14>.)
2	**LKF—Link Fail Status** When set, the 10BASE-T link test is in fail state. When reset, the 10BASE-T link test is in pass state. During link fail, the 21040 does not transmit any packet to the media. However, any queued packets in the transmit list can be closed by the 21040 with the following set: TDES0<2>—Link fail TDES0<10>—No carrier TDES0<11>—Loss of carrier The 21040 moves from the link fail state to the link pass state when it receives two consecutive packets. The driver receives no indication about these packets. Following this, no transmit packet is pending and no carrier is sensed.
1	**NCR—Network Connection Error** This bit has two meanings: • In AUI, when set, it indicates no carrier. The status resets itself during the next transmission attempt. • In 10BASE-T, this bit sets if no link pass state was established within 2.4 seconds from switching to 10BASE-T (indicating cable failure, for example). If a link pass state was established within 2.4 seconds from switching to 10BASE-T, this bit resets.
0	**PAUI—PIN AUI_TP Indication** When set, indicates that the external AUI_TP PIN is connected to the supply voltage (VDD), requesting AUI interface. When reset, indicates that the external AUI_TP PIN is connected to ground (VSS), requesting 10BASE-T interface.

Table 3–49 lists the access rules for CSR12.

Table 3–49 CSR12 Access Rules

Category	Description
Value after reset	FFFFFFC4H or FFFFFFC5H
Read access rules	–
Write access rules	Read-only register

3.2.2.2 SIA Connectivity Register (CSR13)

CSR13 contains the SIA connectivity control bits that permit the interconnection of different sections within the SIA to allow coverage of the required operation and test options.

Figure 3–21 shows the CSR13 bit fields, and Table 3–50 describes the bit fields.

Figure 3–21 CSR13 SIA Connectivity Register

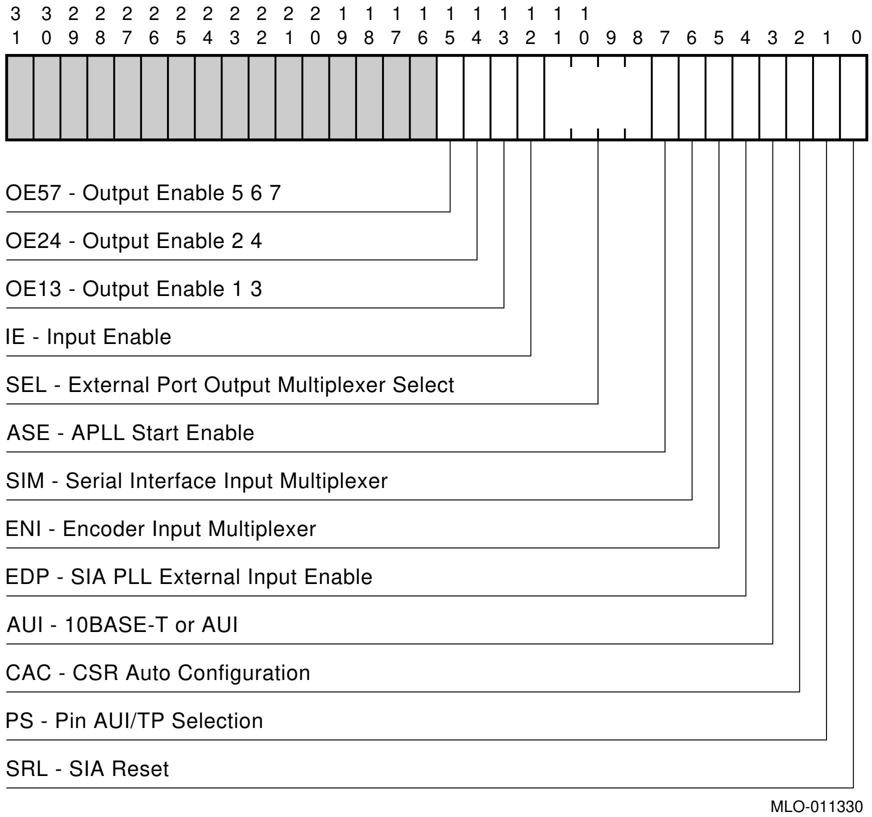


Table 3–50 CSR13 SIA Connectivity Register Description

Field	Description
15	**OE57—Output Enable 5 6 7** Diagnostic bit. When set, pins 5, 6, and 7 of the external SIA interface are selected as *outputs*. When reset, these pins are selected as *inputs* (Table 3–54).
14	**OE24—Output Enable 2 4** Diagnostic bit. When set, pins 2 and 4 of the external SIA interface are selected as *outputs*. When reset, these pins are selected as *inputs* (Table 3–54).
13	**OE13—Output Enable 1 3** Diagnostic bit. When set, pins 1 and 3 of the external SIA interface are selected as *outputs.* When reset, these pins are selected as *inputs* (Table 3–54).
12	**IE—Input Enable** Diagnostic bit. When set, all the pins that were selected as *inputs* by CSR13<15:13> are *enabled.* When reset, all the pins that were selected as *inputs* by CSR13<15:13> are *disabled.*
11:8	**SEL—External Port Output Multiplexer Select** Diagnostic bit. These bits select the internal signals routed to the EXTERNAL_SIA port. The routing control enables this port to resume different functions for normal and diagnostic mode operation. (Table 3–55 lists the signals that can be routed to the port.)
7	**ASE—APLL Start Enable** Diagnostic bit. When set, enables the analog phase lock loop differential mode starter. This bit is used for engineering purposes.
6	**SIM—Serial Interface Input Multiplexer** Diagnostic bit. When set, enables the selection of the external SIA operating mode (Table 3–53).
5	**ENI—Encoder Input Multiplexer** Diagnostic bit. When reset, normal operation mode is selected. When set, it allows direct driving of the encoder inputs from the EXTERNAL_SIA port (Table 3–54).
4	**EDP—SIA PLL External Input Enable** Diagnostic bit. When set, enables direct driving of the PLL from the EXTERNAL_SIA port (Table 3–54).
3	**AUI—10BASE-T or AUI**

(continued on next page)

Table 3–50 (Cont.) CSR13 SIA Connectivity Register Description

Field	Description
	When reset, forces the 21040 to select the 10BASE-T interface. When set to 1, forces the 21040 to select the AUI interface. This bit is valid only if the AUI/TP pin CSR13<1> is reset (Table 3–52).
2	**CAC—CSR Auto Configuration**
	When set, forces CSR13, CSR14, and CSR15 into a predetermined value according to the selection of AUI CSR13<3> bit. This bit is valid only if CSR13<1> is reset (Table 3–52).
1	**PS—Pin AUI/TP Selection**
	When set, forces CSR13, CSR14, and CSR15 into a predetermined value according to the selection of the AUI_TP pin (Table 3–52).
0	**SRL—SIA Reset**
	When reset, resets all the SIA functions and machines. This bit is valid only if the AUI/TP pin selection CSR13<1> and CSR auto configuration CSR13<2> are both reset.

Table 3–51 lists the access rules for CSR13.

Table 3–51 CSR13 Access Rules

Category	Description
Value after reset	FFFF0000H.
Read access rules	If either AUI/TP pin selection CSR13<1> or CSR auto configuration CSR13<2> is set, the value of CSR13 reflects the internal states rather than the values written into the CSR.
Write access rules	CSR13 should be reset to 00000000H before writing to any SIA CSR and released with or after the last CSR write.

3.2.2.3 SIA Operational Modes

The following four bits are used to determine the AUI and 10BASE-T modes of operation. Using these four bits in *SIA_auto_configuration* or *SIA_pin_configuration* format overrides all other CSR13, CSR14, and CSR15 bits (Table 3–52).

CSR13<3>—10BASE-T or AUI selection
CSR13<2>—CSR auto configuration
CSR13<1>—Pin AUI/TP selection
CSR13<0>—SIA reset

Table 3–52 AUI—10BASE-T Selection Using SIA_Auto_Configuration and SIA_Pin_Configuration

CSR13<1>	CSR13<2>	CSR13<3>	AUI_TP Pin	Setting
1	X	X	VDD	AUI Mode—SIA_pin_configuration
1	X	X	VSS	TP Mode—SIA_pin_configuration
0	1	1	X	AUI Mode—SIA_auto_configuration
0	1	0	X	TP Mode—SIA_auto_configuration

Table 3–53 lists the programming of the different SIA modes using CSR13, CSR14, and CSR15. The states of operating mode CSR6<11:10> and full-duplex mode CSR6<9> are also identified.

Table 3–53 Programming of SIA Modes Using CSR13, CSR14, and CSR15

Mode	CSR13	CSR14	CSR15	CSR6<11:10>	CSR6<9>	Note
10BASE-T normal	8F01H	FFFFH	0000H	00	0	LEDs enabled.
10BASE-T normal	EF01H	FFFFH	0000H	00	0	External SIA port enabled for diagnostics.
10BASE-T normal	0F01H	FFFFH	0000H	00	0	External SIA port disabled.
10BASE-T full-duplex	8F01H	FFFDH	0000H	00	1	See Section 5.8.

(continued on next page)

Table 3–53 (Cont.) Programming of SIA Modes Using CSR13, CSR14, and CSR15

Mode	CSR13	CSR14	CSR15	CSR6<11:10>	CSR6<9>	Note
10BASE-T internal loopback	8F01H	FEFBH	0000H	10	0	See Section 5.7.
10BASE-T external loopback	8F01H	F9FDH	0000H	10	0	See Section 5.7.
AUI normal	8F09H	0705H	0006H	00	0	LEDs enabled.
AUI normal	EF09H	0705H	0006H	00	0	External SIA port enabled for diagnostics.
AUI normal	0F09H	0705H	0006H	00	0	External SIA port disabled.
AUI external loopback	8F09H	0705H	0006H	10	0	See Section 5.7.
External SIA	3041H	0000H	0006H	00	0	—
Internal loopback	xxxxH	xxxxH	xxxxH	01	0	See Section 5.7.

3.2.2.4 SIA Port Configurations

The following list contains three sets of output enable bits that control the external SIA port interface.

OE57—Output enable bits 5, 6, and 7 (CSR13<15>)
OE24—Output enable bits 2 and 4 (CSR13<14>)
OE13—Output enable bits 1 and 3 (CSR13<13>)

The SIA port contains three sub-ports. Each sub-port can be used as either an input or output sub-port regardless of how the other sub-ports are configured. Each configuration enables the external SIA port to reflect different functional interfaces for different configurations of the SIA section, mainly for diagnostics. Table 3–54 lists the external SIA port mode selections.

Table 3–54 External SIA Port Mode Selections

Mode	Pin Name	Port Number	CSR13<15:12>	Function
Normal (no LEDs)	–	–	0000	Internal 10BASE-T and AUI interfaces are on.
	EXT_TXEN	1		Tristate, no input
	EXT_TCLK	2		Tristate, no input
	EXT_TX	3		Tristate, no input
	EXT_RCLK	4		Tristate, no input
	EXT_RXEN	5		Tristate, no input
	EXT_RX	6		Tristate, no input
	EXT_CLSN	7		Tristate, no input
Normal (LEDs)	–	–	1000	Internal 10BASE-T and AUI interfaces are on.
	EXT_TXEN	1		Tristate, no input
	EXT_TCLK	2		Tristate, no input
	EXT_TX	3		Tristate, no input
	EXT_RCLK	4		Tristate, no input
	EXT_RXEN	5		Output—network activity LED
	EXT_RX	6		Output—LinkPass LED
	EXT_CLSN	7		Output—AUI/10BASE-T LED
External SIA	–	–	0011	External chip eliminates need for internal SIA functions.
	EXT_TXEN	1		Output
	EXT_TCLK	2		Input
	EXT_TX	3		Output
	EXT_RCLK	4		Input
	EXT_RXEN	5		Input
	EXT_RX	6		Input
	EXT_CLSN	7		Input

(continued on next page)

Table 3–54 (Cont.) External SIA Port Mode Selections

Mode	Pin Name	Port Number	CSR13<15:12>	Function
Trace	–	–	1110	Different internal signals are reflected through the port (Table 3–55).
	EXT_TXEN	1		Output—multiplexed signals
	EXT_TCLK	2		Output—multiplexed signals
	EXT_TX	3		Output—multiplexed signals
	EXT_RCLK	4		Output—multiplexed signals
	EXT_RXEN	5		Output—multiplexed signals
	EXT_RX	6		Output—multiplexed signals
	EXT_CLSN	7		Output—multiplexed signals

The following four select lines for the output multiplexer enable the routing of 56 internal SIA signals to the external SIA port.

SEL0 (CSR13<8>)
SEL1 (CSR13<9>)
SEL2 (CSR13<10>)
SEL3 (CSR13<11>)

Table 3–55 lists the external SIA output multiplexer selection.

Table 3–55 External SIA Output Multiplexer Selection

CSR13<11:8>	Pin Name	Port Number	Signal Name	Function
00XX	–	–	–	21040—SIA interface signals (external SIA mode)
	EXT_TXEN	1	jab_txen	
	EXT_TCLK	2	tclk	
	EXT_TX	3	jab_txd	
	EXT_RCLK	4	rclk	
	EXT_RXEN	5	rxen	

(continued on next page)

Table 3–55 (Cont.) External SIA Output Multiplexer Selection

CSR13<11:8>	Pin Name	Port Number	Signal Name	Function
	EXT_RX	6	rx	
	EXT_CLSN	7	clsn	
01XX	–	–	–	Diagnostics—SIA interface signals
	EXT_TXEN	1	decmx_rxd	
	EXT_TCLK	2	tlp_reset	
	EXT_TX	–	decmx_rxen	
	EXT_RCLK	4	dmux_rxen	
	EXT_RXEN	5	tp_cmp_out	
	EXT_RX	–	poslp_detect_set	
	EXT_CLSN	7	neglp_detect_set	
1111	–	–	–	LED and external driver signals (AUI or TP mode with LEDs)
	EXT_TXEN	1	aui_txen	
	EXT_TCLK	2	sndlnk	
	EXT_TX	3	tp_txen	
	EXT_RCLK	4	clk419_4304m	
	EXT_RXEN	5	xver_active	
	EXT_RX	6	link_pass	
	EXT_CLSN	7	lcsr13_aui	
100X	–	–	–	PLL diagnostic signals
	EXT_TXEN	1	wp_all<5>	
	EXT_TCLK	2	wp_all<6>	
	EXT_TX	3	wp_all<7>	
	EXT_RCLK	4	wp_all<8>	
	EXT_RXEN	5	wp_all<9>	
	EXT_RX	6	wp_all<10>	
	EXT_CLSN	7	wp_all<11>	

(continued on next page)

Table 3–55 (Cont.) External SIA Output Multiplexer Selection

CSR13<11:8>	Pin Name	Port Number	Signal Name	Function
101X	–	–	–	PLL diagnostic signals
	EXT_TXEN	1	apll_cphase<5>	
	EXT_TCLK	2	Reserved	
	EXT_TX	3	Reserved	
	EXT_RCLK	4	Reserved	
	EXT_RXEN	5	Reserved	
	EXT_RX	6	Reserved	
	EXT_CLSN	7	Reserved	
1100	–	–	–	SIA-RxM diagnostic signals
	EXT_TXEN	1	poslpulse	
	EXT_TCLK	2	poseoframe	
	EXT_TX	3	neglpulse	
	EXT_RCLK	4	negeoframe	
	EXT_RXEN	5	colpulsm_on	
	EXT_RX	6	rcvpulsp_on	
	EXT_CLSN	7	rcvpulsp_on	
1101	–	–	–	SIA-RxM machine diagnostic signals
	EXT_TXEN	1	aui_clsn	
	EXT_TCLK	2	rcv_pulse	
	EXT_TX	3	clr_dtct	
	EXT_RCLK	4	col_pulsem	
	EXT_RXEN	5	rcvff1	
	EXT_RX	6	rcvff2	
	EXT_CLSN	7	rcvff4	
1110	–	–	–	Link test and other diagnostic signals
	EXT_TXEN	1	plsmaxtmr2	
	EXT_TCLK	2	plsmintmr2	

(continued on next page)

Table 3–55 (Cont.) External SIA Output Multiplexer Selection

CSR13<11:8>	Pin Name	Port Number	Signal Name	Function
	EXT_TX	3	plsendcnt	
	EXT_RCLK	4	eoftmr	
	EXT_RXEN	5	txwatch_exp$ss	
	EXT_RX	6	rlocked$ss	
	EXT_CLSN	7	aui_tpc$ss	

3.2.2.5 SIA Transmit and Receive Register (CSR14)

CSR14 configures the SIA transmitter and receiver operating modes. Figure 3–22 shows the CSR14 bit fields, and Table 3–56 describes the bit fields. This register is mainly used for diagnostic purposes.

Figure 3–22 CSR14 SIA Transmit and Receive Register

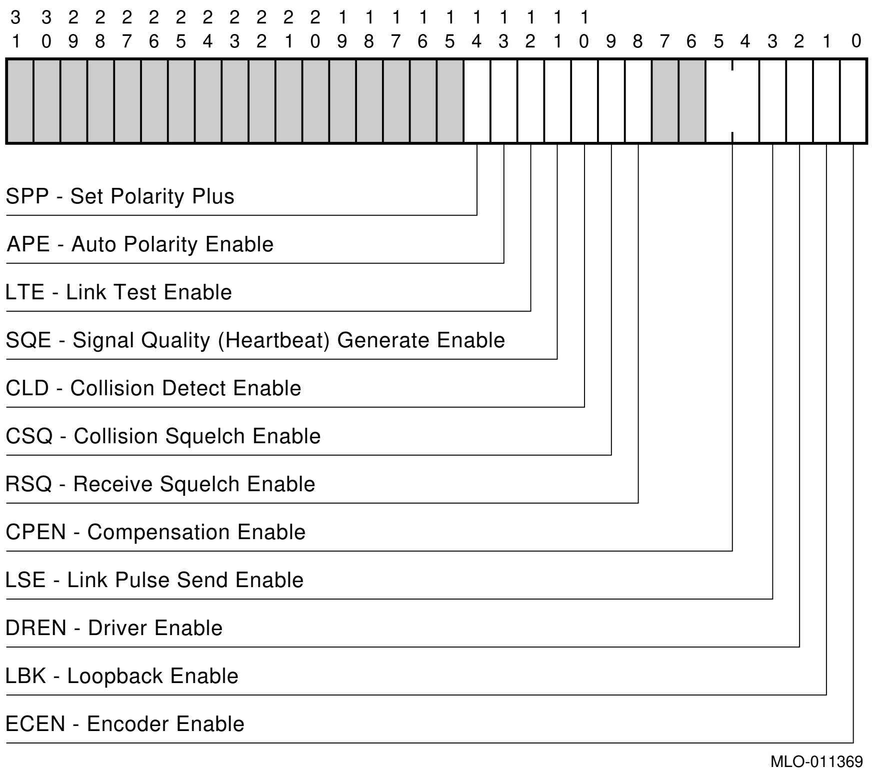


Table 3–56 CSR14 SIA Transmit and Receive Register Description

Field	Description
14	**SPP—Set Polarity Plus** When reset and auto polarity enable (CSR14 <13>) is reset, the polarity of the incoming data is switched. This feature can be used by the driver to reverse polarity of incoming packets; otherwise, this bit should be set. This bit is valid only in 10BASE-T mode.
13	**APE—Auto Polarity Enable** When set and link test enable CSR14<12> is also set, the auto polarity function logic is enabled (Section 7.1.7). When reset, the polarity is determined by set polarity plus (CSR14<14>). When link test enable (CSR14<12>) is reset, this bit (CSR14<13>) should be also reset. This bit is valid only in 10BASE-T mode.
12	**LTE—Link Test Enable** When set, the link test function logic is enabled. In AUI mode, it should be reset. In 10BASE-T mode, resetting this bit forces the link test function to link pass state.
11	**SQE—Signal Quality (Heartbeat) Generate Enable** Controls the signal quality (SQE) generator ability to imitate external medium attachment unit (MAU) behavior. When set, a short heartbeat signal is generated after the conclusion of a transmitted packet. In 10BASE-T mode, SQE (CSR14<11>) should be set; otherwise, a heartbeat fail (TDES0<7>) is set. In AUI mode, SQE (CSR14<11>) should be reset.
10	**CLD—Collision Detect Enable** When set, the collision detect logic is enabled.
9	**CSQ—Collision Squelch Enable** When set, the AUI collision receivers are active. This bit is valid only when AUI is selected.
8	**RSQ—Receive Squelch Enable** When set, the AUI or 10BASE-T receivers are active in accordance with the selected mode.
5:4	**CPEN—Compensation Enable** Table 3–58 defines twisted-pair compensation behavior. This bit is valid only in 10BASE-T mode.
3	**LSE—Link Pulse Send Enable** When set, the link pulse generator is enabled. In AUI mode, this bit should be reset.

(continued on next page)

Table 3–56 (Cont.) CSR14 SIA Transmit and Receive Register Description

Field	Description
2	**DREN—Driver Enable** When set, the transmit SIA driver is enabled for AUI or 10BASE-T operation. When reset, the transmit driver is disabled, preventing the data and link pulse transmission to the external wires.
1	**LBK—Loopback Enable** Enables loopback operation in SIA (Table 3–59 and Section 5.7.3).
0	**ECEN—Encoder Enable** When set, the transmit data encoder is enabled, and the encoded data is transferred to the output drivers. When reset, the transmit data encoder is disabled, and the encoded data is blocked from propagating to the output drivers.

