![CRYSTAL - A Cirrus Logic Company]

CS8900

Highly-Integrated ISA Ethernet Controller

Features

- Single-Chip IEEE 802.3 Ethernet Controller with Direct ISA-Bus Interface
- Efficient PacketPage™ Architecture Operates in I/O and Memory Space, and as DMA Slave
- Full Duplex Operation
- On-Chip RAM Buffers Transmit & Receive Frames
- 10BASE-T Port with Analog Filters, Provides:
  - Automatic Polarity Detection and Correction
- AUI Port for 10BASE2, 10BASE5 and 10BASE-F
- Programmable Transmit Features:
  - Automatic Re-transmission on Collision
  - Automatic Padding and CRC Generation
- Programmable Receive Features:
  - StreamTransfer™ for Reduced CPU Overhead
  - Auto-Switch Between DMA and On-Chip Memory
  - Early Interrupts for Frame Pre-Processing
  - Automatic Rejection of Erroneous Packets
- EEPROM Support for Jumperless Configuration
- Boot PROM Support for Diskless Systems
- Boundary Scan and Loopback Test
- LED Drivers for Link Status and LAN Activity
- Standby and Suspend Sleep Modes

Description

The CS8900 is a low-cost Ethernet LAN Controller optimized for Industry Standard Architecture (ISA) Personal Computers. Its highly-integrated design eliminates the need for costly external components required by other Ethernet controllers. The CS8900 includes on-chip RAM, 10BASE-T transmit and receive filters, and a direct ISA-Bus interface with 24 mA Drivers.

In addition to high integration, the CS8900 offers a broad range of performance features and configuration options. Its unique PacketPage architecture automatically adapts to changing network traffic patterns and available system resources. The result is increased system efficiency and minimized CPU overhead.

The CS8900 is available in a thin 100-pin TQFP package ideally suited for small form-factor, cost-sensitive Ethernet applications, such as desktop and portable motherboards and adapters. With the CS8900, system engineers can design a complete Ethernet circuit that occupies less than 1.5 square inches (10 sq. cm) of board space.

ORDERING INFORMATION:

CS8900-CQ	0 to 70° C	100-pin TQFP
CRD8900-1		Evaluation Kit
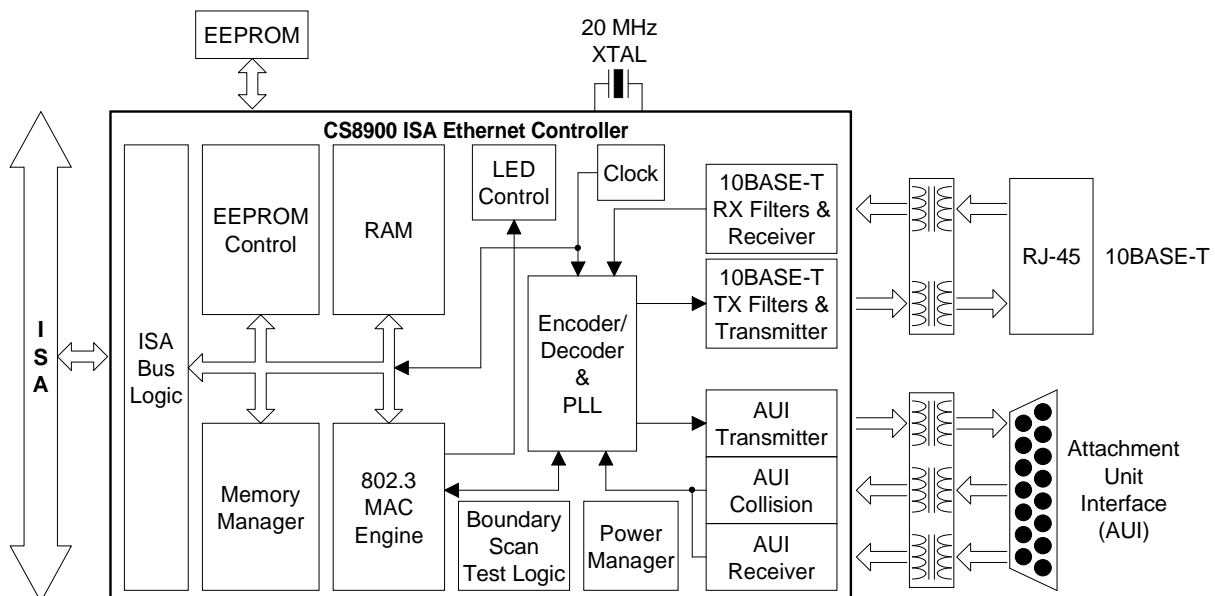
Preliminary Product Information | This document contains information for a new product. Crystal Semiconductor reserves the right to modify this product without notice.

CONTENTS

1.0 INTRODUCTION

1.1 General Description

The CS8900 is a true single-chip, full-duplex, Ethernet solution, incorporating all of the analog and digital circuitry needed for a complete Ethernet circuit. Major functional blocks include: a direct ISA-bus interface; an 802.3 MAC engine; integrated buffer memory; a serial EEPROM interface; and a complete analog front end with both 10BASE-T and AUI.

Direct ISA-Bus Interface

Included in the CS8900 is a direct ISA-bus interface with full 24 mA drive capability. Its configuration options include a choice of four interrupts and three DMA channels (one of each selected during initialization). In Memory Mode, it supports Standard or Ready Bus cycles without introducing additional wait states.

Integrated Memory

The CS8900 incorporates a 4-Kbyte page of on-chip memory, eliminating the cost and board area associated with external memory chips. Unlike most other Ethernet controllers, the CS8900 buffers entire transmit and receive frames on chip, eliminating the need for complex, inefficient memory management schemes. In addition, the CS8900 operates in either Memory space, I/O space, or with external DMA controllers, providing maximum design flexibility.