Table 3–57 lists the access rules for CSR14.

Table 3–57 CSR14 Access Rules

Category	Description
Value after reset	FFFFFFFFH.
Read access rules	In both SIA_auto_configuration and SIA_pin_configuration modes, a CSR14 read reflects internal states, rather than the values written into the CSR.
Write access rules	CSR14 should be reset to 00000000H before writing any SIA CSR and released with or just after last CSR write.

Table 3–58 lists the compensation field (CSR14<5:4>) definitions.

Table 3–58 Twisted-Pair Compensation Behavior

CSR14<5:4> Value	Transmitter Output
00, 01	Compensation Disabled Mode—Twisted-pair driver does not compensate for 10 megahertz versus 5 megahertz media attenuation (differential voltages are bound between 1.5 volts and 2.1 volts).
10	High Power Mode—Twisted-pair driver drives only high-differential voltage (between 2.2 volts and 2.8 volts).
11	Normal Compensation Mode—Driver compensates for 10 megahertz versus 5 megahertz media attenuation by driving high-differential voltage for transients and driving low if the signal is stable for more than 50 nanoseconds.

3.2.2.6 SIA Mode Programming

Table 3–59 lists normal, full-duplex, and loopback programming. To monitor these modes, the following signals are used in the table.

CSR6<11:10>—Operating mode
CSR6<9>—Full-duplex operating mode
CSR14<1>—Loopback enable
CSR14<10>—Collision detect enable
CSR14<8>—Receive squelch enable

Table 3–59 Normal, Full-Duplex, and Loopback Programming

Mode	Note	CSR6 <11:10>	CSR6 <9>	CSR14 <1>	CSR14 <10>	CSR14 <8>
TP normal	Packets are looped back from encoder output to PLL input.	00	0	1	1	1
TP full-duplex	Loopback is disabled at SIA level allowing PLL to lock onto incoming packets.	00	1	0	X	1
TP on-chip loopback	Packets are looped back from encoder output to decoder input.	10	0	1	0	0
TP off-chip loopback	Requires external shunt for board testing.	10	0	0	0	1
AUI normal	The external MAU performs loopback, but the receive and transmit machines do not work simultaneously.	00	0	0	1	1
AUI external loopback	Diagnostic checks up-to-MAU path integrity. If collision, no comparison is done.	10	0	0	1	1
External SIA normal	The external MAU performs loopback, but the receive and transmit machines do not work simultaneously.	00	0	0	0	0
External SIA loopback	Diagnostic checks up-to-MAU path integrity. If collision, no comparison is done.	10	0	0	0	0
Internal loopback	For internal diagnostics on any mode. The transmit packet is looped back at the MAC level.	01	0	X	X	X

Note

Internal loopback is not permitted while in full-duplex mode.

3.2.2.7 SIA General Register (CSR15)

Figure 3–23 shows the CSR15 bit fields, and Table 3–60 describes the bit fields. This register is mainly used for diagnostic purposes.

Figure 3–23 CSR15 SIA General Register

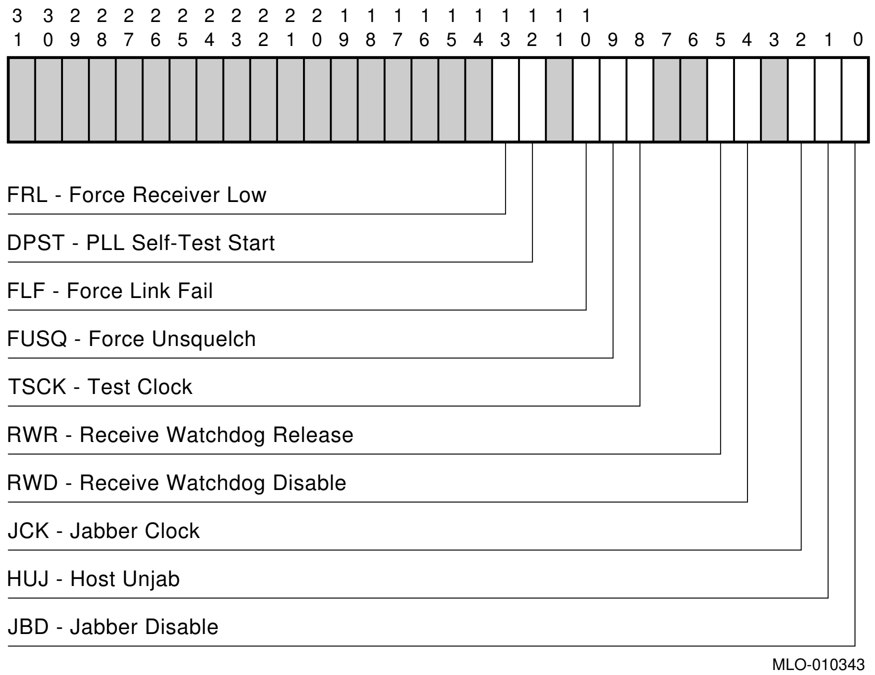


Table 3–60 CSR15 SIA General Register Description

Field	Description
13	**FRL—Force Receiver Low** Testing feature that forces the RX input to PLL samplers to a constant low. It is used to detect stacked samplers.
12	**DPST—PLL Self-Test Start** Testing feature that starts the PLL built-in integrity self-test. The status of the test result is marked by PLL self-test done (CSR12<4>) and PLL self-test pass (CSR12<5>) respectively.
11	**FLF—Force Link Fail** Testing feature that forces a link fail state and resets both link test and auto polarity detector.
9	**FUSQ—Force Unsquelch** Testing feature that asserts the receiver RCVEN signal for testing purposes.
8	**TSCK—Test Clock** Testing feature that affects certain SIA clocks. When test clock is asserted, it increases, by 1024, all SIA clocks with a cycle time longer than 2 microseconds, to increase events during product testing. Test clock assertion also causes the jabber timer to expire 1000 times faster (transmitted packet is retried and not stopped.)
5	**RWR—Receive Watchdog Release** Defines the time interval *no carrier* from receive watchdog expiration until re-enabling the receive channel. When set, the receive watchdog is released 40- to 48-bit-times from the last carrier deassertion. When reset, the receive watchdog is released 16- to 24-bit-times from the last carrier deassertion.
4	**RWD—Receive Watchdog Disable** When set, the receive watchdog counter is disabled. Receive carriers longer than 2560 bytes are guaranteed to cause the watchdog counter to time-out. Packets shorter than 2048 bytes are guaranteed to pass.
2	**JCK—Jabber Clock** When set, transmission is cut after 2048 to 2560 bytes are transmitted (1.6 to 2.0 milliseconds). When reset, transmission is cut after 26 milliseconds to 33 milliseconds.
1	**HUJ—Host Unjab** Defines the time interval between transmit jabber expiration until re-enabling of the transmit channel. When set, the transmit channel is released immediately after the jabber expiration. When reset, the transmit jabber is released 365 to 420 milliseconds after jabber expiration.
0	**JBD—Jabber Disable** When set, the transmit jabber function is disabled.

Table 3–61 lists the access rules for CSR15.

Table 3–61 CSR15 Access Rules

Category	Description
Value after reset	FFFF0000H.
Read access rules	In SIA_auto_configuration and SIA_pin_configuration modes, CSR15 read reflects internal states, rather than the values written into the CSR.
Write access rules	CSR15 should be reset to 00000000H before writing any SIA CSR and released with or just after the last CSR write.

4

Host Communication Area

Descriptor lists and data buffers, collectively called the host communication area, reside in the host memory and manage the actions and status related to buffer management.

4.1 Data Communication

The 21040 and the driver communicate through two data structures:

- Command and status registers (CSRs) described in Chapter 3.
- Descriptor lists and data buffers described in this chapter.

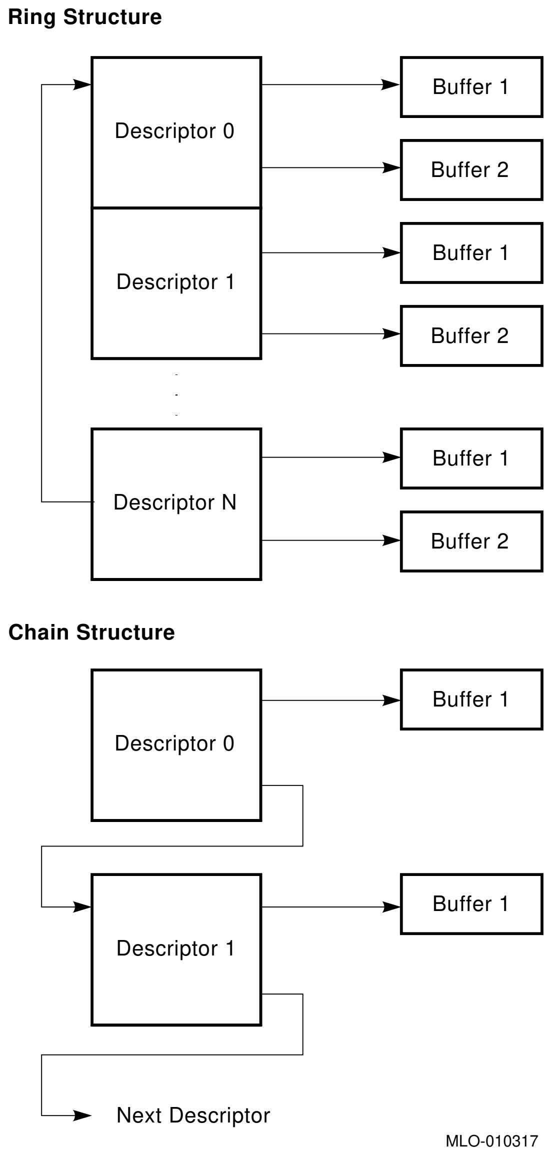
4.2 Descriptor Lists and Data Buffers

The 21040 transfers frame data to and from the receive and transmit buffers in host memory. Descriptors that reside in the host memory act as pointers to these buffers.

There are two descriptor lists, one for receive and one for transmit. The base address of each list is written into CSR3 and CSR4, respectively. A descriptor list is forward-linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both the receive and transmit descriptors (RDES1<24> and TDES1<24>). The descriptor lists reside in the host *physical* memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, and not contiguous in memory (Figure 4–1).

A data buffer consists of either an entire frame or part of a frame, but it cannot exceed a single frame. Buffers contain only data; buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. Data chaining can be enabled or disabled. Data buffers reside in host *physical* memory space.

Figure 4–1 Descriptor Ring and Chain Structures



4.2.1 Receive Descriptors

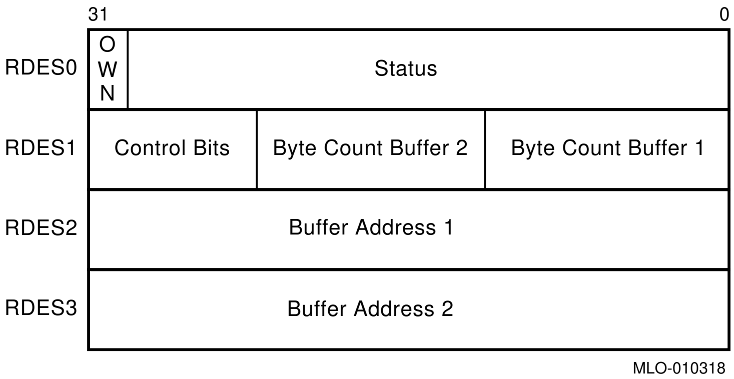
Figure 4–2 shows the receive descriptor format.

Note

Descriptors and receive buffer addresses must be longword-aligned.

Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory management schemes.

Figure 4–2 Receive Descriptor Format



4.2.1.1 Receive Descriptor 0 (RDES0)

RDES0 contains the received frame status, the frame length, and the descriptor ownership information. Figure 4–3 shows the RDES0 bit fields, and Table 4–1 describes the bit fields.

Figure 4–3 RDES0 Receive Descriptor 0

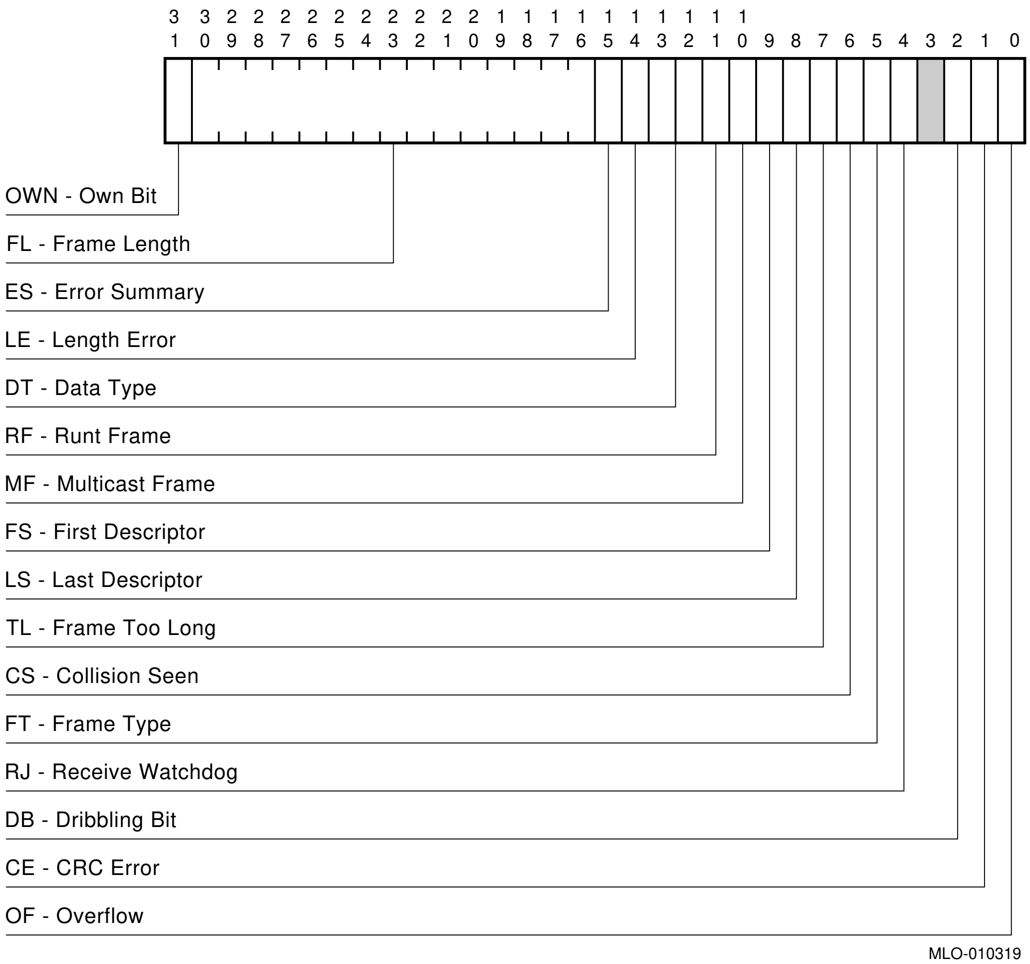


Table 4–1 RDES0 Receive Descriptor 0 Description

Field	Description
31	**OWN—Own Bit** When set, indicates that the descriptor is owned by the 21040. When reset, indicates that the descriptor is owned by the host. The 21040 clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30:16	**FL—Frame Length** Indicates the length in bytes of the received frame including the cyclic redundancy check (CRC). This field is valid only when last descriptor (RDES<8>) is set and length error (RDES0<14>) is reset.
15	**ES—Error Summary** Indicates the logical OR of the following RDES0 bits: RDES0<0>—Overflow RDES0<1>—CRC error RDES0<6>—Collision seen RDES0<7>—Frame to long RDES0<11>—Runt frame RDES0<14>—Length error This field is valid only when last descriptor (RDES<8>) is set.
14	**LE—Length Error** When set, indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers and indicates that the 21040 does not own the next descriptor. The frame is truncated. This field is valid only when last descriptor (RDES<8>) is set.
13:12	**DT—Data Type** This field is valid only when last descriptor (RDES<8>) is set.

(continued on next page)

Table 4–1 (Cont.) RDES0 Receive Descriptor 0 Description

Field	Description
	Indicates the type of frame the buffer contains. 00—Serial received frame. 01—Internal loopback frame. 10—External loopback frame or serial received frame. The 21040 does not differentiate between loopback and serial received frames; therefore, this information is global and reflects only the operating mode (CSR6<11:10>). 11—Reserved.
11	**RF—Runt Frame** When set, indicates that this frame was damaged by a collision or premature termination before the collision window had passed. Runt frames are passed on to the host only if the pass bad frames bit (CSR6<3>) is set. This field is valid only when last descriptor (RDES<8>) is set and overflow (RDES0<0>) is reset.
10	**MF—Multicast Frame** When set, indicates that this frame has a multicast address. This field is valid only when last descriptor (RDES<8>) is set.
9	**FS—First Descriptor** When set, indicates that this descriptor contains the first buffer of a frame. If the buffer size of the first buffer is 0, the second buffer contains the beginning of the frame. If the buffer size of the second buffer is also 0, the second descriptor contains the beginning of the frame.
8	**LS—Last Descriptor** When set, indicates that the buffers pointed to by this descriptor, are the last buffers of the frame.
7	**TL—Frame Too Long** When set, indicates that the frame length exceeds the maximum Ethernet specified size of 1518 bytes. This field is valid only when last descriptor (RDES<8>) is set.

Note

Frame too long is only a frame length indication and does not cause any frame truncation.

(continued on next page)

Table 4–1 (Cont.) RDES0 Receive Descriptor 0 Description

Field	Description
6	**CS—Collision Seen** When set, indicates that the frame was damaged by a collision that occurred after the 64 bytes following the start frame delimiter (SFD). This is a late collision. This field is valid only when last descriptor (RDES<8>) is set.
5	**FT—Frame Type** When set, indicates that the frame is an Ethernet type frame (frame length field is greater than 1500 bytes). When clear, indicates that the frame is an IEEE 802.3 frame. This field is not valid for runt frames of less than 14 bytes. This field is valid only when last descriptor (RDES<8>) is set.
4	**RJ—Receive Watchdog** When set, indicates that the receive watchdog timer expired while receiving the current packet with length greater than 2048–2560 bytes. Receive watchdog time-out (CSR5<9>) is set. When RDES0<4> is set, the frame length field in RDES0<30:16> is not valid and length error (RDES0<14>) is not set. This field is valid only when last descriptor (RDES<8>) is set.
2	**DB—Dribbling Bit** When set, indicates that the frame contained a non-integer multiple of 8 bits. This error is reported only if the number of dribbling bits in the last byte is greater than 2. This field is not valid if either collision seen (RDES0<6>) or runt frame RDES0<11> are set. This field is valid only when last descriptor (RDES<8>) is set.
1	**CE—CRC Error** When set, indicates that a cyclic redundancy check (CRC) error occurred on the received frame. The CRC check is performed independent of a dribbling bit (RDES0<2>) error. However, only whole bytes are run through the CRC logic. Consequently, received frames with up to 6 dribbling bits cause this bit to be set. This field is valid only when last descriptor (RDES<8>) is set.
0	**OF—Overflow**

(continued on next page)

Table 4–1 (Cont.) RDES0 Receive Descriptor 0 Description

Field	Description
	When set, indicates received data in this descriptor's buffer were truncated due to FIFO overflow. This generally occurs if 21040 bus requests are not granted before the internal receive FIFO fills up. This field is valid only when last descriptor (RDES<8>) is set.

4.2.1.2 Receive Descriptor 1 (RDES1)

Figure 4–4 shows the RDES1 bit fields, and Table 4–2 describes the bit fields.

Figure 4–4 RDES1 Receive Descriptor 1

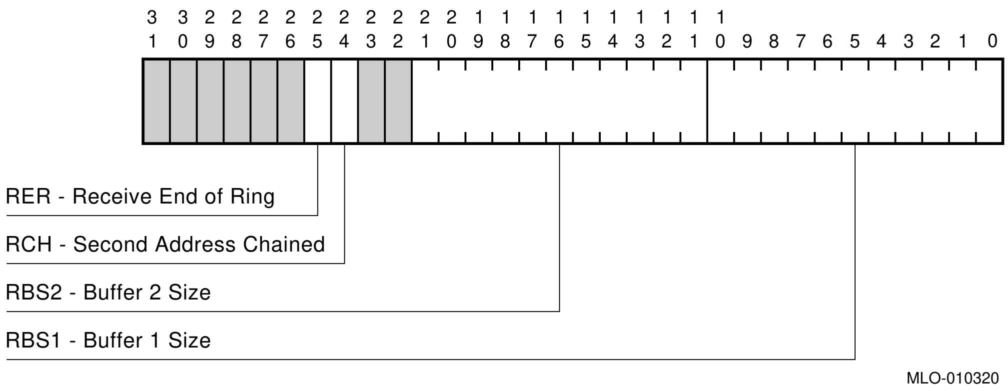


Table 4–2 RDES1 Receive Descriptor 1 Description

Field	Description
25	**RER—Receive End of Ring** When set, indicates that the descriptor list reached its final descriptor. The 21040 returns to the base address of the list (Section 3.2.1.4) creating a descriptor ring.
24	**RCH—Second Address Chained** When set, indicates that the second address in the descriptor is the next descriptor address, rather than the second buffer address. RDES1<25> takes precedence over RDES1<24>.
21:11	**RBS2—Buffer 2 Size** Indicates the size, in bytes, of the second data buffer. If this field is 0, the 21040 ignores this buffer and fetches the next descriptor. The buffer size must be a multiple of 4. This field is not valid if RDES1<24> is set.
10:0	**RBS1—Buffer 1 Size** Indicates the size, in bytes, of the first data buffer. If this field is 0, the 21040 ignores this buffer and uses buffer 2. The buffer size must be a multiple of 4.

4.2.1.3 Receive Descriptor 2 (RDES2)

Figure 4–5 shows the RDES2 bit field, and Table 4–3 describes the bit field.

Figure 4–5 RDES2 Receive Descriptor 2

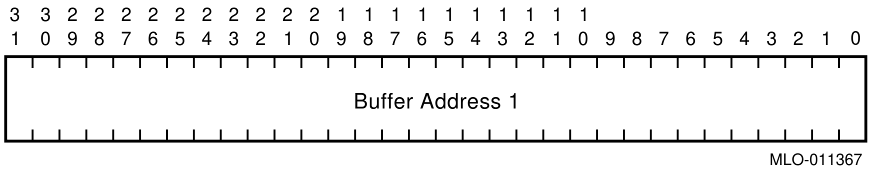


Table 4–3 RDES2 Receive Descriptor 2 Description

Field	Description
31:0	**Buffer Address 1** Indicates the physical address of buffer 1. The buffer must be longword-aligned (RDES2<1:0> = 00).

4.2.1.4 Receive Descriptor 3 (RDES3)

Figure 4–6 shows the RDES3 bit field, and Table 4–4 describes the bit field.

Figure 4–6 RDES3 Receive Descriptor 3

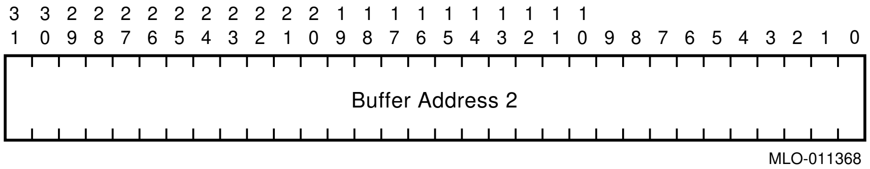


Table 4–4 RDES3 Receive Descriptor 3 Description

Field	Description
31:0	**Buffer Address 2** Indicates the physical address of buffer 2. The buffer must be longword-aligned (RDES3<1:0> = 00).

4.2.1.5 Receive Descriptor Status Validity

Table 4–5 lists the validity of the receive descriptor status bits in relation to the reception completion status. The following list defines the abbreviations used in the table.

RF—Runt frame (RDES0<11>)
CS—Collision seen (RDES0<6>)
FT—Frame type (RDES0<5>)
DB—Dribbling bit (RDES0<2>)
CE—CRC error (RDES0<1>)
ES—Error summary (RDES0<15>)
LE—Length error (RDES0<14>)
DT—Data type (RDES0<13:12>)
FS—First descriptor (RDES0<9>)
LS—Last descriptor (RDES0<8>)
FL—Frame length (RDES0<30:16>)
OF—Overflow (RDES0<0>)

Table 4–5 Receive Descriptor Status Validity

	Receive Status Report					
Reception Status	**RF**	**CS**	**FT**	**DB**	**CE**	**(ES, LE, DT, FS, LS, FL, OF)**
Overflow	NV	NV	V	NV	NV	V
Collision after 512 bits	V	V	V	NV	NV	V
Runt frame	V	V	V	NV	NV	V
Runt frame less than 14 bytes	V	V	NV	NV	NV	V
Watchdog time-out	V	NV	V	NV	NV	V

V—Valid
NV—Not valid

4.2.2 Transmit Descriptors

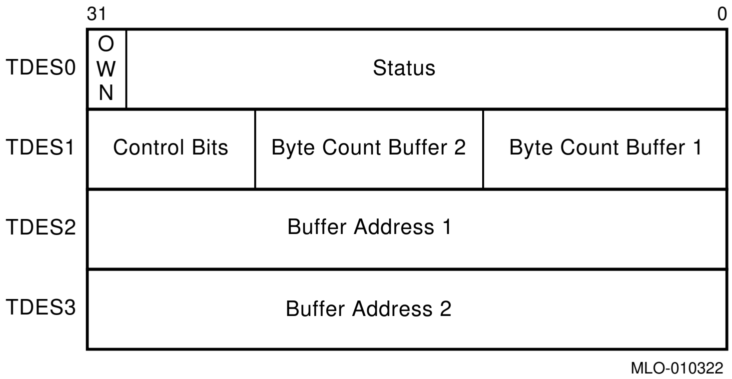
Figure 4–7 shows the Transmit descriptor format.

Note

Descriptors and receive buffer addresses must be longword-aligned.

Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory management schemes.

Figure 4–7 Transmit Descriptor Format



4.2.2.1 Transmit Descriptor 0 (TDES0)

TDES0 contains transmitted frame status and descriptor ownership information. Figure 4–8 shows the TDES0 bit fields, and Table 4–6 describes the bit fields.

Figure 4–8   TDES0 Transmit Descriptor 0

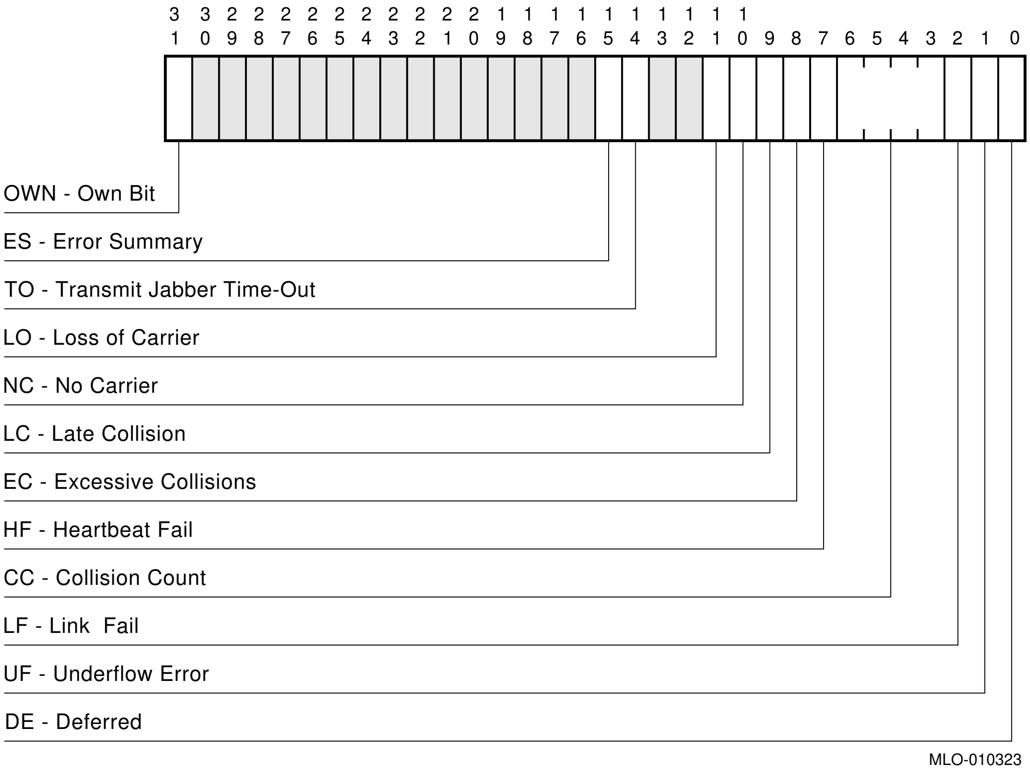


Table 4–6 TDES0 Transmit Descriptor 0 Description

Field	Description
31	**OWN—Own Bit** When set, indicates that the descriptor is owned by the 21040. When cleared, indicates that the descriptor is owned by the host. The 21040 clears this bit either when it completes the frame transmission or when the buffers that are allocated in the descriptor are empty. The ownership bit of the first descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between the 21040 fetching a descriptor and the driver setting an ownership bit.
15	**ES—Error Summary** Indicates the logical OR of the following bits: TDES0<1>—Underflow error TDES0<2>—Link fail TDES0<8>—Excessive collisions TDES0<9>—Late collision TDES0<10>—No carrier TDES0<11>—Loss of carrier TDES0<14>—Transmit jabber time-out
14	**TO—Transmit Jabber Time-Out** When set, indicates that the transmit jabber timer timed out and that the 21040 transmitter was babbling. The transmit jabber time-out interrupt CSR5<3> is set. The transmission process is *aborted* and placed in the STOPPED state. When TDES0<14> is set, any heartbeat fail indication (TDES0<7>) is not valid.
11	**LO—Loss of Carrier** When set, indicates loss of carrier during transmission (possible short circuit in the Ethernet cable). Not valid in internal loopback mode (CSR6<11:10>=01).
10	**NC—No Carrier** When set, indicates that the carrier signal from the transceiver was not present during transmission (possible problem in the transceiver or the transceiver cable). Not valid in internal loopback mode (CSR6<11:10>=01).
9	**LC—Late Collision**

(continued on next page)

Table 4–6 (Cont.) TDES0 Transmit Descriptor 0 Description

Field	Description
	When set, indicates that the frame transmission was aborted due to collision occurring after the collision window of 64 bytes. Not valid if underflow error (TDES0<1>) is set.
8	**EC—Excessive Collisions**
	When set, indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame.
7	**HF—Heartbeat Fail**
	When set, indicates a heartbeat collision check failure (the transceiver failed to return a collision pulse as a check after the transmission). Some transceivers do not support heartbeat: in this case, heartbeat fail is always set but not valid. This bit is not valid if underflow error (TDES0<1>) is set. In 10BASE-T mode, if signal quality enable (CSR14<11>) is reset, heartbeat fail (TDES0<7>) is always set. In AUI mode or external SIA mode, signal quality enable (CSR14<11>) has no effect on heartbeat fail (TDES0<7>). On the second transmission attempt, after the first transmission was aborted due to collision, the 21040 does not check heartbeat fail and (TDES0<7>) is reset.
6:3	**CC—Collision Count**
	This 4-bit counter indicates the number of collisions that occurred before the frame was transmitted. Not valid when the excessive collisions bit (TDES0<8>) is also set.
2	**LF—Link Fail**
	When set, indicates that the link fail test failed and the twisted-pair line cannot transmit the packet. (See link fail status CSR12<2>.)
1	**UF—Underflow Error**
	When set, indicates that the transmitter aborted the message because data arrived late from memory. Underflow error indicates that the 21040 encountered an empty transmit FIFO while transmitting a frame. The transmission process enters the suspended state and sets both transmit underflow (CSR5<5>) and transmit interrupt (CSR5<0>).
0	**DE—Deferred**
	When set, indicates that the 21040 had to defer while ready to transmit a frame because the carrier was asserted.

4.2.2.2 Transmit Descriptor 1 (TDES1)

Figure 4–9 shows the TDES1 bit fields, and Table 4–7 describes the bit fields.

Figure 4–9 TDES1 Transmit Descriptor 1

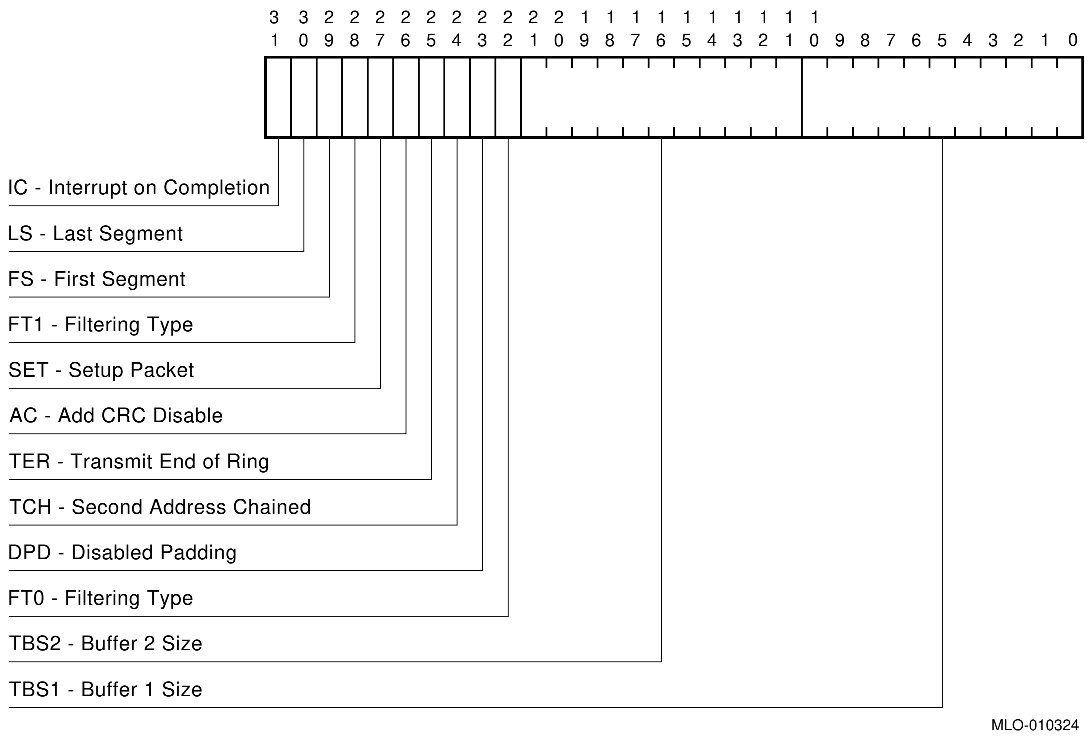


Table 4–7 TDES1 Transmit Descriptor 1 Description

Field	Description
31	**IC—Interrupt on Completion** When set, the 21040 sets transmit interrupt (CSR5<0>) after the present frame has been transmitted. It is valid only when last segment (TDES1<30>) is set.
30	**LS—Last Segment** When set, indicates that the buffer contains the last segment of a frame.

(continued on next page)

Table 4–7 (Cont.) TDES1 Transmit Descriptor 1 Description

Field	Description
29	**FS—First Segment** When set, indicates that the buffer contains the first segment of a frame.
28	**FT1—Filtering Type** Table 4–8 lists the filtering types.
27	**SET—Setup Packet** When set, indicates that the current descriptor is a setup frame descriptor (Section 4.2.3).
26	**AC—Add CRC Disable** When set, the 21040 does not append the cyclic redundancy check (CRC) to the end of the transmitted frame. This field is valid only when first segment (TDES1<29>) is set.
25	**TER—Transmit End of Ring** When set, indicates that the descriptor pointer has reached its final descriptor. The 21040 returns to the root address of the list (Section 3.2.1.4). This creates a descriptor ring.
24	**TCH—Second Address Chained** When set, indicates that the second address in the descriptor is the next descriptor address, rather than the second buffer address. Transmit end of ring (TDES1<25>) takes precedence over second address chained (TDES1<24>).
23	**DPD—Disabled Padding** When set, the 21040 does not automatically add a padding field to a packet shorter than 64 bytes. When reset, the 21040 automatically adds a padding field and a CRC field to a packet shorter than 64 bytes. The CRC field is added despite the state of the add CRC disable (TDES1<26>) flag.
22	**FT0—Filtering Type** Table 4–8 lists the filtering types.
21:11	**TBS2—Buffer 2 Size** Indicates the size, in bytes, of the second data buffer. If this field is 0, the 21040 ignores this buffer and fetches the next descriptor. This field is not valid if second address chained (TDES1<24>) is set.

(continued on next page)

Table 4–7 (Cont.) TDES1 Transmit Descriptor 1 Description

Field	Description
10:0	**TBS1—Buffer 1 Size** Indicates the size, in bytes, of the first data buffer. If this field is 0, the 21040 ignores this buffer and uses buffer 2.

Table 4–8 lists the filtering types. Section 3.2.1.6 provides additional information about filtering.

Table 4–8 Filtering Types

FT1	FT0	Description
0	0	**Perfect filtering** The 21040 interprets the descriptor buffer as a setup perfect table of 16 addresses and sets the 21040 filtering mode to perfect filtering. This field is valid only when setup packet (TDES1<27>) is set (Table 3–38).
0	1	**Hash filtering** The 21040 interprets the descriptor buffer as a setup hash table of 512-bit-plus-one perfect address. If an incoming receive packet destination address is a multicast address, the 21040 executes an imperfect address filtering compared with the hash table. However, if the incoming receive packet destination address is a physical address, the 21040 executes a perfect filtering compared with the perfect address. This field is valid only when setup packet (TDES1<27>) is set (Table 3–38).
1	0	**Inverse filtering** The 21040 interprets the descriptor buffer as a setup perfect table of 16 addresses and sets the 21040 filtering mode to inverse filtering. The 21040 receives the incoming frames with destination addresses that do not match the perfect addresses and rejects the frames with destination addresses that match one of the perfect addresses. This field is valid only when setup packet (TDES1<27>) is set (Table 3–38).
1	1	**Hash-only filtering** The 21040 interprets the descriptor buffer as a setup 512-bit hash table. If an incoming receive packet destination address is multicast or physical, the 21040 executes an imperfect address filtering against the hash table. This field is valid only when setup packet (TDES1<27>) is set (Table 3–38).

4.2.2.3 Transmit Descriptor 2 (TDES2)

Figure 4–10 shows the TDES2 bit field, and Table 4–9 describes the bit field.

Figure 4–10 TDES2 Transmit Descriptor 2

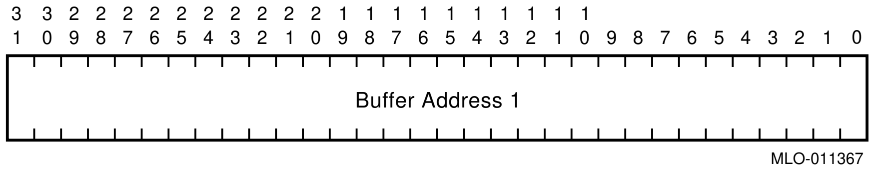


Table 4–9 TDES2 Transmit Descriptor 2 Description

Field	Description
31:0	**Buffer Address 1** Physical address of buffer 1.

4.2.2.4 Transmit Descriptor 3 (TDES3)

Figure 4–11 shows the TDES3 bit field, and Table 4–10 describes the bit field.

Figure 4–11 TDES3 Transmit Descriptor 3

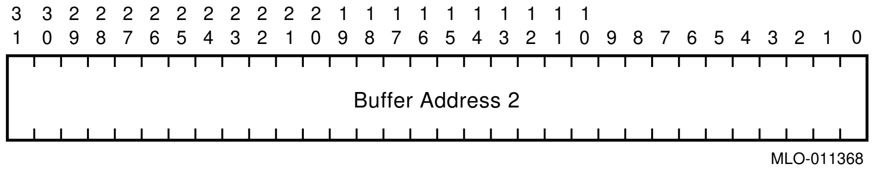


Table 4–10 TDES3 Transmit Descriptor 3 Description

Field	Description
31:0	**Buffer Address 2** Physical address of buffer 2.

4.2.2.5 Transmit Descriptor Status Validity

Table 4–11 lists the validity of the transmit descriptor status bits during transmission completion status. The following list defines the abbreviations used in the table.

LO—Loss of carrier (TDES0<11>)
NC—No carrier (TDES0<10>)
LC—Late collision (TDES0<9>)
EC—Excessive collisions (TDES0<8>)
HF—Heartbeat fail (TDES0<7>)
CC—Collision count (TDES0<6:3>)
ES—Error summary (TDES0<15>)
TO—Transmit jabber time-out (TDES0<14>)
UF—Underflow error (TDES0<1>)
DE—Deferred (TDES0<0>)

Table 4–11 Transmit Descriptor Status Validity

	Transmit Status Report						
Transmission Status	**LO**	**NC**	**LC**	**EC**	**HF**	**CC**	**(ES, TO, UF, DE)**
Underflow	NV	NV	V	V	NV	V	V
Excessive collisions	V	V	V	V	V	NV	V
Watchdog time-out	NV	V	NV	NV	NV	V	V
Internal loopback	NV	NV	V	V	NV	V	V

V—Valid
NV—Not valid

4.2.3 Setup Frame

A setup frame defines the 21040 Ethernet destination addresses. These addresses filter all incoming frames. The setup frame is *never* transmitted on the Ethernet wire nor is it looped back to the receive list. When processing the setup frame, the receiver logic temporarily disengages from the Ethernet wire. The setup frame size must be *exactly* 192 bytes.

Note

The setup frame must be allocated in a single buffer that is longword-aligned. First segment (TDES1<29>) and last segment (TDES1<30>) must both be 0.

When the setup frame load is completed, the 21040 closes the setup frame descriptor by clearing its ownership bit and setting all other bits to 1.