802.3 Ethernet MAC Engine

The CS8900's Ethernet Media Access Control (MAC) engine is fully compliant with the IEEE 802.3 Ethernet standard (ISO/IEC 8802-3, 1993), and supports full-duplex operation. It handles all aspects of Ethernet frame transmission and reception, including: collision detection, preamble generation and detection, and CRC generation

and test. Programmable MAC features include automatic re-transmission on collision, and automatic padding of transmitted frames.

EEPROM Interface

The CS8900 provides a simple and efficient serial EEPROM interface that allows configuration information to be stored in an optional EEPROM, and then loaded automatically at power-up. This eliminates the need for costly and cumbersome switches and jumpers.

Complete Analog Front End

The CS8900's analog front end incorporates a Manchester encoder/decoder, clock recovery circuit, 10BASE-T transceiver, and complete Attachment Unit Interface (AUI). It provides manual and automatic selection of either 10BASE-T or AUI, and offers three on-chip LED drivers for link status, bus status, and Ethernet line activity.

The 10BASE-T transceiver includes drivers, receivers, and analog filters, allowing direct connection to low-cost isolation transformers. It supports 100, 120, and 150 Ω shielded and unshielded cables, extended cable lengths, and automatic receive polarity reversal detection and correction.

The AUI port provides a direct interface to 10BASE-2, 10BASE-5 and 10BASE-FL networks, and is capable of driving a full 50-meter AUI cable.

1.2 System Applications

The CS8900 is designed to work well in either motherboard or adapter applications.

Motherboard LANs

The CS8900 requires the minimum number of external components needed for a full Ethernet
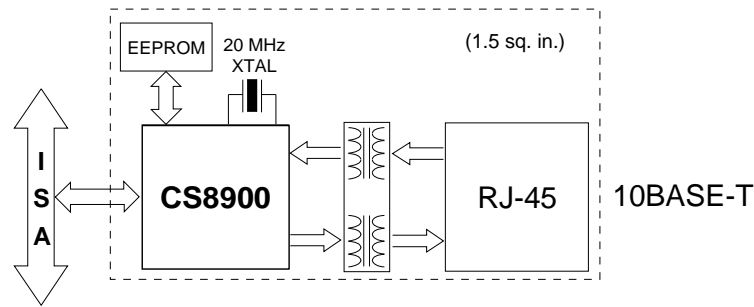
Figure 1.1. Complete Ethernet Motherboard Solution

node. Its small-footprint package and high level of integration allow System Engineers to design a complete Ethernet circuit that occupies as little as 1.5 square inches of PCB area (Figure 1.1). In addition, the CS8900's power-saving features and CMOS design make it a perfect fit for power-sensitive portable and desktop PCs. Motherboard design options include:

- An EEPROM can be used to store node-specific information, such as the Ethernet Individual Address and node configuration.

- The 20 MHz crystal oscillator may be replaced by a 20 MHz clock signal.

Ethernet Adapter Cards

The CS8900's highly efficient PacketPage architecture, with StreamTransfer[TM] and Auto-Switch

DMA options, make it an excellent choice for high-performance, low-cost ISA adapter cards (Figure 1.2). The CS8900's wide range of configuration options and performance features allow engineers to design Ethernet solutions that meets their particular system requirements. Adapter card design options include:

- A Boot PROM can be added to support diskless applications.

- The 10BASE-T transmitter and receiver impedance can be adjusted to support 100, 120, or 150 Ω twisted pair cables.

- An external Latchable-Address-bus decode circuit can be added to operate the CS8900 in Upper-Memory space.
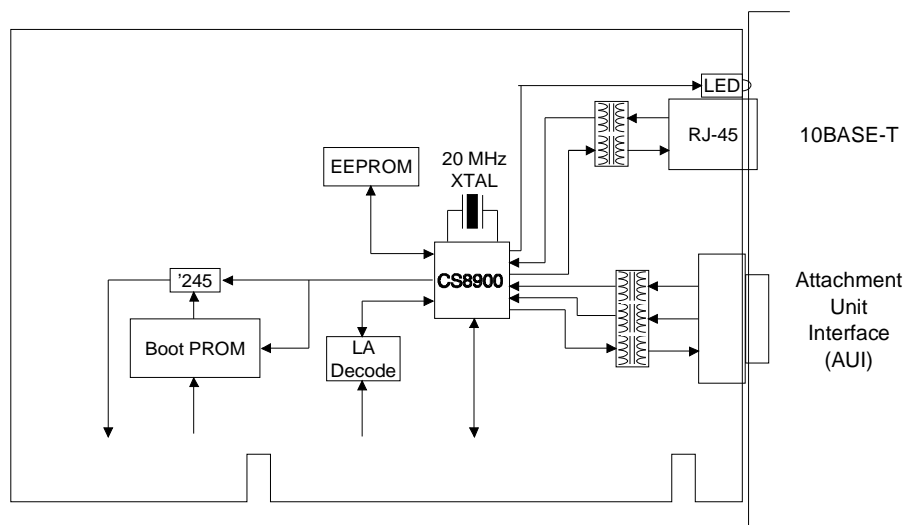
Figure 1.2. Full-Featured ISA Adapter Solution

- On-chip LED ports can be used for either optional LEDs, or as programmable outputs.

1.3   Key Features and Benefits

Very Low Cost

The CS8900 is designed to provide the lowest-cost Ethernet solutions available for ISA desktop motherboards, portable motherboards, and adapter cards. Cost-saving features include:

- Integrated RAM eliminates the need for expensive external memory chips.

- On-chip 10BASE-T filters allow designers to use simple isolation transformers instead of more costly filter/transformer packages.

- The serial EEPROM port, used for configuration and initialization, eliminates the need for expensive switches and jumpers.

- The CS8900 is designed to be used on a 2-layer circuit board instead of a more expensive multi-layer board.

- The 8900-based solution offers the smallest footprint available, saving valuable printed circuit board area.

- A set of certified software drivers is available at no charge, eliminating the need for costly software development.