4.2.3.1 First Setup Frame

A setup frame must be processed before the reception process is started, except when it operates in promiscuous filtering mode.

4.2.3.2 Subsequent Setup Frames

Subsequent setup frames may be queued to the 21040 regardless of the reception process state. The only requirement for setup frame processing is that the transmission process must be *running*. The setup frame is processed after all preceding frames have been transmitted and the current frame reception, if any, is completed.

The setup frame does not affect the reception process state, but during setup frame processing, the 21040 is disengaged from the Ethernet wire.

4.2.3.3 Perfect Filtering Setup Frame Buffer

This section describes how the 21040 interprets a setup frame buffer in perfect filtering mode (CSR6<0> = 0).

The 21040 can store 16 destination addresses (full 48-bit Ethernet addresses). The 21040 compares the addresses of any incoming frame to these addresses and tests the status of the inverse filtering (CSR6<4>). It rejects addresses that

- Do not match if inverse filtering (CSR6<4> = 0) occurs
- Match if inverse filtering (CSR6<4> = 1) occurs

The setup frame must *always* supply all 16 addresses. Any mix of physical and multicast addresses can be used. Unused addresses should duplicate one of the valid addresses.

Figure 4–12 shows the perfect filtering setup frame buffer format of the addresses.

Figure 4–12 Perfect Filtering Setup Frame Buffer Format

	31 — 16	15 — 0
<3:0>	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 00 (Bytes <1:0>)
<7:4>	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 00 (Bytes <3:2>)
<11:8>	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 00 (Bytes <5:4>)
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 01
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 01
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 01
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 02
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 02
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 02
		Physical Address 03
	. . . . . .	
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 14
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 14
	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 14
<183:180>	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 15 (Bytes <1:0>)
<187:184>	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 15 (Bytes <3:2>)
<191:188>	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Physical Address 15 (Bytes <5:4>)

xxxxxx = don't care

MLO-010327

The low-order bit of the low-order bytes is the multicast bit of the address.

Example 4–1 shows a perfect filtering setup buffer (fragment).

Example 4–1 Perfect Filtering Buffer

```
  Ethernet addresses to be filtered:
❶  A8-09-65-12-34-76
   09-BC-87-DE-03-15
     .
     .
     .

  Setup frame buffer fragment while in little endian byte ordering:
❷  xxxx09A8
   xxxx1265
   xxxx7634
   xxxxBC09
   xxxxDE87
   xxxx1503
     .
     .
     .
  Setup frame buffer fragment while in big endian byte ordering:
❸  A809xxxx
   6512xxxx
   3476xxxx
   09BCxxxx
   87DExxxx
   0315xxxx
     .
     .
     .
```

❶ This displays two Ethernet addresses written according to the Ethernet specification for address display.

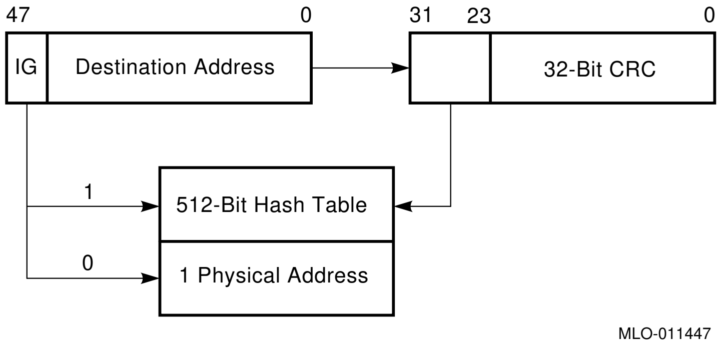
❷ This displays two addresses as they would appear in the buffer in little endian format.

❸ This displays two addresses as they would appear in the buffer in big endian format.

4.2.3.4 Imperfect Filtering Setup Frame Buffer

This section describes how the 21040 interprets a setup frame buffer in imperfect filtering mode (CSR6<0> is set). Figure 4–13 shows imperfect filtering.

Figure 4–13 Imperfect Filtering



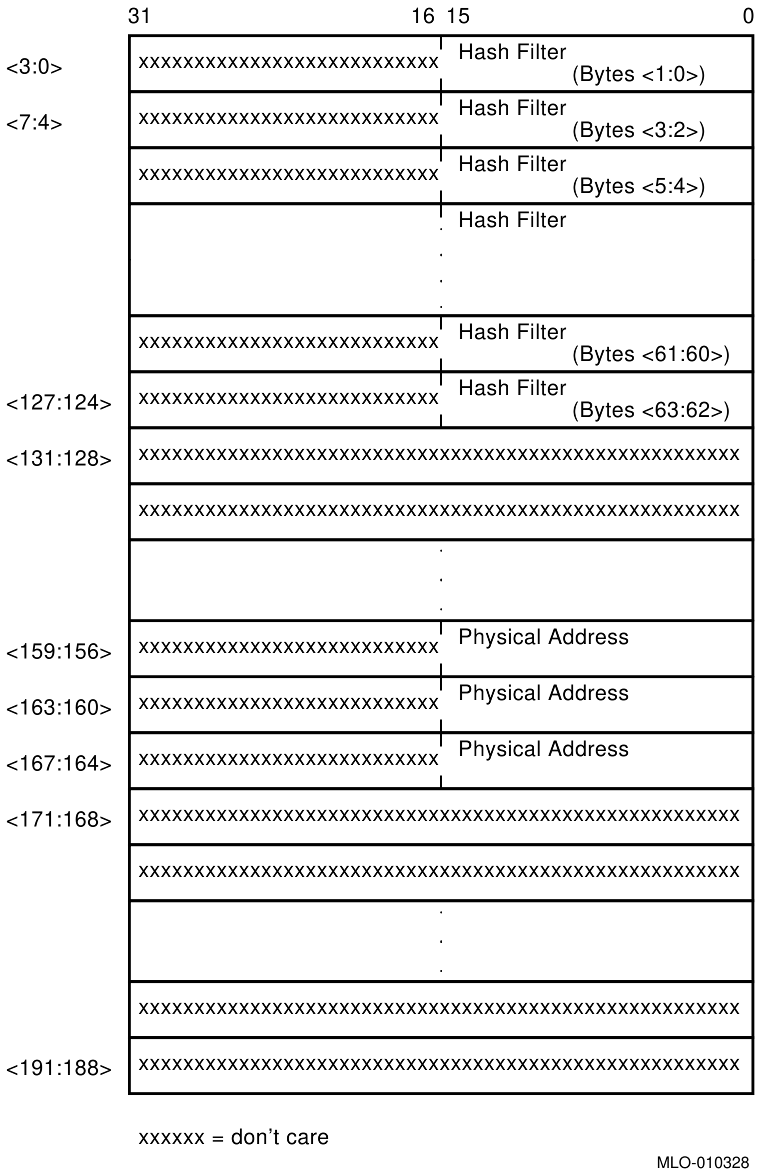
The 21040 can store 512 bits serving as hash bucket heads and one *physical* 48-bit Ethernet address. Incoming frames with multicast destination addresses are subjected to imperfect filtering. Frames with physical destination addresses are checked against the single physical address.

For any incoming frame with a multicast destination address, the 21040 applies the standard Ethernet cyclic redundancy check (CRC) function to the first 6 bytes that contain the destination address, then it uses the most significant 9 bits of the result as a bit index into the table. If the indexed bit is set, the frame is accepted. If the bit is cleared, the frame is rejected. (Appendix C provides an example of a hash index for a given Ethernet address.)

This filtering mode is called imperfect because multicast frames not addressed to this station may slip through, but it still decreases the number of frames that the host can receive.

Figure 4–14 shows the format for the hash table and the physical address.

Figure 4–14 Imperfect Filtering Setup Frame Format



Bits are sequentially numbered from right to left and down the hash table. For example, if the CRC (destination address) <8:0> = 33, the 21040 examines bit 1 in the fourth longword.

Example 4–2 shows an imperfect filtering setup frame buffer.

Example 4–2 Imperfect Filtering Buffer

```
Ethernet addresses to be filtered:
❶ 25-00-25-00-27-00
   A3-C5-62-3F-25-87
   D9-C2-C0-99-0B-82
   7D-48-4D-FD-CC-0A
   E7-C1-96-36-89-DD
   61-CC-28-55-D3-C7
   6B-46-0A-55-2D-7E

❷ A8-12-34-35-76-08


Setup frame buffer while in little endian byte ordering:

❸ xxxx0000
   xxxx0000
   xxxx0000
   xxxx1000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx4000
   xxxx0080
   xxxx0000
   xxxx0000
   xxxx0010
```

(continued on next page)

Example 4–2 (Cont.) Imperfect Filtering Buffer

```
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx1000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0001
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0040
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx

❹  xxxx12A8
   xxxx3534
   xxxx0876
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
```

(continued on next page)

Example 4–2 (Cont.) Imperfect Filtering Buffer

```
Setup frame buffer while in big endian byte ordering:
❺ 0000xxxx
  0000xxxx
  0000xxxx
  0010xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0040xxxx
  8000xxxx
  0000xxxx
  0000xxxx
  1000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0010xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0100xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  4000xxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
```

(continued on next page)

Example 4–2 (Cont.) Imperfect Filtering Buffer

```
❻ A812xxxx
  3435xxxx
  7608xxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
  xxxxxxxx
```

❶ This displays Ethernet multicast addresses written according to the Ethernet specification for address display.

❷ This displays an Ethernet physical address.

❸ This displays the first part of an imperfect filter setup frame buffer, in little endian byte ordering, with set bits for the multicast addresses as in ❶.

❹ This displays the second part of the buffer with the physical address as in ❷, in little endian byte ordering.

❺ This displays the first part of an imperfect filter setup frame buffer, in big endian byte ordering, with set bits for the multicast addresses as in ❶.

❻ This displays the second part of the buffer with the physical address as in ❷, in big endian byte ordering.

5

Functional Description

This chapter describes the reset commands, interrupt handling, and startup. It also describes the transmit and receive processes.

The functional operation of the 21040 is controlled by the driver interface located in the host communication area. The driver interface activity is controlled by command and status registers (CSRs), descriptor lists, and data buffers.

Descriptor lists and data buffers, collectively referred to as the host communication area, reside in host memory. These data structures process the actions and status related to buffer management. The 21040 transfers frame data to and from the receive and transmit buffers in host memory. Descriptors resident in the host memory act as pointers to these buffers (Chapter 4).

5.1 Reset Commands

Two commands are available to reset the 21040: hardware and software.

1. Assert **rst_l**, to initiate a hardware reset.
2. Assert **csr0 bit 0**, to initiate a software reset.

For a proper reset operation, both clocks (**clk_l**, **xtal1**) should operate normally. For both the hardware and software reset commands, the 21040 *aborts* all processing and starts the reset sequence. The 21040 initializes all internal states and registers.

Note

No internal states are retained, no descriptors are owned, and all the host-visible registers are set to the reset values. However, a reset command has no effect on the configuration registers.

The 21040 does not explicitly disown any owned descriptor; descriptor OWNed bits can be left in a state indicating 21040 ownership. Sections 4.2.1.1 and 4.2.2.1 provide detailed descriptions of OWN bits.

After either a hardware or software reset command, the first bus transaction to the 21040 should not be initiated for at least 50 PCI clock cycles. When the reset sequence completes, the 21040 can accept host commands. The receive and transmit processes are placed in the stopped state (Table 5–2 and Table 5–3). It is permissible to issue successive reset commands (hardware or software).

5.2 Arbitration Scheme

In general, the arbitration scheme used by the 21040 grants precedence to the receive process instead of the transmit process (CSR0<1>). Table 5–1 lists a description of the arbitration scheme. The technical expressions used in this table are described in the following list.

- **Txreq**—A DMA request for the transmit process to
  - Fetch descriptor, or
  - Close descriptor, or
  - Setup packet, or
  - Sufficient space allocated in the transmit FIFO for a full data burst
- **Rxreq**—A DMA request for the receive process to
  - Fetch descriptor, or
  - Close descriptor, or
  - Sufficient data entered into the receive FIFO for a full data burst, **or** the end of receive packet (Rxpacket) and the data in the receive FIFO is less than a full burst
- **Txen—**21040 is currently transmitting.

- **RxF>tr**— The amount of free bytes left in the receive FIFO. The values are taken from the programmed threshold values in CSR6<15:14>. Table 3–37 lists the coding for the programmed values which include 72, 96, 128, and 160 bytes.
- **TxF>tr**—The data in the transmit FIFO exceeds the programmed transmit FIFO threshold. Table 3–37 lists the coding for the programmed values which include 72, 96, 128, and 160 bytes.

Table 5–1 Arbitration Scheme

Txreq	Rxreq	Txen	RxF>tr	TxF>tr	Chosen Process
0	0	0	–	–	–
0	0	1	–	–	–
0	1	0	–	–	Receive process
0	1	1	–	–	Receive process
1	0	0	–	–	Transmit process
1	0	1	–	–	Transmit process
1	1	0	–	–	Receive process
1	1	1	0	0	Transmit process
1	1	1	0	1	Transmit process
1	1	1	1	0	Transmit process
1	1	1	1	1	Receive process

In addition to the arbitration scheme listed in Table 5–1, two other factors must be considered:

- The transmit process obtains a window for one burst between two consecutive receive packets.
- The receive process obtains a window for one burst between two consecutive transmit packets.

5.3 Interrupts

Interrupts can be generated as a result of various events. CSR5 contains all the status bits that may cause an interrupt. The following list contains the events that cause interrupts.

CSR5<0>—Transmit interrupt
CSR5<1>—Transmit process stopped
CSR5<2>—Transmit buffer unavailable
CSR5<3>—Transmit jabber time-out
CSR5<5>—Transmit underflow
CSR5<6>—Receive interrupt
CSR5<7>—Receive buffer unavailable
CSR5<8>—Receive process stopped
CSR5<9>—Receive watchdog time-out
CSR5<10>—AUI/TP pin
CSR5<11>—Full-duplex short frame received
CSR5<12>—Link fail
CSR5<13>—System error

Interrupt bits are cleared by writing a 1 to the bit position. This enables additional interrupts from the same source.

Interrupts are not queued and, if the interrupting event recurs *before* the driver has responded to it, no additional interrupts are generated. For example, receive interrupt (CSR5<6>) indicates that one or more received frames were delivered to host memory. The driver must scan *all* descriptors, from the last recorded position to the first one owned by the 21040.

An interrupt is generated only *once* for simultaneous, multiple interrupting events. The driver must scan CSR5 for the interrupt cause or causes. The interrupt is not generated *again*, unless a new interrupting event occurs after the driver has cleared the appropriate CSR5 bits.

For example, transmit interrupt (CSR5<0>) and receive interrupt (CSR5<6>) are set simultaneously. The host acknowledges the interrupt, and the driver begins executing by reading CSR5. Next, receive buffer unavailable (CSR5<7>) is set. The driver writes back its copy of CSR5, clearing transmit interrupt and receive interrupt. The interrupt line is deasserted for one cycle and then asserted again with receive buffer unavailable.

5.4 Startup Procedure

The following sequence of checks and commands must be performed by the driver to prepare the 21040 for operation.

1. Wait 50 PCI clock cycles for the 21040 to complete its reset sequence.
2. Update configuration registers (Section 3.1)
   - Read the configuration register to identify the 21040 and its revision.
   - Write the configuration interrupt register (if interrupt mapping is necessary).
   - Write the configuration command register.
   - Write the configuration register to map the 21040 I/O or memory address space into the appropriate processor address space.
3. Write CSR0 to set global host bus operating parameters (Section 3.2.1.1).
4. Write the Interrupt Mask Register to mask unnecessary (depending on the particular application) interrupt causes.
5. If the driver wants to change the default settings of the jabber timers and SIA initial setting, write to CSR13, CSR14, and CSR15 (Section 3.2.2).
6. The driver must create the transmit and receive descriptor lists. Then, it writes to both CSR3 and CSR4 providing the 21040 with the starting address of each list (Section 3.2.1.4). The first descriptor on the transmit list may contain a setup frame (Section 4.2.3).

Caution

If address filtering (either perfect or imperfect) is desired, the receive process should be started only after the setup frame has been processed (Section 4.2.3).

7. Write CSR6 (Section 3.2.1.6) to set global serial parameters and start both the receive and transmit processes. The receive and transmit processes enter the running state and attempt to acquire descriptors from the respective descriptor lists. Then the receive and transmit processes begin processing incoming and outgoing frames. The receive and transmit processes are independent of each other and can be started and stopped separately.

5.5 Receive Process

While in the running state, the receive process polls the receive descriptor list, attempting to acquire free descriptors. Incoming frames are processed and placed in acquired descriptors' data buffers. Status information is written to receive descriptor 0.

5.5.1 Descriptor Acquisition

The 21040 always attempts to acquire an extra descriptor in anticipation of incoming frames. Descriptor acquisition is attempted if any of the following conditions are satisfied.

- When start/stop receive (CSR6<1>) sets immediately after being placed in the running state.
- When the 21040 begins writing frame data to a data buffer pointed to by the current descriptor, and the buffer ends before the frame ends.
- When the 21040 completes the reception of a frame, and the current receive descriptor has been closed.
- When the receive process is suspended because of host-owned buffer (RDES0<31>=0), and a new frame is received.
- When receive poll demand is issued (Section 3.2.1.3).

5.5.2 Frame Processing

As incoming frames arrive, the 21040 recovers the incoming Manchester data, converts it to data and clock pulses, and then sends it to the receive engine. The receive engine strips the preamble bits and stores the frame data in the receive FIFO. Concurrently, the receive section performs address filtering depending on the results of inverse filtering (CSR6<6>), hash/perfect receive filtering mode (CSR6<0>), and hash-only receive filtering mode (CSR6<2>), as well as its internal filtering table. If the frame fails the address filtering, it is ignored and purged from the FIFO. Frames that are shorter than 64 bytes, because of collision or premature termination, are also ignored and purged from the FIFO (unless pass bad frames bit CSR6<3> is set).

After 64 bytes have been received, the 21040 begins transferring the frame data to the buffer pointed to by the current descriptor. The 21040 sets first descriptor (RDES0<9>) to delimit the frame. Then, the descriptors are released when the OWN (RDES0<31>) bit is reset to 0 either as the data buffers fill up or as the last segment of a frame is transferred to a buffer. If a frame is contained in a single descriptor, both last descriptor (RDES0<8>) and first descriptor (RDES0<9>) are set.

The 21040 sets last descriptor (RDES0<8>) and fetches the next descriptor, and the RDES0 status bits in the last frame descriptor is released. After the last frame descriptor is released, the 21040 sets receive interrupt (CSR5<6>) and fetches the next descriptor. The same process repeats unless the 21040 encounters a descriptor flagged as being owned by the host. If this occurs, the receive process sets receive buffer unavailable (CSR5<7>) and then enters the suspended state. The position in the receive list is retained.

5.5.3 Receive Process Suspended

If any frames enter while the receive process is suspended, the 21040 fetches the current descriptor in host memory. If the descriptor is now owned by the 21040, the receive process reenters the running state and starts the frame reception.

If a receive frame arrives while the receive process is suspended, the 21040 refetches the next descriptor. If the descriptor is still owned by the host, the 21040 increments the missed frames counter (CSR8<15:0>) and discards the current frame in the receive FIFO. If more than one frame is stored in the receive FIFO, the process repeats.

5.5.4 Receive Process State Transitions

Table 5–2 lists the receive process state transitions and the resulting actions.

Table 5–2 Receive Process State Transitions

From State	Event	To State	Action
Stopped	Start receive command.	Running	Receive polling begins from last list position or from the list head, if this is the first start receive command issued or if the receive descriptor list address (CSR3) was modified by the driver.
Running	21040 attempts to acquire a descriptor owned by the host.	Suspended	Receive buffer unavailable (CSR5<7>) sets when the last acquired descriptor buffer is consumed. The position in the list is retained.

(continued on next page)

Table 5–2 (Cont.) Receive Process State Transitions

From State	Event	To State	Action
Running	Stop receive command.	Stopped	Receive process is stopped after the current frame, if any, is completely transferred to data buffers. Receive process stopped (CSR5<8>) sets. The position in the list is retained.
Running	Memory or host bus parity error encountered.	Running	21040 operation is stopped and system error (CSR5<13>) sets. The 21040 remains in the running state. A software reset must be issued to release the 21040.
Running	Reset command.	Stopped	Receive capability is cut off.
Suspended	Receive poll demand or incoming frame and available descriptor.	Running	Receive polling resumes from last list position.
Suspended	Stop receive command.	Stopped	Receive process stopped (CSR5<8>) sets.
Suspended	Reset command.	Stopped	None.

5.6 Transmit Process

While in the running state, the transmit process polls the transmit descriptor list for frames requiring transmission. After polling starts, it continues in either sequential descriptor ring order or chained order. When it completes frame transmission, status information is written into transmit descriptor 0 (TDES0). If the 21040 detects a descriptor flagged as owned by the host, or if an error condition occurs, the transmit process is suspended and both transmit buffer unavailable (CSR5<2>) and normal interrupt summary (CSR5<16>) are set.

Transmit interrupt (CSR5<0>) is set after completing transmission of a frame that has interrupt on completion (TDES1<31>) set in its last descriptor. When this occurs, the transmission process continues to run.