High Performance

The CS8900 is a full 16-bit Ethernet controller designed to provide optimal system performance by minimizing time on the ISA bus and CPU overhead per frame. It offers equal or superior performance for less money when compared to other Ethernet controllers. The CS8900's PacketPage architecture allows software to select whichever access method is best suited to each particular CPU/ISA-bus configuration. When compared to older I/O-space designs, PacketPage is faster, simpler and more efficient.

To boost performance further, the CS8900 include several key features that increase throughput and lower CPU overhead, including:

- StreamTransfer cuts up to 87% of interrupts to the host CPU during large block transfers.

- Auto-Switch DMA allows the CS8900 to maximize throughput while minimizing missed frames.

- Early interrupts allow the host to preprocess incoming frames.

- On-chip buffering of full frames cuts the amount of host bandwidth needed to manage Ethernet traffic.

Low Power and Low Noise

For low power needs, the CS8900 offers three power-down options: Hardware Standby, Hardware Suspend, and Software Suspend. In Standby mode, the chip is powered down with the exception of the 10BASE-T receiver, which is enabled to listen for link activity. In either Hardware or Software Suspend mode, the receiver is disabled and power consumption drops to the micro-ampere range.

In addition, the CS8900 has been designed for very low noise emission, thus shortening the time required for EMI testing and qualification.

Complete Support

The CS8900 comes with a suite of software drivers for immediate use with most industry standard network operating systems. In addition, complete evaluation kits and manufacturing packages are available, significantly reducing the cost and time required to produce new Ethernet products.

CRYSTAL
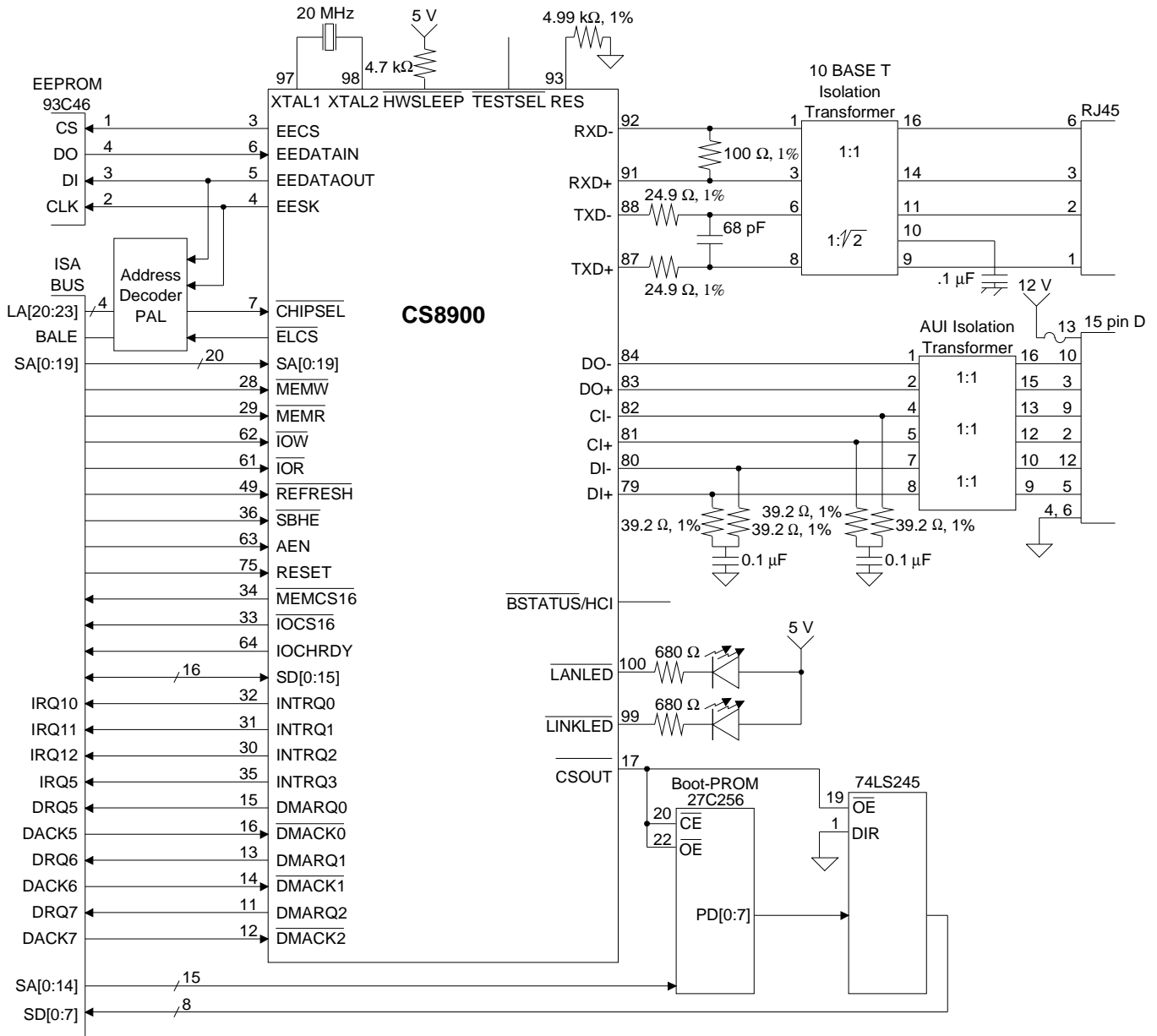
Figure 1.3. Typical Connection Diagram