While in the running state, the transmit process can simultaneously acquire two frames. As the transmit process completes copying the first frame, it immediately polls the transmit descriptor list for the second frame. If the second frame is valid, the transmit process copies the frame before writing the status information of the first frame.

5.6.1 Frame Processing

Frames can be data-chained and span several buffers. Frames must be delimited by the first descriptor (TDES1<29>) and the last descriptor (TDES1<30>) respectively.

As the transmit process starts execution, the first descriptor must have TDES1<29> set. When this occurs, frame data transfers from the host buffer to the internal FIFO. Concurrently, if the current frame has the last descriptor TDES1<30> clear, the transmit process attempts to acquire the next descriptor. The transmit process expects this descriptor to have TDES1<29> clear. If TDES1<30> is clear, it indicates an intermediary buffer. If TDES1<30> is set, it indicates the last buffer of the frame.

After the last buffer of the frame has been transmitted, the 21040 writes back the final status information to the transmit descriptor 0 (TDES0) word of the descriptor that has the last segment set in transmit descriptor 1 (TDES1<30>). At this time, if interrupt on completion (TDES1<31>) was set, the transmit interrupt (CSR5<0>) is set, the next descriptor is fetched, and the process repeats.

Actual frame transmission begins after the internal FIFO had reached either a programmable threshold CSR6<15:14> (Table 3–37) or after a full frame is contained in the FIFO. Descriptors are released (OWN bit TDES0<31> clears) when the 21040 completes the packet transmission.

5.6.2 Transmit Polling Suspended

Transmit polling can be suspended by either of the following conditions:

- The 21040 detects a descriptor owned by the host (TDES0<31>=0). To resume, the driver must give descriptor ownership to the 21040 and then, issue a poll demand command.
- A frame transmission is aborted when a locally induced error is detected. The appropriate transmit descriptor 0 (TDES0) bit is set.

If either of the previous two conditions occur, both abnormal interrupt summary (CSR5<15>) and transmit interrupt (CSR5<0>) are set, and the information is written to transmit descriptor 0 causing the suspension.

In both of the cases just described, the position in the transmit list is retained. The retained position is that of the *descriptor following the last descriptor closed* (set to host ownership) by the 21040.

Note

The 21040 *does not* automatically poll the transmit descriptor list. The driver *must* explicitly issue a transmit poll demand command after rectifying the suspension cause, unless the transmit automatic polling (CSR0<18:17>) field is set to 1.

5.6.3 Transmit Process State Transitions

Table 5–3 lists the transmit process state transitions and the resulting actions.

Table 5–3 Transmit Process State Transitions

From State	Event	To State	Action
Stopped	Start transmit command.	Running	Transmit polling begins from one of the following positions: The last list position. The head of the list, if this is the first start command issued after CSR4 was initialized or modified.
Running	21040 attempts acquisition of a descriptor owned by the host.	Suspended	Transmit buffer unavailable (CSR5<2>) sets.
Running	Frame transmission aborts because a locally induced underflow error (TDES0<1>) is detected (Section 4.2.2.1).	Suspended	The following bits set: TDES0<1>—Underflow error CSR5<5> —Transmit underflow CSR5<15>—Abnormal interrupt summary
Running	Stop transmit command.	Stopped	Transmit process is stopped after the current frame, if any, is transmitted.

(continued on next page)

Table 5–3 (Cont.) Transmit Process State Transitions

From State	Event	To State	Action
Running	Frame transmission aborts because a transmit jabber time-out (TDES0<14>) was detected (Section 4.2.2.1).	Stopped	The following bits are set: TDES0<14>—Transmit jabber time-out CSR5<1>—Transmit process stopped CSR5<3>—Transmit jabber time-out CSR5<15>—Abnormal interrupt summary
Running	Parity error detected by memory or host bus.	Running	Transmission is cut off and system error (CSR5<13>) sets. The 21040 remains in the running state. If a software reset occurs, normal operation continues.
Running	Reset command.	Stopped	Transmission is cut off. If CSR4 was not changed, the position in the list is retained. If CSR4 was changed, the next descriptor address is fetched from the header list (CSR4) when the poll demand command is issued. Transmit process stopped (CSR5<1>) sets.
Suspended	Transmit poll demand command issued.	Running	Transmit polling resumes from the last list position.
Suspended	Stop transmit command.	Stopped	Transmit process stopped (CSR5<1>) sets.
Suspended	Reset command.	Stopped	None.

5.7 Loopback Operations

The 21040 supports two loopback modes: internal loopback and external loopback. The loopback data rate is 10 megabits per second.

5.7.1 Internal Loopback Mode

Internal loopback mode is normally used to verify that the internal logic operations function correctly. In loopback mode, the 21040 takes frames from the transmit list and loops them back internally to the receive list. In loopback mode, the 21040 disengages from the Ethernet wire. Internal loopback mode also supports the following modes of operation (Table 3–59).

- Media access control (MAC) internal loopback mode in which transmit packets are looped back at the MAC level and the 21040 disengages the SIA. The loopback data rate is 10 megabits per second.
- 10BASE-T internal loopback mode in which transmit packets from the encoder output are selected and looped back to the decoder input. The loopback data rate is 10 megabits per second.

5.7.2 External Loopback Mode

External loopback mode is normally used to verify that the logic operations up to the Ethernet cable function correctly. In external loopback mode, the 21040 takes frames from the transmit list and transmits them on the Ethernet wire. Concurrently, the 21040 listens to the line that carries its own transmissions and places incoming frames in the receive list.

Caution

In external loopback mode, when transmitted frames are placed on the Ethernet wire, the 21040 does not check the origin of any incoming frames. It is possible for frames not originating from the 21040 to enter the receive buffers.

External loopback mode also supports the following modes of operation (Table 3–59).

- 10BASE-T external loopback mode transmits packets using twisted-pair wires. Concurrently, the 21040 disables the internal collision detector and thus listens to the line that carries its own transmission.

  The board designer must use an external shunt to connect the transmit line with the receive line.
- AUI external loopback mode transmits packets using the AUI cable up-to-MAU (medium attachment unit) to check the MAU integrity.
- SIA external loopback mode transmits packets using the external SIA up-to-MAU to check the MAU integrity.

5.7.3 Driver Entering Loopback Mode

To enter a specific loopback mode, the driver must take the following actions:

Note

All address filtering and validity checking rules apply while in either MAC or 10BASE-T mode.

---

1. Stop the receive and transmit processes by writing 0 to both the start/stop receive (CSR6<1>) and the start/stop transmit (CSR6<13>) fields. The driver must wait for any previously scheduled frame activity to cease by polling the transmit process state (<22:20>) and the receive process state (<19:17>) fields in CSR5.
2. Prepare the appropriate transmit and receive descriptor lists in host memory. These lists can either follow the existing lists at the point of suspension or be new lists that have to be identified to the 21040 as indicated by the receive list base address in CSR3 and by the transmit list address in CSR4.
3. Stop the SIA by setting CSR13 to a value of 00000000H.
4. Set the following bits to the desired loopback mode (Table 3–53).

   CSR6<11:10>—Operating mode
   CSR14<1>—Loopback enable
   CSR14<8>—Receive squelch enable
   CSR14<10>—Collision detect enable

5. Set the following bits to the desired SIA operation mode (Table 3–53).

   CSR13<3>—10BASE-T or AUI
   CSR13<2>—CSR auto configuration
   CSR13<1>—AUI/TP pin selection
   CSR13<0>—SIA reset

6. Use start commands to place both the transmit and receive processes into the running state.
7. As in normal processing, execute any 21040 interrupts.

5.7.4 Driver Restoring Normal Operation

To restore normal operation, the driver must execute the following procedure:

1. Stop both the receive and transmit processes. The driver must wait for any previously scheduled frame activity to cease by polling both the transmit (CSR5<22:20>) and receive process state (CSR5<19:17>) fields in CSR5.
2. Prepare appropriate transmit and receive descriptor lists in host memory. These lists can either follow the existing lists at the point of suspension or be new lists that have to be identified to the 21040 by the receive list base address in CSR3 and the transmit list base address in CSR4.
3. Stop the SIA by setting CSR13 to a value of 00000000H.
4. Set the following bits as indicated (Table 3–53):

   CSR6<11:10>—Operating mode = 00
   CSR14<1>—Loopback enable = 0
   CSR14<8>—Receive squelch enable = 1
   CSR14<10>—Collision detect enable = 1

5. Set the following bits to the desired SIA operation mode (Table 3–53).

   CSR13<3>—10BASE-T or AUI
   CSR13<2>—CSR auto configuration
   CSR13<1>—AUI/TP pin selection
   CSR13<0>—SIA reset

6. Use start commands to place both the transmit and receive processes into the running state.
7. Resume normal processing. Execute any 21040 interrupts.

5.8 Full-Duplex Operation

The 21040 supports full-duplex operation only in 10BASE-T mode. The 21040 activates the transmit and receive processes simultaneously. It also supports receive back-to-back packets with an interpacket gap (IPG) of 9.6 microseconds in parallel with transmit back-to-back packets with an IPG of 9.6 microseconds.

The 21040 implements a programmable full-duplex operating mode (CSR6<9>) bit that commands the MAC to ignore both the carrier and the collision detect signal.

The driver must take the following actions to enter full-duplex operation.

1. Stop the receive and transmit processes by writing 0 to CSR6<1> and CSR6<13> fields respectively. The driver must wait for any previously scheduled frame activity to cease by polling the transmit process state (<22:20>) and receive process state (<19:17>) fields in CSR5.
2. Prepare appropriate transmit and receive descriptor lists in host memory. These lists can use the existing lists at the point of suspension or create new lists that must be identified to the 21040 by referencing the receive list base address in CSR3 and the transmit list base address in CSR4.
3. Set full-duplex mode (CSR6<9>) and operating mode (CSR6<11:10>).
4. Using Table 3–53 as a guide, set CSR13 through CSR15.
5. Place the transmit and receive processes in the running state by using the start commands.
6. Resume normal processing. Execute any 21040 interrupts.

6 Host Bus Operation

This chapter describes the bus slave and bus master read and write cycles. It also describes the initiation of termination cycles by the bus master and bus slave. Master termination cycles can be initiated by the 21040 and the memory controller.

6.1 Overview

The peripheral component interconnect (PCI) is the physical interconnection used between highly integrated peripheral controller components and the host system. The 21040 uses the PCI bus to communicate with the host CPU and memory.

The 21040 is directly compatible with the *PCI Local Bus Specification, Revision 2.0*. The 21040 supports a subset of the PCI-bus cycles (transactions). When communicating with the host, the 21040 operates as a bus slave; when communicating with the memory, it acts as a bus master.

All signals are sampled on the rising edge of the clock. Each signal has a setup and hold aperture with respect to the rising clock edge. Refer to the *DECchip 21040 Ethernet LAN Controller for PCI Data Sheet* for detailed timing information. Table 2–2 lists the codes for bus commands.

Note

The term **clock cycle**, as used in this chapter, refers to the PCI bus clock period specification.

6.2 Bus Slave Operation

All host accesses to CSRs and configuration registers in the 21040 are executed with the 21040 acting as the slave. The bus slave operations include the following:

- I/O read
- I/O write
- Configuration read
- Configuration write
- Memory read
- Memory write
- Memory read/write (includes memory write and invalidate, memory read line, and memory read multiple)

Note

- The 21040 does not support the following bus transactions:

  Interrupt acknowledge
  Special cycle
  Dual-address cycle

- If the 21040 is targeted for a burst I/O or memory operation, it responds with a retry on the second data transaction.

6.2.1 Slave Read Cycle (I/O or Memory Target)

Figure 6–1 shows a typical slave read cycle. The 21040 I/O read cycle is executed as follows:

1. The host initiates the slave read cycle by asserting the **frame_l** signal, driving the address on the **ad** lines and driving the bus command (slave read operation) on the **c_be_l** lines.
2. The 21040 samples the address and the bus command on the next clock edge.
3. The host deasserts **frame_l** signal and asserts **irdy_l** signal.
4. The 21040 asserts **devsel_l** and, at the next cycle, drives the data on the **ad** lines.

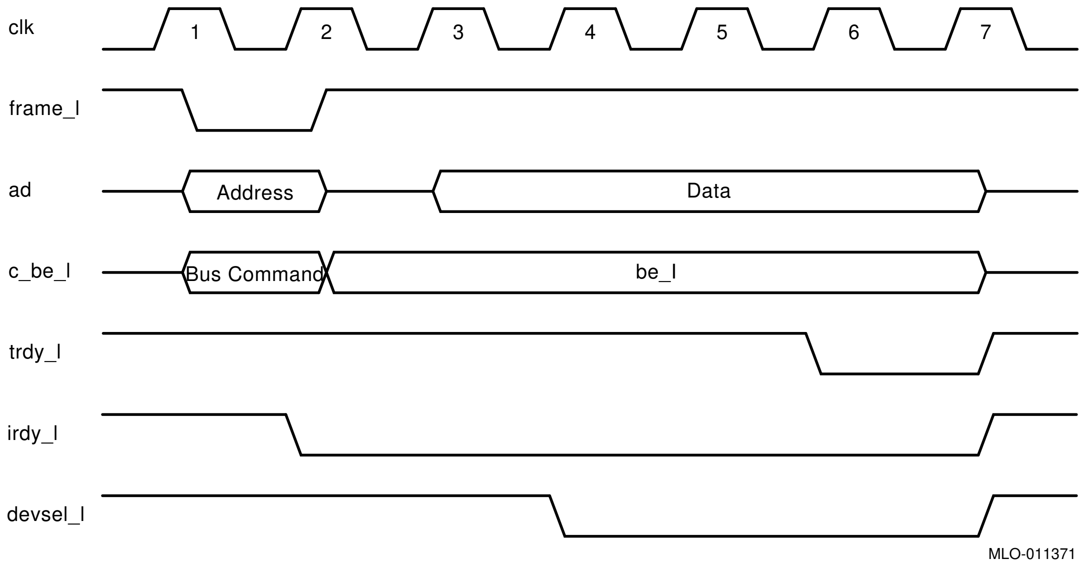
5. The read transaction completes when both **irdy_l** and **trdy_l** are asserted by the host and the 21040, respectively, on the same clock edge.

   The 21040 assumes that **c_be_l** lines are 0000 (longword access).

   If the **c_be_l** lines are 1111, the **ad** bus read is 00000000H with correct parity.

6. The host and the 21040 terminates the cycle by deasserting **irdy_l** and **trdy_l**, respectively.

Figure 6–1 Slave Read Cycle



6.2.2 Slave Write Cycle (I/O or Memory Target)

Figure 6–2 shows a typical slave write cycle. The 21040 slave write cycle is executed as follows:

1. The host initiates the slave write cycle by asserting the **frame_l** signal and driving the address on the **ad** lines and the bus command (slave write operation) on the **c_be_l** lines.

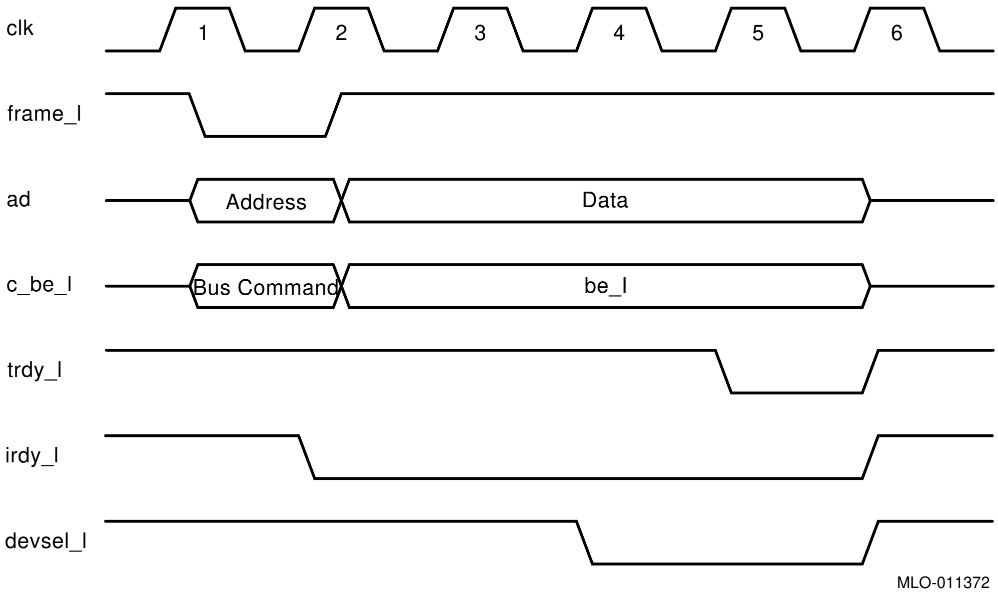
2. The 21040 samples the address and the bus command on the next clock edge.

3. The host deasserts **frame_l** and drives the data on the **ad** lines along with **irdy_l**.
4. The 21040 samples the data and asserts **devsel_l** and **trdy_l**.
5. The host and the 21040 complete the write transaction by asserting **irdy_l** and **trdy_l**, respectively, on the same clock edge.

   The 21040 assumes that **c_be_l** lines are 0000 (longword access).

   If the **c_be_l** lines are 1111, the write transaction completes normally on the bus, but the write to the CSR is not executed.
6. The host and the 21040 terminate the cycle by deasserting **irdy_l** and **trdy_l**, respectively.

Figure 6–2 Slave Write Cycle



6.2.3 Configuration Read and Write Cycles

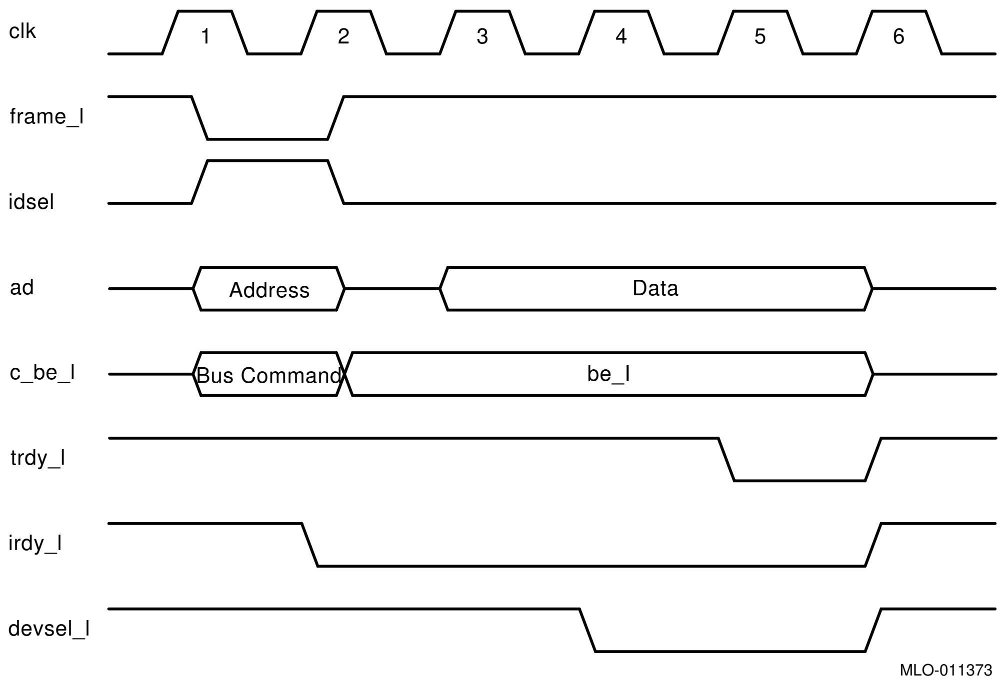
The 21040 provides a way for software to analyze and configure the system before defining any address assignments or mapping. The 21040 provides 256 bytes of configuration registers. Section 3.1 describes these registers.

Note

c_be_l lines are supported in the configuration space accesses.

Figure 6–3 shows a configuration read cycle. The hosts selects the 21040 by asserting **idsel**. The 21040 responds by asserting **devsel_l**. The remainder of the read cycle is similar to the slave read cycle (Section 6.2.1).

Figure 6–3 Configuration Read Cycle



6.3 Bus Master Operation

All memory accesses are completed with the 21040 as the master on the PCI bus. The bus master operations include the following:

- Bus arbitration
- Memory read cycle

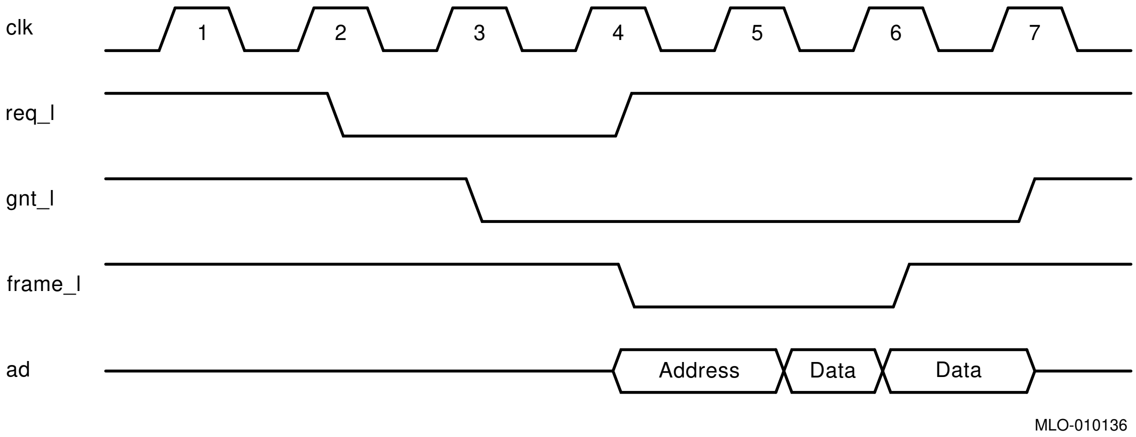
- Memory write cycle
- Terminations

6.3.1 Bus Arbitration

The 21040 uses the PCI central arbitration mechanism with its unique request (**req_l**) and grant (**gnt_l**) signals. Figure 6–4 shows the bus arbitration mechanism. The 21040 bus arbitration is executed as follows:

1. The 21040 requests the bus by asserting **req_l**.
2. The arbiter, in response, asserts **gnt_l** (**gnt_l** can be deasserted on any clock).
3. The 21040 ensures that its **gnt_l** is asserted on the clock edge on which it wants to drive **frame_l**. (If **gnt_l** is deasserted, the 21040 does not proceed).
4. The 21040 deasserts **req_l** on the cycle that it asserts **frame_l**.

Figure 6–4 Bus Arbitration



The 21040 uses **gnt_l** according to the following rules:

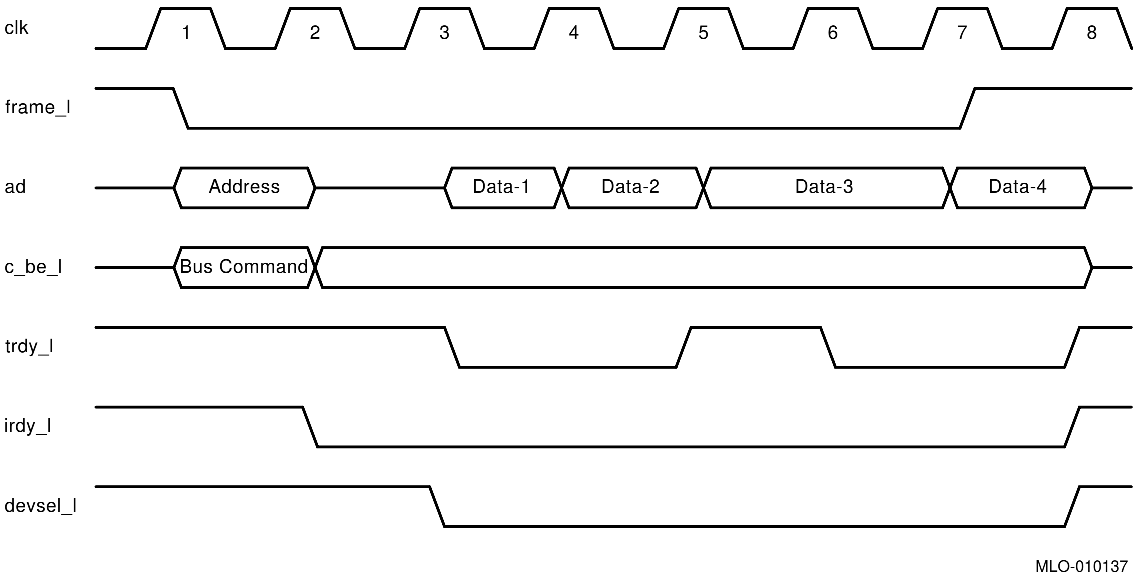
- If **gnt_l** is deasserted together with the assertion of **frame_l**, the 21040 continues its bus transaction.
- If **gnt_l** is asserted while **frame_l** remains deasserted, the arbiter can deassert **gnt_l** at any time. The 21040 must assert **req_l**, then **gnt_l** again, before asserting **frame_l**.

6.3.2 Memory Read Cycle

Figure 6–5 shows the memory read cycle. The memory read cycle is executed as follows:

1. The 21040 initiates the memory read cycle by asserting **frame_l**. It also drives the address on the **ad** lines and the appropriate bus command (read operation) on the **c_be_l** lines.
2. The memory controller samples the address and the bus command on the next clock edge.
3. The 21040 asserts **irdy_l** until the end of the read transaction.
4. During the data transfer cycles, **c_be_l** indicates which byte lines are involved in each cycle. The 21040 drives 0000 on the **c_be_l** lines (longword access).
5. The memory controller drives the data on the **ad** lines and asserts **trdy_l**.
6. The 21040 samples the data on each rising clock edge when both **irdy_l** and **trdy_l** are asserted.
7. The previous two steps may be repeated a number of times.
8. The cycle is terminated when **frame_l** is deasserted by the 21040.
9. **irdy_l** is deasserted by the 21040 and **trdy_l** is deasserted by memory controller.

Figure 6–5 Memory Read Cycle



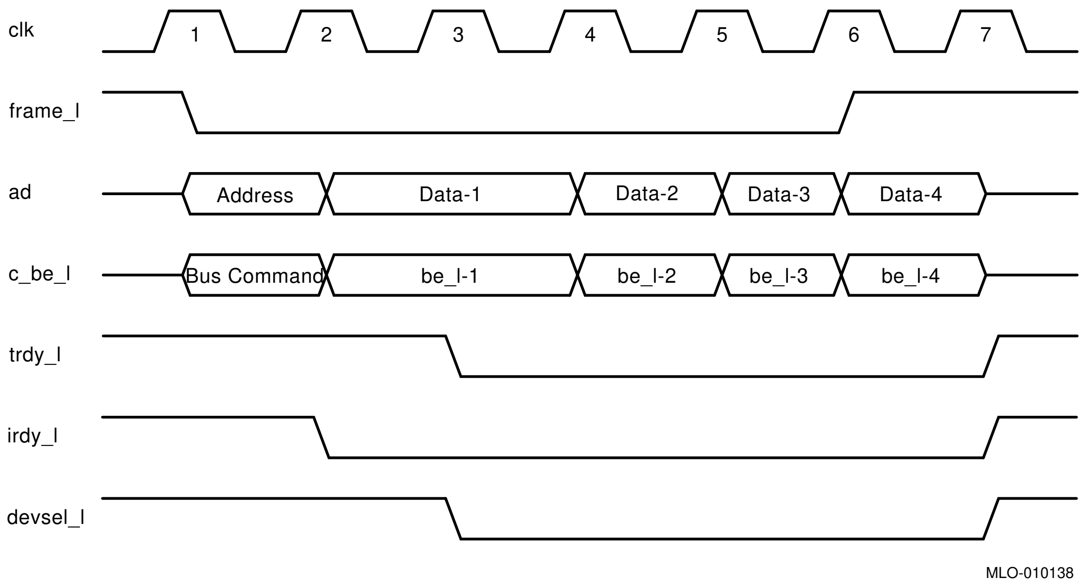
6.3.3 Memory Write Cycle

Figure 6–6 shows the memory write cycle. The memory write cycle is executed as follows:

1. The 21040 initiates the memory write cycle by asserting **frame_l**. It also drives both the address on the **ad** lines and the write operation bus command on the **c_be_l** lines.
2. The 21040 asserts **irdy_l** until the end of the transaction and drives the data on the **ad** lines.
3. The memory controller samples the address and the bus command on the next clock edge and asserts **devsel_l**.
4. During the data transfer, cycles **c_be_l** indicate which byte lanes are involved in each cycle. The 21040 drives 0000 on the **c_be_l** lines (longword access).
5. The memory controller samples the data and asserts **trdy_l**. Each data cycle is completed on the rising clock edge when both **irdy_l** and **trdy_l** are asserted.
6. The previous two steps may be repeated a number of times.

7. The 21040 terminates the cycle by deasserting **frame_l**.
8. The 21040 deasserts **irdy_l** and the memory controller deasserts **trdy_l**.

Figure 6–6 Memory Write Cycle



6.4 Termination Cycles

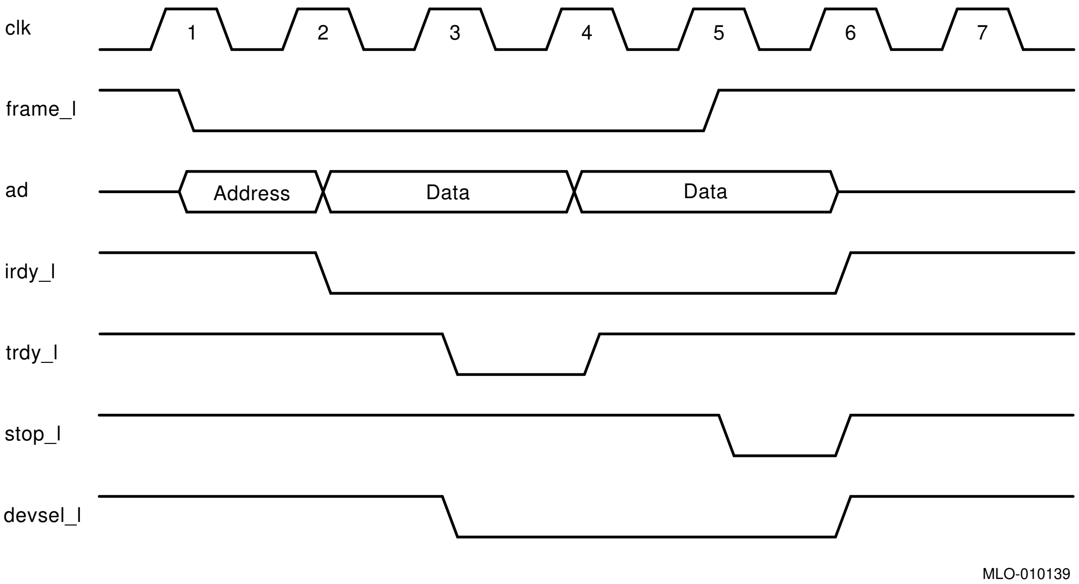
Termination cycles can be initiated during either slave or master cycles.

6.4.1 Slave-Initiated Termination

The 21040 initiates termination in slave mode when it is accessed by the host with I/O or memory burst cycles. The 21040 asserts **stop_l** to request that the host terminate the transaction. After **stop_l** is asserted, it remains asserted until **frame_l** is deasserted.

Figure 6–7 shows the retried device (the host) releasing the bus. The host retries the last data transaction after acquiring the bus in a different arbitration.

Figure 6–7 21040-Initiated Retry Cycle



6.4.2 Master-Initiated Termination

A master-initiated termination can occur when the 21040 operates as a master device on the PCI bus. Terminations can be issued by either the 21040 or the memory controller.

21040 terminations include the following:

Normal completion
Time out
Master abort

Memory-controller terminations (target) include the following:

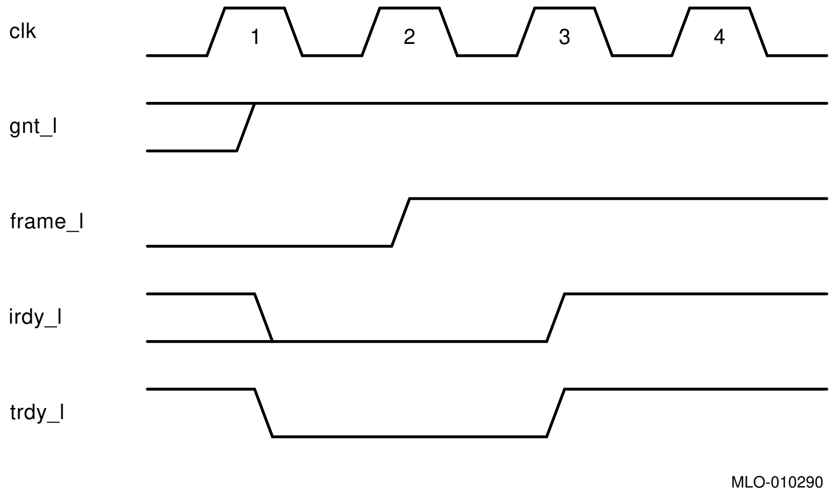
Target abort
Target-initiated
Target retry

6.4.2.1 21040-Initiated Termination

A 21040-initiated termination occurs when **frame_l** is deasserted and **irdy_l** is asserted. This indicates to the memory controller that the final data phase is in progress. The final data transfer occurs when both **irdy_l** and **trdy_l** assert. The transaction completes when both **frame_l** and **irdy_l** deassert. This is an idle bus condition.

6.4.2.1.1 Normal Completion Figure 6–8 shows a normal completion cycle termination. This indicates that the 21040 successfully completed its intended transaction.

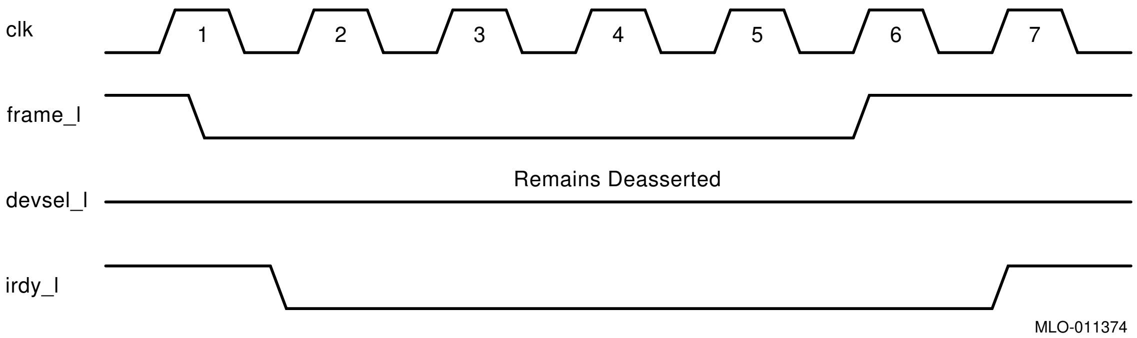
Figure 6–8 Normal Completion Cycle



6.4.2.1.2 Time Out A time-out cycle termination occurs when the **gnt_l** line has been deasserted by the arbiter and the 21040 internal latency timer has expired. However, the intended transaction has not completed. A maximum of two additional data phases are permitted, then the 21040 performs a normal transaction completion.

6.4.2.1.3 Master Abort If the target does not assert **devsel_l** within five cycles from the assertion of **frame_l**, the 21040 performs a normal completion. It then aborts the cycle by asserting both master abort (CFCS<29>) and abnormal interrupt summary (CSR5<15>). Figure 6–9 shows the 21040 master abort termination.

Figure 6–9 Master Abort Cycle

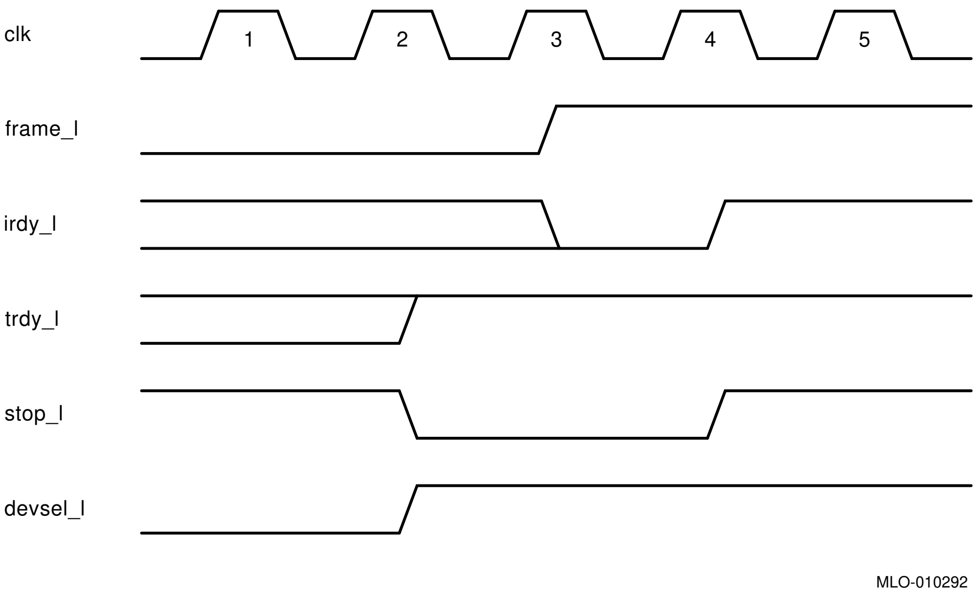


6.4.2.2 Memory-Controller-Initiated Termination

The memory controller or target can initiate certain terminations when the 21040 is the bus master.

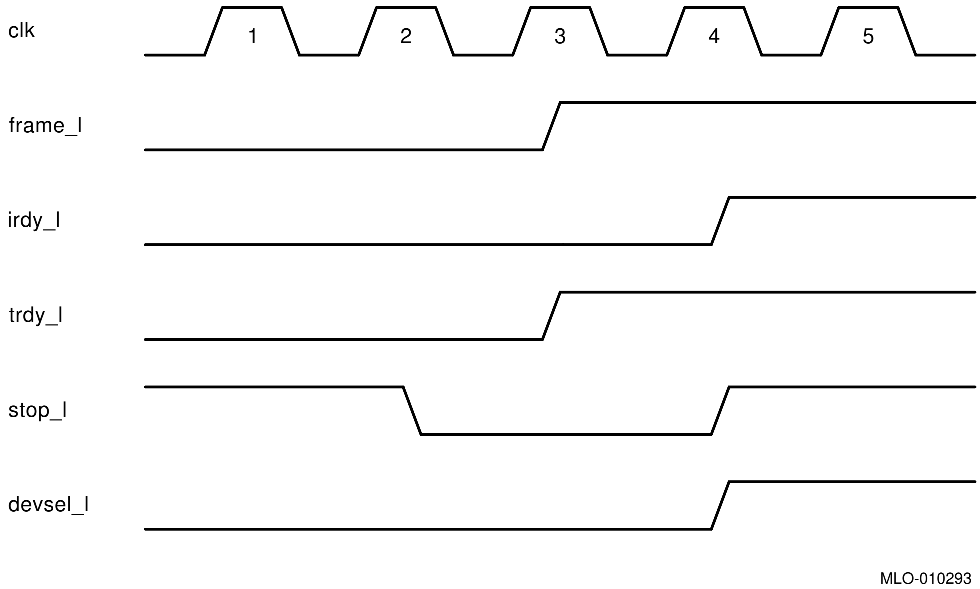
6.4.2.2.1 Target Abort The 21040 aborts the bus transaction when the target asserts **stop_l** and deasserts **devsel_l**. This indicates that the target wants the transaction to be aborted. The 21040 releases the bus and aborts the operation by asserting both received target abort (CFCS<28>) and abnormal interrupt summary (CSR5<15>). Figure 6–10 shows the 21040 abort cycle.

Figure 6–10 Abort Cycle



6.4.2.2.2 Target-Initiated Termination The 21040 terminates the bus transaction when the target asserts **stop_l**, which remains asserted until **frame_l** is deasserted. The 21040 releases the bus. Then, it retries at least the last data transaction after regaining the bus in another arbitration. Figure 6–11 shows the 21040 termination cycle.

Figure 6–11 Termination Cycle



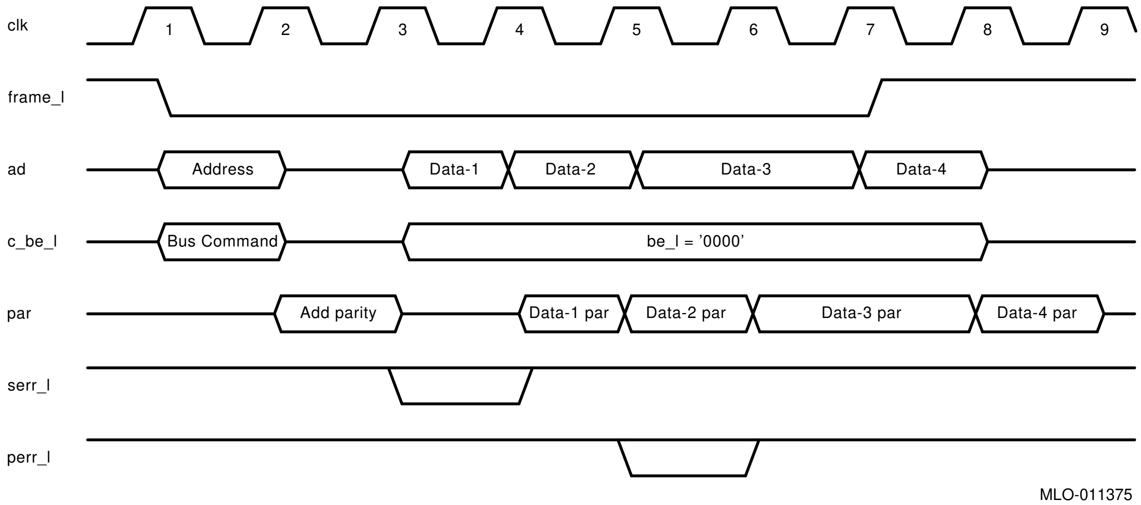
6.4.2.2.3 Target Retry The 21040 retries the bus transaction when the target asserts **stop_l** and deasserts **trdy_l**; **stop_l** remains asserted until **frame_l** is deasserted. The 21040 releases the bus. Then, it retries at least the last two data transactions after regaining the bus in another arbitration.