2.0 PIN DESCRIPTION

2.1 Pin Diagram
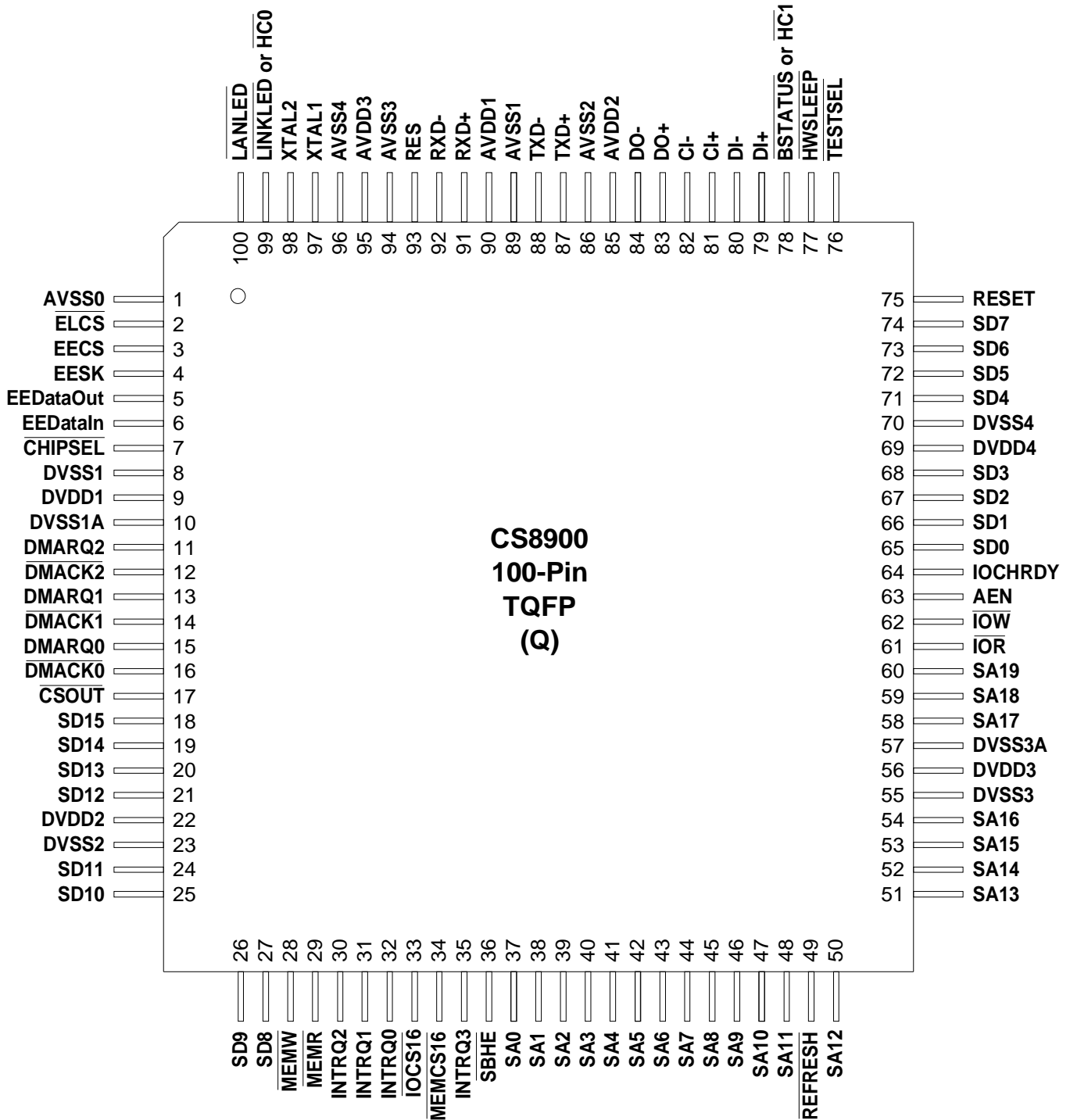
2.2 Pin Description

ISA Bus Interface

Symbol	Pin Number	Type	Description
SA0-SA11 SA12-SA16 SA17-SA19	37-48 50-54 58-60	I	**System Address Bus:** Lower 20 bits of the 24-bit System Address Bus used to decode accesses to CS8900 I/O and Memory space, and attached Boot PROM. SA0-SA15 are used for I/O Read and Write operations. SA0-SA19 are used in conjunction with external decode logic for Memory Read and Write operations.
SD0-SD3 SD4-SD7 SD8-SD11 SD12-SD15	65-68 71-74 27-24 21-18	B24	**System Data Bus:** Bi-directional 16-bit System Data Bus used to transfer data between the CS8900 and the host.
RESET	75	I	**Reset:** Active-high asynchronous input used to reset the CS8900. Must be stable for at least 400 ns before the CS8900 recognizes the signal as a valid reset.
AEN	63	I	**Address Enable:** When TESTSEL is high, this active-high input indicates to the CS8900 that the system DMA controller has control of the ISA bus. When AEN is high, the CS8900 will not perform slave I/O space operations. When TESTSEL is low, this pin becomes the shift clock input for the Boundary Scan Test.
MEMR	29	I	**Memory Read:** Active-low input indicates that the host is executing a Memory Read operation.
MEMW	28	I	**Memory Write:** Active-low input indicates that the host is executing a Memory Write operation.
MEMCS16	34	OD24	**Memory Chip Select 16-bit:** Open-drain, active-low output generated by the CS8900 when it recognizes an address on the ISA bus that corresponds to its assigned Memory space (CS8900 must be in Memory Mode with the MemoryE bit (Register 17, BusCTL, Bit A) set for MEMCS16 to go active). Tri-stated when not active.
REFRESH	49	I	**Refresh:** Active-low input indicates to the CS8900 that a DRAM refresh cycle is in progress. When REFRESH is low, MEMR, MEMW, IOR, IOW, DMACK0, DMACK1, and DMACK2 are ignored.
IOR	61	I	**I/O Read:** When IOR is low and a valid address is detected, the CS8900 outputs the contents of the selected 16-bit I/O register onto the System Data Bus. IOR is ignored if REFRESH is low.
IOW	62	I	**I/O Write:** When IOW is low and a valid address is detected, the CS8900 writes the data on the System Data Bus into the selected 16-bit I/O register. IOW is ignored if REFRESH is low.
IOCS16	33	OD24	**I/O Chip Select 16-bit:** Open-drain, active-low output generated by the CS8900 when it recognizes an address on the ISA bus that corresponds to its assigned I/O space. Tri-stated when not active.

Pin Types:

dI	=	Differential Input Pair	I	=	Input	G	=	Ground
dO	=	Differential Output Pair	O	=	Output	ts	=	Tri-State
B	=	Bi-Directional with Tri-State Output	P	=	Power	w	=	Internal Weak Pullup
OD	=	Open Drain Output						

Digital outputs are followed by drive in mA (Example: OD24 = Open Drain Output with 24 mA drive).

ISA Bus Interface (continued)

Symbol	Pin Number	Type	Description
IOCHRDY	64	OD24	**I/O Channel Ready:** When driven low, this open-drain, active-high output extends I/O Read and Memory Read cycles to the CS8900. This output is functional when the IOCHRDYE bit in the Bus Control register (Register 17) is clear. This pin is always tri-stated when the IOCHRDYE bit is set.
SBHE	36	I	**System Bus High Enable:** Active-low input indicates a data transfer on the high byte of the System Data Bus (SD8-SD15).
INTRQ0 INTRQ1 INTRQ2 INTRQ3	32 31 30 35	O24ts	**Interrupt Request:** Active-high output indicates the presence of an interrupt event. Interrupt Request goes low once the Interrupt Status Queue (ISQ) is read as all 0's. Only one Interrupt Request output is used (one is selected during configuration). All non-selected Interrupt Request outputs are placed in a high-impedance state. (See sections 3.2 and 5.1.)
DMARQ0 DMARQ1 DMARQ2	15 13 11	O24ts	**DMA Request:** Active-high, tri-stateable output used by the CS8900 to request a DMA transfer. Only one DMA Request output is used (one is selected during configuration). All non-selected DMA Request outputs are placed in a high-impedance state.
DMACK0 DMACK1 DMACK2	16 14 12	I	**DMA Acknowledge:** Active-low input indicates acknowledgment by the host of the corresponding DMA Request output.
CHIPSEL	7	I	**Chip Select:** Active-low input generated by external Latchable Address bus decode logic when a valid memory address is present on the ISA bus. If Memory Mode operation is not needed, CHIPSEL should be tied low.

EEPROM and Boot PROM Interface

Symbol	Pin Number	Type	Description
EESK	4	O4	**EEPROM Serial Clock:** Serial clock used to clock data into or out of the EEPROM.
EECS	3	O4	**EEPROM Chip Select:** Active-high output used to select the EEPROM.
EEDataIn	6	Iw	**EEPROM Data In:** Serial input used to receive data from the EEPROM. Connects to the DO pin on the EEPROM. EEDataIn is also used to sense the presence of the EEPROM.
ELCS	2	B4w	**External Logic Chip Select:** Bi-directional signal used to configure external Latchable Address (LA) decode logic. If external LA decode logic is not needed, ELCS should be tied low.
EEDataOut	5	O4	**EEPROM Data Out:** Serial output used to send data to the EEPROM. Connects to the DI pin on the EEPROM. When TESTSEL is low, this pin becomes the output for the Boundary Scan Test.
CSOUT	17	O4	**Chip Select for External Boot PROM:** Active-low output used to select an external Boot PROM when the CS8900 decodes a valid Boot PROM memory address.

Pin Types:

dI	=	Differential Input Pair	I	=	Input	G	=	Ground
dO	=	Differential Output Pair	O	=	Output	ts	=	Tri-State
B	=	Bi-Directional with Tri-State Output	P	=	Power	w	=	Internal Weak Pullup
OD	=	Open Drain Output						

Digital outputs are followed by drive in mA (Example: OD24 = Open Drain Output with 24 mA drive).

10BASE-T Interface

Symbol	Pin Number	Type	Description
TXD+ TXD-	87 88	dO	**10BASE-T Transmit:** Differential output pair drives 10 Mb/s Manchester-encoded data to the 10BASE-T transmit pair.
RXD+ RXD-	91 92	dI	**10BASE-T Receive:** Differential input pair receives 10 Mb/s Manchester-encoded data from the 10BASE-T receive pair.

Attachment Unit Interface (AUI)

Symbol	Pin Number	Type	Description
DO+ DO-	83 84	dO	**AUI Data Out:** Differential output pair drives 10 Mb/s Manchester-encoded data to the AUI transmit pair.
DI+ DI-	79 80	dI	**AUI Data In:** Differential input pair receives 10 Mb/s Manchester-encoded data from the AUI receive pair.
CI+ CI-	81 82	dI	**AUI Collision In:** Differential input pair connects to the AUI collision pair. A collision is indicated by the presence of a 10 MHz +/-15% signal with duty cycle no worse than 60/40.

Pin Types:

dI	=	Differential Input Pair	I	=	Input	G	=	Ground
dO	=	Differential Output Pair	O	=	Output	ts	=	Tri-State
B	=	Bi-Directional with Tri-State Output	P	=	Power	w	=	Internal Weak Pullup
OD	=	Open Drain Output						

Digital outputs are followed by drive in mA (Example: OD24 = Open Drain Output with 24 mA drive).