6.5 Parity

The 21040 supports parity generation on all address, data, and command bits. Parity is always checked and generated on the 32-bit address and data bus (**ad**) as well as on the four command (**c_be_l**) lines. The 21040 always transfers stable values (1 or 0) on all the **ad** and **c_be_l** lines. If a data parity is detected or **perr_l** is asserted when the 21040 is a bus master, the 21040 asserts detected parity error (CFCS<31>) and abnormal interrupt summary (CSR5<15>).

Figure 6–12 shows an example of parity generation on a memory write burst transaction. Note that valid parity is generated one cycle after the address and data segments were generated on the bus. One cycle after the assertion of the address parity, **serr_l** is asserted for one cycle because of an address parity error during slave operation. One cycle after the assertion of the data parity, **perr_l** is asserted because of a parity data error in either slave write or master read operations.

Figure 6–12 Parity Operation Cycle



6.6 Parking

Parking in the PCI bus allows the central arbiter to pause any selected agent. The 21040 enters the parking state when the arbiter asserts its **gnt_l** line while the bus is idle.

7

Network Interface

This chapter describes the 10BASE-T or twisted-pair (TP) interface and the attachment unit interface (AUI).

Note

All frequency and timing information in this chapter is for 10 megabits per second serial operation.

7.1 10BASE-T and AUI Functions

The 10BASE-T and AUI functions include the following:

- Supports data driver and is receiver compatible with 10BASE-T specifications
- Supports data driver and is receiver compatible with AUI specifications
- Provides AUI collision receiver compatible with AUI specifications
- Selects either AUI or 10BASE-T interfaces
- Implements Manchester decoder for incoming data
- Implements Manchester encoder for outgoing data
- Contains on-chip 20-megahertz crystal oscillator circuitry
- Enables watchdog timers on incoming and outgoing data
- Contains 10BASE-T enhanced features that include:

- Smart squelch, rejecting noise detected by the 10BASE-T receiver interface
- Combined auto polarity and LinkTest detection, presenting a robust algorithm for detection of both wire failure and switching of wires. Polarity correction is automatically done, while wire failure is reported to higher layers.

7.1.1 Receivers and Drivers

The host selects one set of data receivers and drivers at a time: either AUI or twisted-pair (TP). The other receiver and driver sets are disabled.

7.1.2 Manchester Decoder

The Manchester decoder is a phase-lock loop-based decoder that provides received clocks and data to the media access control (MAC) interface (Section 7.2).

7.1.3 Manchester Encoder

The Manchester encoder receives clocked data from the transmit engine and uses the 20-megahertz clock to provide Manchester encoded data. The encoder provides the transition to idle for the AUI and TP drivers.

7.1.4 Oscillator Circuitry

The 21040 supports two options for generating the internal 10-megahertz clock required by the internal circuitry.

1. An external parallel resonant crystal connected between **xtal1** and **xtal2** to drive the 21040-integrated oscillator circuitry.
2. An external clock generator module connected to **xtal1**; **xtal2** remains unconnected.

In both cases, the 21040 must be provided with a 20-megahertz signal that is internally divided by 2 to generate the 10-megahertz clock.

When driving the oscillator from an external clock source, an external clock having the following characteristics must be used to ensure proper operation of the 21040:

- Clock Frequency: 20-megahertz ±0.01% (TTL or CMOS)
- Rise/Fall Time: < 4 nanoseconds
- Duty Cycle: 40%–60%

Table 7–1 lists the specifications for the crystal oscillator.

Table 7–1 Crystal Oscillator Specification

Category	Value
Frequency	20 megahertz
Tolerance	±0.01% at 25°C
Stability	±0.005% at 0°C to 70°C

7.1.5 Jabber and Watchdog Timers

The jabber timer monitors the time of each packet transmission. The watchdog timer monitors the time of each packet reception. If a single packet transmission or reception exceeds a programmable value (Section 3.2.2.7), the jabber and watchdog circuitry automatically disables both the transmit and receive path. The transmit jabber timer provides the jabber function for 10BASE-T mode by cutting off transmission and asserting the collision signal to the MAC.

The packet descriptor closes with both transmit jabber time out (TDES0<14>) and late collision (TDES0<9>) setting if the jabber timer expires on a transmit packet.

7.1.6 Smart Squelch

The 21040 implements an intelligent squelch on its TP receiver to ensure that impulse noise detected on the receive inputs is not mistaken for valid signals. The squelch circuitry employs a combination of both amplitude and timing measurements to determine the validity of data received on the TP inputs.

The squelch circuit allows only valid differential receive data to pass through to the Manchester decoder provided that the following two conditions are satisfied:

1. The input amplitude is greater than the minimum signal threshold level.
2. A specific pulse sequence is received.

Satisfying these two conditions ensures that a good signal-to-noise ratio is maintained while the signal pair is active and prevents system noise from causing false squelch deactivation.

The line squelch quickly activates and deactivates within the specified time intervals, when the input squelch threshold is exceeded and a specific pulse sequence of proper polarity is detected.

The squelch circuitry rejects system noise by ignoring received pulses that are less than the required fixed time width. It also rejects pulses that are greater than the expected signal duration.

7.1.7 Auto Polarity Detector

The auto polarity detector (CSR14<13>) provides a method of detecting wire polarity by switching the polarity of the data going into the MAC layer accordingly. To detect polarity, the 21040 uses the link test pulse and the end-of-frame delimiter in an algorithm integrated into the link integrity test, as specified in the IEEE 802.3 10BASE-T supplement.

7.2 Media Access Control Operation

The LAN controller functions in a send and receive half- or full-duplex mode system. The 21040 functions in either transmit or receive in half-duplex mode or in both receive and transmit in loopback or full-duplex mode.

Before transmission, the 21040 checks that there is no competition for the network bus. In addition to listening for a clear line before transmitting, the 21040 handles collisions in a predetermined way. If two nodes attempt to transmit at the same time, the signals collide and the data on the line is garbled. The 21040 listens while it is transmitting, and it can detect a collision. If a collision is detected, the 21040 continues to transmit for a predetermined length of time, thus jamming the network, ensuring that all nodes have recognized the collision. The 21040 then delays its retransmission for a random time period determined by the truncated-binary backoff algorithm implemented in the 21040. This minimizes the possibility of a collision on retransmission.

7.2.1 Frame Format

The 21040 transmits or receives information in frames. It recognizes and transmits Ethernet and IEEE 802.3 frames.

7.2.1.1 Ethernet and IEEE 802.3 Frames

Ethernet is the generic name for the network type. An Ethernet frame has a minimum length of 64 bytes and a maximum length of 1518 bytes, exclusive of the preamble.

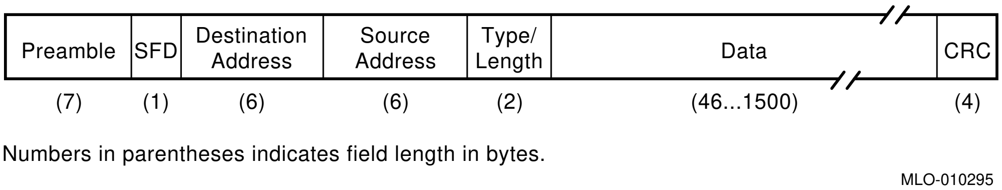
An Ethernet frame format consists of the following:

Preamble
Start frame delimiter (SFD)
Two address fields
Type or length field
Data field
Frame check sequence (FCS)

7.2.1.2 Frame Format Description

Figure 7–1 shows the Ethernet frame format.

Figure 7–1 Ethernet Frame Format



The following list defines the byte field content for the Ethernet frame.

- Preamble—A 7-byte field of 56 alternating 1s and 0s, beginning with a 1.
- SFD (Start frame delimiter)—A 1-byte field that contains the value 10101011; the most-significant bit is transmitted and received first.
- Destination address—A 6-byte field that contains either a specific station address, the broadcast address, or a multicast (logical) address where this frame is directed.
- Source address—A 6-byte field that contains the specific station address where this frame originated.
- Type/Length—A 2-byte field that indicates whether the frame is in IEEE 802.3 format or Ethernet format.

  Table 7–2 lists the possible values for this field. The values are expressed in hexadecimal notation and the 2-byte field is displayed with a hyphen separating the 2 bytes. The byte on the left of the hyphen is the most-significant byte (MSB) and is transmitted first.

Table 7–2 Frame Format Table

Frame Format	Length or Type	Hexadecimal Value
IEEE 802.3	Length field	00-00 to 05-DC
Ethernet	Type field	05-DD to FF-FF

A field smaller or equal to 1500 (05-DC) is interpreted as a length field, which indicates the number of data bytes in the frame.

A field greater than 1500 is interpreted as a type field, which defines the type of protocol of the frame.

- Data—Consists of 46 to 1500 bytes of information that is fully transparent because any arbitrary sequence of bits can occur.

  A data field shorter than 46 bytes, which is specified by the length field, is allowed. Padding is automatically added by the 21040 when transmitting to fill the data field up to 46 bytes.

- FCS (Frame check sequence)—A 32-bit cyclic redundancy check (CRC) value that is computed as a function of the destination address field, source address field, type field, and data field. The FCS is appended to each transmitted frame and used at reception to determine if the received frame is valid.

The CRC polynomial, as specified in the Ethernet specification, is as follows:

$$FCS(X) = X^{31} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} +$$

$$X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X^{1} + \ 1$$

The 32 bits of the CRC value are placed in the FCS field so that the X^{31} term is the right-most bit of the first octet, and the X^{0} term is the left-most bit of the last octet. The bits of the CRC are thus transmitted in the order X^{31}, X^{30}, ... , X^{1}, X^{0}.

7.2.2 Ethernet Reception Addressing

The 21040 can be set up to recognize any one of the Ethernet receive address groups described in Table 7–3. Each group is separate and distinct from the other groups.

Table 7–3 Ethernet Receive Address Groups

Group	Description
1	16 address perfect filtering The 21040 provides support for the perfect filtering of up to 16 Ethernet physical or multicast addresses. Any mix of addresses may be used for this perfect filter function of the 21040. The 16 addresses are issued in setup frames to the 21040.
2	One physical address; unlimited multicast addresses imperfect filtering The 21040 provides support for one, single physical address to be perfectly filtered with an unlimited number of multicast addresses to be imperfectly filtered. This case supports the needs of applications that require one, single physical address to be filtered as the station address, while enabling reception of more than 16 multicast addresses, without suffering the overhead of pass-all-multicast mode. The single physical address for perfect filtering, and a 512-bit mask for imperfect filtering using a hash algorithm, are issued in a setup frame to the 21040. When hash hits are detected, the 21040 delivers the received frame (Section 4.2.3).
3	Unlimited physical addresses; unlimited multicast addresses imperfect filtering The 21040 provides support for unlimited physical addresses to be imperfectly filtered with an unlimited number of multicast addresses to be imperfectly filtered as well. This case supports applications that require more than one physical address to be filtered as the station address, while enabling the reception of more than 16 multicast addresses, without suffering the overhead of pass-all-multicast mode. A 512-bit mask for imperfect filtering using a hash algorithm, is issued in a setup frame to the 21040. When hash hits are detected, the 21040 delivers the received frame (Section 4.2.3).
4	Promiscuous Ethernet reception The 21040 provides support for reception of all frames on the network regardless of their destination. This function is controlled by a CSR bit. This group is typically used for network monitoring.

(continued on next page)

Table 7–3 (Cont.) Ethernet Receive Address Groups

Group	Description
5	16 address perfect filtering and reception of all multicast Ethernet addresses This group augments the receive address Group 1 and also receives all frames on the Ethernet with a multicast address.
6	16 address inverse filtering In this mode, the 21040 applies the reverse filter of Group 1. The 21040 provides support for the rejection of up to 16 Ethernet physical or multicast addresses. Any mix of addresses may be used for this filter function of the 21040. The 16 addresses are issued in setup frames to the 21040.

7.2.3 Collision Detection and Implementation

The Ethernet CSMA/CD network access algorithm is implemented completely within the 21040. In addition to listening for a clear line before transmitting, the 21040 handles collisions in a predetermined way as defined by the standard. If two transmitters attempt to seize the line at the same time, a collision occurs and the data on the line is garbled. The 21040 listens while it transmits, and if a collision is detected, it continues to transmit for a predetermined length of time, thus *jamming* the network and ensuring that all nodes have recognized the collision.

The transmitting 21040 then delays retransmission for a random amount of time according to the Ethernet *truncated-binary-backoff* algorithm so that the colliding nodes do not try to repeatedly access the network at the same time. The 21040 makes up to 16 attempts to access the network before reporting an error due to excessive collisions (Table B–1).

7.2.4 Transmit Mode

In transmit mode, the 21040 initiates a DMA cycle to access data from a transmit buffer. It prefaces the data with a preamble and start frame delimiter (SFD) pattern, adds padding if needed, and then, if enabled, calculates and appends a 32-bit CRC.

After a frame is assembled, the 21040 waits for the internal transmit mechanism to allow transmission on the network, then serializes the data and outputs it to the external SIA.

7.2.5 Receive Mode

In receive mode, the internal SIA sends decoded data and the clock to the internal receive mechanism. The data is then deserialized by the receive mechanism and fed into the internal FIFO.

As the data is received, the address is checked by the 21040 and a CRC is calculated and compared to the CRC checksum at the end of the frame. If the calculated CRC does not agree with the frame CRC, an error bit is set in the receive descriptor. The host processor is notified of all received frames, including those with CRC errors or excessive dribbling errors. These are called *runt* frames and are not delivered to the host unless the 21040 is programmed to do so. Both promiscuous mode (CSR6<6>) and pass bad frames (CSR6<3>) are set.

7.3 Detailed Transmission Operation

This section describes the transmission operation in detail as supported by the 21040. This description includes the specific control register definitions, setup frame definitions, and mechanism used by the host processor software to manipulate the transmit list (that is, the descriptors and buffers which can be found in Section 4.2).

7.3.1 Transmission Initiation

The host CPU initiates a transmission by storing the entire information content of the frame to be transmitted in one or more buffers in memory. The host processor software prepares a companion transmit descriptor, also in host memory, for the transmit buffer and signals the 21040 to take it. After the 21040 has been notified of this transmit list, the 21040 starts to move the data bytes from the host memory to the internal transmit FIFO.

When the transmit FIFO is adequately filled to the programmed threshold level, or when there is a full frame buffered into the transmit FIFO, the 21040 begins to encapsulate the frame.

The threshold level can be programmed with four quantities: 72, 96, 128, and 160 bytes (Table 3–37). The lower threshold is for low bus latency systems and the high threshold is for high bus latency systems.

The transmit encapsulation is performed by the transmit state machine, which delays the actual transmission of the data onto the network until the network has been idle for a minimum interpacket gap (IPG) time (9.6 microseconds).

7.3.2 Frame Encapsulation

The transmit data frame encapsulation stream consists of the following:

64 preamble bits
4 information fields
CRC

The CRC is computed in real time by the 21040 and, if enabled, is automatically appended to the frame at the end of the serial data. The preamble and CRC encapsulation support Ethernet frame format.

For outgoing frames, the following information is prepared by the host processor in the buffer memory before the host CPU initiates transmission:

Destination address
Source address
Type or length field
Data field

The 21040 encapsulates these fields into an Ethernet frame by inserting a preamble before the information fields and appending both padding and an optional CRC after the information fields.

If the data field length is shorter than 46 bytes, the 21040 pads with 00s up to 46 bytes and also appends 4 bytes of CRC - regardless of the state of the add CRC disable (TDES1<26>) flag. This results in a frame of at least 64 bytes.

7.3.3 Initial Deferral

The 21040 constantly monitors the line and can initiate a transmission any time the host CPU requests it. Actual transmission of the data onto the network occurs only if the network has been idle for the 9.6 microsecond IPG time, and any backoff time requirements have been satisfied.

The IPG time is divided into two parts: IPS1 and IPS2.

1. IPS1 time (6.0 microseconds)—The 21040 monitors the network for an idle state. If a carrier is sensed on the serial line during this time, the 21040 defers and waits until the line is idle again before restarting the IPS1 time count.

2. IPS2 time (3.6 microseconds)—The 21040 continues to count time even though a carrier has been sensed on the network, and thus forces collisions on the network. This enables all network stations to have access to the serial line.

7.3.4 Collision

A collision occurs when concurrent transmissions from two or more Ethernet nodes take place. The 21040 halts the transmission of the data bytes in the transmit FIFO and then transmits a jam pattern consisting of hexadecimal AAAAAAAA. At the end of the jam transmission, the 21040 begins the backoff wait period.

If the collision was detected during the preamble transmission, the jam pattern is transmitted after completing the preamble. This results in a minimum 96-bit fragment.

The 21040 scheduling of retransmission is determined by a controlled randomization process called truncated binary exponential backoff. The delay is an integer multiple of slot times. The number of slot times of delay before the n^{th} retransmission attempt is chosen as a uniformly distributed random integer r in the range:

$0 \leq r < 2^k$
k = min (n, N) and N = 10

When 16 attempts have been made at transmission and all have been terminated by a collision, the 21040 sets an error status bit and, if enabled, issues an interrupt to the host.

Note

The jam pattern is a fixed pattern that is not compared with the actual frame CRC. This has the very low probability (0.5^{32}) of having a jam pattern equal to the CRC.

7.3.5 Terminating Transmission

A specific frame transmission is terminated by any of the following conditions:

- Normal—The frame has been transmitted successfully. When the last byte is serialized, the pad and CRC are optionally appended and transmitted, thus concluding frame transmission.
- Underflow—Transmit data is not ready when needed for transmission. The underflow status bits (TDES0<1> and CSR5<5>) are set, and the packet is terminated on the network with a bad CRC.
- Excessive collisions—If a collision occurs for the 15th consecutive retransmission attempt of the same frame, TDES0<8> is set.

- Jabber timer expired—If the timer expires (TDES0<14> sets) while transmission continues, the programmed interval transmission is cut off.
- Memory error—This generic error indicates either a host bus time-out or a host memory error.
- Late collision—If a collision occurs after the collision window (receiving at least 64 bytes), transmission is cut off and TDES0<9> sets.

At the completion of every frame transmission, status information about the frame is written into the transmit descriptor. Status information is written into CSR5 if an error occurs during the operation of the transmit machine itself.

7.3.6 Transmit Parameter Values

The transmit parameter values follow.

- Defer time: IPS1 + IPS2 = 96-bit-time = 9.6 microseconds for 10-megahertz serial bit rate
- IPS1 = 60-bit-time = 6.0 microseconds for 10-megahertz serial bit rate
- IPS2 = 36-bit-time = 3.6 microseconds for 10-megahertz serial bit rate
- Slot time interval = 512-bit-time = 51.2 microseconds for 10-megahertz serial bit rate (collision window)
- Network acquisition time = 512-bit-time = 51.2 microseconds for 10-megahertz serial bit rate, starting from the assertion of TXEN
- Transmission attempts limit: 16
- Backoff limit: 10
- Jabber timer: Default = 1.6 to 2 microseconds, programmable range = 26 to 32 microseconds.
- Append CRC on frame transmission: Programmable (yes or no).

7.4 Detailed Receiving Operation

This section describes the detailed receiving operation as supported by the 21040. This description includes the specific control register definitions, setup frame definitions, and mechanism used by the host processor software to manipulate the receive list (that is, the descriptors and buffers, which can be found in Section 4.2).

7.4.1 Initiating Reception

The 21040 continuously monitors the network when reception is enabled. When activity is recognized by a preamble being detected on the receive data lines, the 21040 synchronizes itself to the incoming data stream during the preamble, waits for the start frame delimiter (SFD), then examines the destination address field of the frame. Depending on the address match mode specified, the 21040 either recognizes the frame as being addressed to itself, or it discards it.

7.4.2 Preamble Processing

The preamble, as defined by Ethernet, can be up to 64 bits (8 bytes) long.

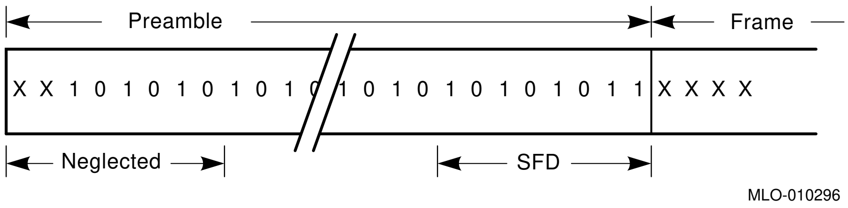
The 21040 allows any arbitrary preamble length, but needs at least 16 bits to recognize a preamble. Recognition occurs as follows:

1. The first 8 preamble bits are ignored.
2. The 21040 checks for the start frame delimiter (SFD) byte content of 10101011.

If the 21040 receives a 00 or a 11 after the first 8 preamble bits and before receiving the SFD, the reception of the current frame is aborted; the frame is not received, and the 21040 waits until the carrier drops and rises again, then begins monitoring the network for a new preamble.