General Pins

Symbol	Pin Number	Type	Description
XTAL1 XTAL2	97 98	I/O	**Crystal:** A 20 MHz crystal should be connected across these pins. If a crystal is not used, a 20 MHz signal should be connected to XTAL1 and XTAL2 should be left open. (See section 9.0 and 13.0.)
SLEEP	77	I$_W$	**Hardware Sleep:** Active-low input used to enable the two hardware sleep modes: Hardware Suspend and Hardware Standby. (See section 3.7.)
LINKLED or HC0	99	OD10	**Link Good LED or Host Controlled Output 0:** When the HCE0 bit of the Self Control register (Register 15) is clear, this active-low output is low when the CS8900 detects the presence of valid link pulses. When the HCE0 bit is set, the host may drive this pin low by setting the HCBO in the Self Control register.
BSTATUS or HC1	78	OD10	**Bus Status or Host Controlled Output 1:** When the HCE1 bit of the Self Control register (Register 15) is clear, this active-low output is low when receive activity causes an ISA bus access. When the HCE1 bit is set, the host may drive this pin low by setting the HCB1 in the Self Control register.
LANLED	100	OD10	**LAN Activity LED:** During normal operation, this active-low output goes low for 6 ms whenever there is a receive packet, a transmit packet, or a collision. During Hardware Standby mode, this output is driven low when the receiver detects network activity.
TESTSEL	76	I$_W$	**Test Enable:** Active-low input used to put the CS8900 in Boundary Scan Test mode. For normal operation, this pin should be high.
RES	93	I	**Reference Resistor:** This input should be connected to a 4.99 KΩ +/-1% resistor needed for biasing of internal analog circuits.
DVDD1 DVDD2 DVDD3 DVDD4	9 22 56 69	P	**Digital Power:** Provides 5 V +/- 5% power to the digital circuits of the CS8900.
DVSS1 DVSS1A DVSS2 DVSS3 DVSS3A DVSS4	8 10 23 55 57 70	G	**Digital Ground:** Provides ground reference (0 V) to the digital circuits of the CS8900.
AVDD1 AVDD2 AVDD3	90 85 95	P	**Analog Power:** Provides 5 V +/- 5% power to the analog circuits of the CS8900.
AVSS0 AVSS1 AVSS2 AVSS3 AVSS4	1 89 86 94 96	G	**Analog Ground:** Provide ground reference (0 V) to the analog circuits of the CS8900.

Pin Types:

dI	=	Differential Input Pair	I	=	Input	G	=	Ground
dO	=	Differential Output Pair	O	=	Output	ts	=	Tri-State
B	=	Bi-Directional with Tri-State Output	P	=	Power	w	=	Internal Weak Pullup
OD	=	Open Drain Output						

Digital outputs are followed by drive in mA (Example: OD24 = Open Drain Output with 24 mA drive).

3.0 FUNCTIONAL DESCRIPTION

3.1 Overview

During normal operation, the CS8900 performs two basic functions: Ethernet packet transmission and reception. Before transmission or reception is possible, the CS8900 must be configured.

Configuration

The CS8900 must be configured for packet transmission and reception at power-up or reset. Various parameters must be written into its internal Configuration and Control registers such as Memory Base Address; Ethernet Physical Address; what frame types to receive; and which media interface to use. Configuration data can either be written to the CS8900 by the host (across the ISA bus), or loaded automatically from an external EEPROM. Operation can begin after configuration is complete.

Sections 3.1 and 3.3 describe the configuration process in detail. Section 4.4 provides a detailed description of the bits in the Configuration and Control Registers.

Packet Transmission

Packet transmission occurs in two phases. In the first phase, the host moves the Ethernet frame into the CS8900's buffer memory. The first phase begins with the host issuing a Transmit Command. This informs the CS8900 that a frame is to be transmitted and tells the chip when (i.e. after 5, 381, or 1021 bytes have been transferred, or after the full frame has been transferred to the CS8900) and how the frame should be sent (i.e. with or without CRC, with or without pad bits, etc.). The Host follows the Transmit Command with the Transmit Length, indicating how much buffer space is required. When buffer space is available, the host writes the Ethernet

frame into the CS8900's internal memory, either as a Memory or I/O space operation.

In the second phase of transmission, the CS8900 converts the frame into an Ethernet packet then transmits it onto the network. The second phase begins with the CS8900 transmitting the preamble and Start-of-Frame delimiter as soon as the proper number of bytes has been transferred into its transmit buffer (5, 381, 1021 bytes or full frame, depending on configuration). The preamble and Start-of Frame delimiter are followed by the Destination Address, Source Address, Length field and LLC data (all supplied by the host). If the frame is less than 64 bytes, including CRC, the CS8900 adds pad bits if configured to do so. Finally, the CS8900 appends the proper 32-bit CRC value.

Section 5.7 provides a detailed description of packet transmission.

Packet Reception

Like packet transmission, packet reception occurs in two phases. In the first phase, the CS8900 receives an Ethernet packet and stores it in on-chip memory. The first phase of packet reception begins with the receive frame passing through the analog front end and Manchester decoder where Manchester data is converted to NRZ data. Next, the preamble and Start-of-Frame delimiter are stripped off and the receive frame is sent through the address filter. If the frame's Destination Address matches the criteria programmed into the address filter, the packet is stored in the CS8900's internal memory. The CS8900 then checks the CRC, and depending on the configuration, informs the processor that a frame has been received.

In the second phase, the host transfers the receive frame across the ISA bus and into host memory. Receive frames can be transferred as Memory space operations, I/O space operations, or as DMA operations using host DMA. Also,

the CS8900 provides the capability to switch between Memory or I/O operation and DMA operation by using Auto-Switch DMA and StreamTransfer.

Sections 5.2 through 5.6 provide a detailed description of packet reception.

3.2   ISA Bus Interface

The CS8900 provides a direct interface to ISA buses running at clock rates from 8 to 11 MHz. Its on-chip bus drivers are capable of delivering 24 mA of drive current, allowing the CS8900 to drive the ISA bus directly, without added external "glue logic".

The CS8900 is optimized for 16-bit data transfers, operating in either Memory space, I/O space, or as a DMA slave.

Note that ISA-bus operation below 8 MHz should use the CS8900's Receive DMA mode to minimize missed frames. See Section 5.4 for a description of Receive DMA operation.

Memory Mode Operation

When configured for Memory Mode operation, the CS8900's internal RAM is mapped into a contiguous 4-Kbyte block of host memory, providing the host with direct access to the CS8900's internal registers and frame buffers. The host initiates Read operations by driving the $\overline{\text{MEMR}}$ pin low and Write operations by driving the $\overline{\text{MEMW}}$ pin low.

For additional information about Memory Mode, see Section 4.9.

I/O Mode Operation

When configured for I/O Mode operation, the CS8900 is accessed through eight, 16-bit I/O ports that are mapped into sixteen contiguous

I/O locations in the host system's I/O space. I/O Mode is the default configuration for the CS8900 and is always enabled.

For an I/O Read or Write operation, the AEN pin must be low, and the 16-bit I/O address on the ISA System Address bus (SA0 - SA15) must match the address space of the CS8900. For a Read, $\overline{\text{IOR}}$ must be low, and for a Write, $\overline{\text{IOW}}$ must be low.

For additional information about I/O Mode, see Section 4.10.

Interrupt Request Signals

The CS8900 has four interrupt request output pins that can be connected directly to any four of the ISA bus Interrupt Request signals. Only one interrupt output is used at a time. It is selected during initialization by writing the interrupt number (0 to 3) into PacketPage Memory base + 0022h. Unused interrupt request pins are placed in a high-impedance state. The selected interrupt request pin goes high when an enabled interrupt is triggered. The pin goes low after the Interrupt Status Queue (ISQ) is read as all 0's (see Section 5.1 for a description of the ISQ).

CS8900 Interrupt Request Pin	ISA Bus Interrupt	PacketPage base + 0022h
INTRQ3 (Pin 35)	IRQ5	0003h
INTRQ0 (Pin 32)	IRQ10	0000h
INTRQ1 (Pin 31)	IRQ11	0001h
INTRQ2 (Pin 30)	IRQ12	0002h

Table 3.1. Interrupt Assignments

Table 3.1 presents one possible way of connecting the interrupt request pins to the ISA bus that utilizes commonly available interrupts and facilitates board layout.

DMA Signals

The CS8900 interfaces directly to the host DMA controller to provide DMA transfers of receive frames from CS8900 memory to host memory. The CS8900 has three pairs of DMA pins that can be connected directly to the three 16-bit DMA channels of the ISA bus. Only one DMA channel is used at a time. It is selected during initialization by writing the number of the desired channel (0, 1 or 2) into PacketPage Memory base + 0024h. Unused DMA pins are placed in a high-impedance state. The selected DMA request pin goes high when the CS8900 has received frames to transfer to the host memory via DMA. If the DMABurst bit (register 17, BusCTL, Bit B) is set, the pin goes low after the DMA operation is complete. If the DMABurst bit is clear, the pin goes low 32 µs after the start of a DMA transfer.

CS8900 DMA Signal (Pin #)	ISA DMA Signal	PacketPage base + 0024h
DMARQ0 (Pin 15)	DRQ5	0000h
DMACK0 (Pin 16)	DACK5	
DMARQ1 (Pin 13)	DRQ6	0001h
DMACK1 (Pin 14)	DACK6	
DMARQ2 (Pin 11)	DRQ7	0002h
DMACK2 (Pin 12)	DACK7	

Table 3.2. DMA Assignments

The DMA pin pairs are arranged on the CS8900 to facilitate board layout. Crystal recommends the configuration in Table 3.2 when connecting these pins to the ISA bus.

For a description of DMA mode, see Section 5.4.

3.3   Reset and Initialization

3.3.1   Reset

Seven different conditions cause the CS8900 to reset its internal registers and circuits.

External Reset, or ISA Reset: There is a chip-wide reset whenever the RESET pin is high for at least 400 ns. During a chip-wide reset, all circuitry and registers in the CS8900 are reset.

Power-Up Reset: When power is applied, the CS8900 maintains reset until the voltage at the supply pins reaches approximately 2.5 V. The CS8900 comes out of reset once Vcc is greater than approximately 2.5 V *and* the crystal oscillator has stabilized.

Power-Down Reset: If the supply voltage drops below approximately 2.5 V, there is a chip-wide reset. The CS8900 comes out of reset once the power supply returns to a level greater than approximately 2.5 V *and* the crystal oscillator has stabilized.

EEPROM Reset: There is a chip-wide reset if an EEPROM checksum error is detected (see Section 3.1).

Software Initiated Reset: There is a chip-wide reset whenever the RESET bit (Register 15, SelfCTL, Bit 6) is set.

Hardware (HW) Standby or Suspend: The CS8900 goes though a chip-wide reset whenever it enters or exits either HW Standby mode or HW Suspend mode (see Section 3.7 for more information about HW Standby and Suspend).

Software (SW) Suspend: Whenever the CS8900 enters SW Suspend mode, all registers and circuits are reset except for the ISA I/O Base Address register (located at PacketPage base + 0020h) and the SelfCTL register (Register 15).

Upon exit, there is a chip-wide reset (see Section 3.7 for more information about SW Suspend).

3.3.2 Allowing Time for Reset Operation

After a reset, the CS8900 goes through a self configuration. This includes calibrating on-chip analog circuitry, and reading EEPROM for validity and configuration. Time required for the reset calibration is typically 10 ms. Software drivers should not access registers internal to the CS8900 during this time. When calibration is done, bit INITD in the Self Status Register (register 16) is set indicating that initialization is complete, and the SIBUSY bit in the same register is cleared indicating the EEPROM is no longer being read or programmed.

3.3.3 Bus Reset Considerations

The CS8900 reads 3000h from IObase+0Ah after the reset, until the software writes a non-zero value at IObase+0Ah. The 3000h address can be used as part of the CS8900 signature when the system scans for the CS8900. See Section 4.10, I/O Space Operation.

After a reset, the ISA bus outputs INTRx and DMARQx are tri-stated, thus avoiding any interrupt or DMA channel conflicts on the ISA bus at power-up time.