Figure 7–2 shows the preamble recognition sequence bit fields.

Figure 7–2 Preamble Recognition Sequence



7.4.3 Address Matching

Ethernet addresses consist of two 6-byte fields, one field for the destination address and one for the source address. The first bit of the destination address signifies whether it is a physical address or a multicast address (Table 7–4).

Table 7–4 Destination Address Bit 1

Bit 1	Address
0	Station address (physical)
1	Multicast address

The 21040 filters the frame based on the Ethernet receive address group (Section 7.2.2) filtering mode that has been enabled.

If the frame address passes the filter, the 21040 removes the preamble and delivers the frame to the host processor memory. If, however, the address does not pass the filter when the mismatch is recognized, the 21040 terminates its reception. In this case, no data is sent to the host memory nor is any receive buffer consumed.

7.4.4 Frame Decapsulation

The 21040 checks the CRC bytes of all received frames before releasing the frame along with the CRC to the host processor.

7.4.5 Terminating Reception

Reception of a specific frame is terminated when any of the following conditions occur:

- Normal termination—The carrier sense line becomes inactive, indicating that traffic is no longer present on the Ethernet cable.
- Overflow—The receive DMA cannot empty the receive FIFO into host processor memory as rapidly as it is filled, and an error occurs as frame data is lost. The overflow status bit (RDES0<0>) is set.
- Watchdog timer expired—The timer expires (CSR5<9> and RDES0<4> both set) while reception is still in process, and reception is cut off.
- Collision—If a late collision occurs after the reception of 64 bytes of the packet, the collision seen status bit (RDES0<6>) is set.

7.4.6 Frame Reception Conditions

When reception terminates, the 21040 determines the status of the received frame and loads it into the receive status word in the buffer descriptor. An interrupt is issued if enabled. The 21040 may report the following conditions at the end of frame reception:

- Overflow—The 21040 receive FIFO overflowed.

- CRC error—The 32-bit CRC transmitted with the frame did not match the CRC calculated upon reception. The CRC check is always executed and is independent of any other errors.
- Dribbling bits error—This indicates the frame did not end on a byte boundary. The 21040 signals a dribbling bits error only if it detects more than two dribbling bits. Only *whole* bytes are run through the CRC check. This means that although up to seven dribbling bits may have occurred and a framing error was signaled, the frame might nevertheless have been received correctly.
- Alignment error—A CRC error and dribbling bit error occur together. This means that the frame did not contain an integral number of bytes and the CRC check failed.
- Frame too short (runt frame)—A frame containing less than 64 bytes was received (including CRC). Reception of runt frames is optionally selectable. The 21040 defaults to inhibit reception of runts.
- Frame too long—A frame containing more than 1500 bytes was received. Reception of frames too long completes with an error indication.
- Collision seen—A frame collision occurred after the 64 bytes following the start frame delimiter (SFD) were received. Reception of such frames is completed and an error bit is set in the descriptor.
- Descriptor error—An error was found in one of the receive descriptors, which disabled the correct reception of an incoming frame.

7.4.7 Capture Effect

The 21040 provides a complete solution for the capture effect on the network (see capture effect enable CSR6<17>). Capture effect is defined as a station that *captures* a channel for an unfair amount of time, transmitting its packets back-to-back, while another station continues to back off and is unable to transmit its waiting packets. In the next two subsections, the capture effect problem demonstrates how this capture can take place and how it is resolved.

7.4.7.1 What is Capture Effect?

Consider two stations on the line, station A and station B. Each station has a significant amount of data ready to transmit. Each station is able to satisfy the minimum IPG rules (both from transmit-to-transmit and from receive-to-transmit). The following steps show how Station A captures the line (Table 7–5).

1. Station A (with data A1) and station B (with data B1) both attempt to transmit simultaneously within a slot time of 51.2 microseconds. Each station has an initial collision count set to 0.
2. The stations experience a collision. Both stations increment their collision count to 1.
3. Each station picks a backoff time value which is uniformly distributed from 0 to (2n)–1 slots. In this example, station B selects a backoff of 1 (a 50% probability), and station A selects a backoff of 0.
4. Station A successfully transmits its A1 data packet. Station B waits for data A1 to be transmitted before attempting to retransmit data B1.
5. Collision count at station B remains at 1, while collision count at station A is reset to 0.
6. If station A has another packet (data A2) ready to transmit while station B still wants to transmit its packet (data B1), the stations both contend for the line again.
7. If these stations collide, the backoff value available for station A is 0 or 1 slots. The backoff value available for station B is 0, 1, 2, or 3 slots because the collision count is now at 2 (station A's collision count is at 1). Station A is more likely to succeed and transmit data A2, while data B1 from station B begins the deferral of completing its backoff interval.
8. It is possible, with this type of behavior between stations, that in the 2-node Ethernet, a station can capture the channel for an unfair amount of time. One station can transmit a significant number of packets back-to-back, while the other station continues to backoff further and further.
9. This process could continue until station B reaches the maximum number of collisions, 16, while attempting to transmit data B1. At this time, station B would access the line and transmit data B1.

Note

If station A completes the transmitting of a stream of packets during this type of capture, and station B is still in backoff, potentially for a long time, the line is idle for this period of time.

Table 7–5 Capture Effect Sequence

Station A	Line	Station B	Collision Count A	Collision Count B
Transmit packet A1	Collision	Transmit packet B1	0	0
Backoff 0, 1		Backoff 0, 1	1	1
Transmit packet A1	Packet A1	Backoff	0	1
Transmit packet A2	Collision	Transmit packet B1	0	1
Backoff 0, 1		Backoff 0, 1, 2, 3	1	2
Transmit packet A2	Packet A2	Backoff	0	2
Transmit packet A3	Collision	Transmit packet B1	0	2
Backoff 0, 1		Backoff 0, 1, 2, ... 7	1	3

7.4.7.2 Resolving Capture Effect

The 21040 generally resolves the capture effect by having the station use, after a successful transmission of a frame by a station, a 2-0 backoff algorithm on the next transmit frame. If the station senses a frame on the network before it attempts to transmit the next frame, regardless of whether the sensed frame destination address matches the station's source address, the station returns to use the standard truncated binary exponential backoff algorithm (Section 7.3.4).

When the station executes the 2-0 backoff algorithm, it always waits for a 2-slot period on the first collision, and for a 0-slot period on the second collision. For retransmission attempts greater than 2, it uses the standard truncated, binary exponential backoff algorithm.

Table 7–6 summarizes the 2-0 backoff algorithm.

Table 7–6 2-0 Backoff Algorithm

Retransmission Attempts	Backoff Period (Number of Slot Times)
n = 1	Backoff = 2 slots
n = 2	Backoff = 0 slots
n = 3-15	Backoff = $0 \leq r < 2^k$ k = min (n, N) and N = 10 r = uniformly distributed random integer

7.4.8 Back Pressure Operation

The 21040 implements the congested receiver algorithm when the 21040 is in the suspended state as a result of receive buffer unavailable. It starts to back pressure the transmit line. Back pressure starts only during the transition from running to suspended (receive buffer available to receive buffer unavailable), when no packet is available for transmission.

A programmable bit, pass bad frames (CSR6<3>), is implemented to activate the back pressure logic and is effective only during half-duplex operation mode.

Back pressure starts 3 microseconds after the carrier has deasserted.

After back pressure is activated, it stops only under the following conditions:

- Receive poll demand is issued to indicate to the 21040 that it has new free receive buffers.
- Back pressure time-out counter expires during the period from 450 to 500 milliseconds.

Caution

Back pressure violates the IEEE 802.3 standard.

7.4.9 External SIA Port

The 21040 provides a full connection external SIA port to an external device that bypasses the internal AUI and 10BASE-T ports. This port includes the following signals:

EXT_TX—Transmit data
EXT_RX—Receive data
EXT_TCLK—Transmit clock
EXT_TXEN—Transmit enable
EXT_RXEN—Receive enable
EXT_RCLK—Receive clock
EXT_CLSN—Collision detect

A

Joint Test Action Group Test Logic

This appendix describes the joint test action group (JTAG) test logic and the associated registers (instruction, bypass, and boundary scan).

A.1 General Description

JTAG test logic supports testing, observing, and modifying circuit activity during the components' normal operation. As a PCI device, the 21040 supports the IEEE standard 1149.1 *test access port and boundary scan architecture*. The IEEE 1149.1 standard specifies the rules and permissions that govern the design of the 21040 JTAG test logic support. Inclusion of JTAG test logic allows boundary scan to be used to test both the device and the board where it is installed. The JTAG test logic consists of the following four pins to serially interface within the 21040 (Table 2–1).

TCK—JTAG clock
TDI—Test data and instructions in
TDO—Test data and instructions out
TMS—Test mode select

Note

If JTAG test logic is not implemented, the TDI, TMS, and TCK pins should be connected to ground. TDO should remain unconnected.

These test pins operate in the same electrical environment as the 21040 PCI I/O buffers.

The system vendor is responsible for the design and operation of the 1149.1 serial chains (rings) required in the system. Typically, an 1149.1 ring is created by connecting one device's TDO pin to another device's TDI pin to create a serial chain of devices. In this application, the 21040 receives the same TCK and TMS signals as the other devices. The entire 1149.1 ring is connected to either a motherboard test connector for test purposes or to a resident 1149.1 controller.

Note

To understand the description of the 21040 JTAG test logic in this section, the system designer should be familiar with the IEEE 1149.1 standard.

A.2 Registers

In JTAG test logic design, three registers are implemented through all the 21040 pads:

Instruction register
Bypass register
Boundary scan register

A.2.1 Instruction Register

The 21040 JTAG test logic instruction register is a 3-bit (IR<2:0>) scan type register that is used to direct the JTAG machine to the appropriate operating JTAG mode (Table A–1). Its contents are interpreted as test instructions. The test instructions select the boundary scan registers for serial transfer of test data by using the TDI and TDO pins. These instructions also control the operation of the selected test features.

Table A–1 Instruction Register

IR<2>	IR<1>	IR<0>	Description
0	0	0	Extest mode (mandatory instruction) allows testing of the 21040 board-level interconnections. Test data is shifted into the boundary scan register of the 21040 and then transferred in parallel to the output pins.
0	0	1	Sample-preload mode (mandatory instruction) allows the 21040 JTAG boundary scan register to be initialized prior to selecting other instructions such as EXTEST. It is also possible to capture data at system pins while the system is running, and to shift that data out for examination.
0	1	0	Reserved.
0	1	1	Reserved.
1	0	0	Reserved.

(continued on next page)

Table A–1 (Cont.) Instruction Register

IR<2>	IR<1>	IR<0>	Description
1	0	1	Tristate mode (optional instruction) allows the 21040 to enter power save mode. When this occurs, all internal clocks are frozen, all external pads enter tristate, and internal phase lock loops reduce their power consumption. All the SIA transceivers continue to operate normally without any power reduction.
1	1	0	Continuity mode (optional instruction) allows the 21040 continuity test while in production.
1	1	1	Bypass mode (mandatory instruction) allows the test features on the 21040 JTAG test logic to be bypassed. This instruction selects the bypass register to be connected between TDI and TDO. When the bypass mode is selected, the operation of the test logic has no effect on the operation of the system logic. Bypass mode is selected automatically, when power is applied.

A.2.2 Bypass Register

The bypass register is a 1-bit shift register that provides a single-bit serial connection between the TDI and TDO pins when either no other test data register in the 21040 JTAG test logic registers is selected, or the test logic in the 21040 JTAG is bypassed. When power is applied, JTAG test logic resets and then is set to bypass mode.

A.2.3 Boundary Scan Register

The JTAG boundary scan register consists of cells located at the PCI and the EXT_SIA pins of the 21040. This register provides an interconnections test. It also provides additional control and observation of the 21040 pins during the testing phases. For example, the 21040 boundary scan register can observe the output enable control signals of the I/O pads; for example, AD_OE, CBE_OE, and so on. When these signals are programmed to be 1 during EXTEST mode, data is applied to the output from the selected pins.

The following listing contains boundary scan register pads order.

Note

There are no boundary scan register pads on the TP or AUI ports.

```
-> TDI          -> AD_OE        -> AD<0>        -> AD<1>        -> AD<2>
-> AD<3>        -> AD<4>        -> AD<5>        -> AD<6>        -> AD<7>
-> AD<8>        -> AD<9>        -> AD<10>       -> AD<11>       -> AD<12>
-> AD<13>       -> AD<14>       -> AD<14>       -> AD<16>       -> IDSEL
-> AD<17>       -> AD<18>       -> AD<19>       -> AD<20>       -> AD<21>
-> AD<22>       -> AD<23>       -> AD<24>       -> AD<25>       -> AD<26>
-> AD<27>       -> AD<28>       -> AD<29>       -> AD<30>       -> AD<31>
-> CBE_OE       -> C_BE_L<0>    -> C_BE_L<1>    -> C_BE_L<2>    -> C_BE_L<3>
-> FRAME_OE     -> FRAME_L      -> DEVSEL_OE    -> DEVSEL_L     -> STOP_OE
-> STOP_L       -> PAR_OE       -> PAR          -> IRDY_OE      -> IRDY_L
-> TRDY_OE      -> TRDY_L       -> REQ_L        -> PERR_OE      -> PERR_L
-> SERR_L       -> INT_L        -> GNT_L        -> SCLK         -> SRST
-> SDIN         -> RST_L        -> CLK          -> GNDCEL       -> EXT_OE1
-> EXT_TXEN(1)  -> EXT_OE2      -> EXT_TCLK (2) -> EXT_TX (1)   -> EXT_RCLK (2)
-> EXT_OE3      -> EXT_RXEN (3)-> EXT_RX (3)    -> EXT_CLSN (3) -> TDO
```

Note

The boundary scan cell named GNDCEL is connected to device pin 90. This pin was used for Digital engineering evaluation and must be connected to VSS for the normal operation of the 21040.

Table A–2 lists the boundary scan register controls.

Table A–2 Boundary Scan Register Controls

Signal	Pin Control
AD_OE	AD<31:0>
CBE_OE	C_BE_L<3:0>
FRAME_OE	FRAME_L
DEVSEL_OE	DEVSEL_L
STOP_OE	STOP_L
PAR_OE	PAR_L
IRDY_OE	IRDY_L
TRDY_OE	TRDY_L
PERR_OE	PERR_L
EXT_OE1	EXT_TXEN, EXT_TX
EXT_OE2	EXT_TCLK, EXT_RCLK
EXT_OE3	EXT_RXEN, EXT_RX, EXT_CLSN

A.2.4 Test Access Port Controller

The test access port (TAP) controller interprets IEEE P1149.1 protocols received on the TMS pin. The TAP controller generates clocks and control signals to control the operation of the test logic. The TAP controller consists of a state machine and control dispatch logic. The 21040 fully implements the TAP state machine as described in the IEEE P1149.1 standard.

B

DNA CSMA/CD Counters and Events Support

This appendix describes the 21040 features that support the driver in implementing and reporting the specified counters and events.[1] CSMA/CD[2] specified events can be reported by the driver based on these features. Table B–1 lists the counters and features.

Table B–1 CSMA/CD Counters

Counter	21040 Feature
Time since creation counter	Supported by the host driver.
Bytes received	Driver must add the frame length (RDES0<30:16>) fields of all successfully received frames.
Bytes sent	Driver must add the buffer 1 size (TDES1<10:0>) and buffer 2 size (TDES1<21:11>) fields of all successfully transmitted buffers.
Frames received	Driver must count the successfully received frames in the receive descriptor list.
Frames sent	Driver must count the successfully transmitted frames in the transmit descriptor list.
Multicast bytes received	Driver must add the frame length (RDES0<30:16>) fields of all successfully received frames with multicast frame (RDES0<10>) set.
Multicast frames received	Driver must count the successfully received frames with multicast frame (RDES<10>) set.

(continued on next page)

[1] As specified in the *DNA Maintenance Operations (MOP) Functional Specification*, Version T.4.0.0, 28 January 1988.

[2] Carrier-sense multiple access with collision detection

Table B–1 (Cont.) CSMA/CD Counters

Counter	21040 Feature
Frames sent, initially deferred	Driver must count the successfully transmitted frames where deferred (TDES0<0>) is set.
Frames sent, single collision	Driver must count the successfully transmitted frames where collision count (TDES0<6:3>) is equal to 1.
Frames sent, multiple collisions	Driver must count the successfully transmitted frames where collision count (TDES0<6:3>) is greater than 1.
Send failure, excessive collisions	Driver must count the transmit descriptors where the excessive collisions (TDES0<8>) bit is set.
Send failure, carrier check failed	Driver must count the transmit descriptors where both late collision (TDES0<9>) and loss of carrier (TDES0<11>) are set.
Send failure, short circuit	There were two successive transmit descriptors where the no_carrier flag (TDES0<10>) is set. This indicates a short circuit.
Send failure, open circuit	There were two successive transmit descriptors where the excessive_collisions flag (TDES0<8>) is set. This indicates an open circuit.
Send failure, remote failure to defer	Flagged as a late collision (TDES0<9>) in the transmit descriptors.
Receive failure, block check error	Driver must count the receive descriptors where CRC error (RDES0<1>) is set and dribbling bit (RDES0<2>) is cleared.
Receive failure, framing error	Driver must count the receive descriptors where both CRC error (RDES0<1>) and dribbling bit (RDES0<2>) are set.
Receive failure, frame too long	Driver must count the receive descriptors where frame too long (RDES0<7>) is set.
Unrecognized frame destination	Not applicable.
Data overrun	Driver must count the receive descriptors where overflow (RDES0<0>) is set.
System buffer unavailable	Reported in the missed frame counter CSR8<15:0> (Section 3.2.1.8).
User buffer unavailable	Kept by the driver.

(continued on next page)

Table B–1 (Cont.) CSMA/CD Counters

Counter	21040 Feature
Collision detect check failed	Driver must count the transmit descriptors where heartbeat fail (TDES0<7>) is set.

C

Hash C Routine

This appendix provides an example of a C routine that generates the hash index for a given Ethernet address.

```
#include <stdio>

unsigned HashIndex (char *Address);

main (int argc, char *argv[]) {

    int Index;
    char m[6];

    if (argc < 2) {
       printf("usage: hash xx-xx-xx-xx-xx-xx\n");
       return;
    }

    sscanf(argv[1],"%2X-%2X-%2X-%2X-%2X-%2X",
       &m[0],&m[1],&m[2],
       &m[3],&m[4],&m[5]);

    Index = HashIndex(&m[0]);

    printf("hash_index = %d  byte: %d   bit: %d\n",
            Index,Index/8,Index%8);
}


unsigned HashIndex (char *Address) {

     unsigned Crc = 0xffffffff;
     unsigned const POLY  0x04c11db6
     unsigned Msb;
     int BytesLength = 6;

     unsigned char CurrentByte;
     unsigned Index;
     int Bit;
     int Shift;

     for (BytesLength=0; BytesLength<6; BytesLength++) {

        CurrentByte = Address[BytesLength];
```

```
        for (Bit=0; Bit<8 ; Bit++) {

            Msb = Crc >> 31;
            Crc  <<= 1;

            if ( Msb ^ (CurrentByte & 1)) {

                 Crc ^= POLY;
                 Crc |= 0x00000001;
            }
            CurrentByte >>= 1;
        }
     }

    /* the hash index is given by the upper 9 bits of the CRC
     * taken in decreasing order of significance
     * index<0> = crc<31>
     * index<1> = crc<30>
     * ...
     * index<9> = crc<23>
     */

     for (Index=0, Bit=23, Shift=8;
         Shift >= 0;
         Bit++, Shift--)  {
            Index |= ( ( (Crc>>Bit) & 1 ) << Shift );
     }

     return Index;

}
```

D

Technical Support, Ordering, and Associated Literature

This appendix provides information about

- Obtaining DECchip information and technical support
- Ordering DECchip products and associated literature

D.1 Calling the DECchip Information Line for Information and Technical Support

Call the DECchip Information Line for information and technical support:

United States and Canada	**1–800–332–2717**
TTY (United States only)	**1–800–332–2515**
Outside North America	**+1–508–568–6868**

D.2 Ordering DECchip Products

To order the DECchip 21040 Ethernet LAN Controller for PCI or the DECchip 21040 Evaluation Board kit, contact your local Digital sales office. Your sales representative may be able to help you take advantage of discounts and volume pricing.

You can order the following DECchip products from Digital.

Product	Order Number
DECchip 21040 Ethernet LAN Controller for PCI	21040–AA
DECchip 21040 Evaluation Board Kit	21040–01

D.3 Ordering Associated DECchip Literature

The following table lists some of the DECchip literature that is available. For a complete list and for information about ordering, contact the DECchip Information Line.

Title	Order Number
DECchip 21040 Ethernet LAN Controller for PCI Product Brief	EC–N0281–72
DECchip 21040 Ethernet LAN Controller for PCI Data Sheet	EC–N0280–72
DECchip 21040 PCI Evaluation Board User’s Guide	EC–N0753–72
Connecting the DECchip 21040 Ethernet LAN Controller to the Network: An Application Note	EC–N0737–72
Ethernet Address ROM Programming: An Application Note	EC–N3214–72

Index

C

D

E

F

O

P

R

S

T

W