Initialization

After each reset (except EEPROM Reset), the CS8900 checks the sense of the EEDataIn pin to see if an external EEPROM is present. If EEDI is high, an EEPROM is present and the CS8900 automatically loads the configuration data stored in the EEPROM into its internal registers (see next section). If EEDI is low, an EEPROM is not present and the CS8900 comes out of reset with the default configuration shown in Table 3.3.

A low-cost serial EEPROM can be used to store configuration information that is automatically

PacketPage Address	Register Contents	Register Description
0020h	0300h	I/O Base Address*
0022h	XXXX XXXX XXXX X100	Interrupt Number
0024h	XXXX XXXX XXXX XX11	DMA Channel
0026h	0000h	DMA Start of Frame Offset
0028h	X000h	DMA Frame Count
002Ah	0000h	DMA Byte Count
002Ch	XXX0 0000h	Memory Base Address
0030h	XXX0 0000h	Boot PROM Base Address
0034h	XXX0 0000h	Boot PROM Address Mask
0102h	0003h	Register 3 - RxCFG
0104h	0005h	Register 5 - RxCTL
0106h	0007h	Register 7 - TxCFG
0108h	0009h	Register 9 - TxCMD
010Ah	000Bh	Register B - BufCFG
010Ch	Undefined	Reserved
010Eh	Undefined	Reserved
0110h	Undefined	Reserved
0112h	0013h	Register 13 - LineCTL
0114h	0015h	Register 15 - SelfCTL
0116h	0017h	Register 17 - BusCTL
0118h	0019h	Register 19 - TestCTL

* I/O base address is unaffected by Software Suspend mode.

Table 3.3. Default Configuration

EEPROM Type	Size (16-bit words)
'C46 (non-sequential)	64
'CS46 (sequential)	64
'C56 (non-sequential)	128
'CS56 (sequential)	128
'C66 (non-sequential)	256
'CS66 (sequential)	256

Table 3.4. Supported EEPROM Types

loaded into the CS8900 after each reset (except EEPROM reset). The use of an EEPROM is optional and is not required for all applications (e.g. motherboard designs).

The CS8900 operates with any of six standard EEPROM's shown in Table 3.4.

3.4   Configuration with EEPROM

EEPROM Interface

The interface to the EEPROM consists of the four signals shown in Table 3.5.

CS8900 Pin (Pin #)	CS8900 Function	EEPROM Pin
EECS (Pin 3)	EEPROM Chip Select	Chip Select
EESK (Pin 4)	1 MHz EEPROM Serial Clock output	Clock
EEDO (Pin 5)	EEPROM Data Out (data to EEPROM)	Data In
EEDI (Pin 6)	EEPROM Data In (data from EEPROM)	Data Out

Table 3.5   EEPROM Interface

EEPROM Memory Organization

If an EEPROM is used to store initial configuration information for the CS8900, the EEPROM is organized in one or more blocks of 16-bit words. The first block in EEPROM, referred to as the Configuration Block, is used to configure the CS8900 after reset. An example of a typical Configuration Block is shown in Table 3.6. Additional blocks containing user data may be stored in the EEPROM. However, the Configuration Block must always start at address 00h and be stored in contiguous memory locations.

Reset Configuration Block

The first block in EEPROM, referred to as the Reset Configuration Block, is used to automatically program the CS8900 with an initial configuration after a reset. Additional user data may also be stored in the EEPROM if space is available. The additional data are stored as 16-bit words and can occupy any EEPROM address space beginning immediately after the end of the Reset Configuration Block up to address 7Fh, depending on EEPROM size. This additional data can only be accessed through software control (refer to Section 3.5 for more information on accessing the EEPROM). Address space 80h to AFh is reserved.

Reset Configuration Block Structure: The Reset Configuration Block is a block of contiguous 16-bit words starting at EEPROM address 00h. It can be divided into three logical sections: a header, one of more groups of configuration data words, and a checksum value. All of the words in the Reset Configuration Block are read sequentially by the CS8900 after each reset, starting with the header and ending with the checksum. Each group of configuration data is used to program a PacketPage register (or set of PacketPage registers in some cases) with an initial non-default value.

Reset Configuration Block Header: The header (first word of the block located at EEPROM address 00h) specifies the type of EEPROM used, whether or not a Reset Configuration block is present, and if so, how many bytes of configuration data are stored in the Reset Configuration Block.

Determining the EEPROM Type: The LSB of the high byte of the header indicates the type of EEPROM attached: sequential or non-sequential. An LSB of 0 (XXXX-XXX0) indicates a sequential EEPROM. An LSB of 1 (XXXX-XXX1) indicates a non-sequential EEPROM. The CS8900 works equally well with

either type of EEPROM. The CS8900 will automatically generate sequential addresses while reading the Reset Configuration Block if a non-sequential EEPROM is used.

Checking EEPROM for presence of Reset Configuration Block: The read-out of either a binary 101X-XXX0 or 101X-XXX1 (X = do not care)

from the high byte of the header indicates the presence of configuration data. Any other read-out value terminates initialization from the EEPROM. If an EEPROM is attached but not used for configuration, Crystal recommends that the high byte of the first word be programmed with 00h in order to ensure that the CS8900 will

Word Address	Value	Description
FIRST WORD in DATA BLOCK		
00h	A120h	Configuration Block Header. The high byte, A1h, indicates a 'C46 EEPROM is attached. The Link Byte, 20h, indicates the number of bytes to be used in this block of configuration data.
FIRST GROUP of WORDS		
01h	2020h	Group Header for first group of words. Three words to be loaded, beginning at 0020h in PacketPage memory.
02h	0300h	I/O Base Address
03h	0003h	Interrupt Number
04h	0001h	DMA Channel Number
SECOND GROUP of WORDS		
05h	502Ch	Group Header for second group of words. Six words to be loaded, beginning at 002Ch in PacketPage memory.
06h	E000h	Memory Base Address - low word
07h	000Fh	Memory Base Address - high word
08h	0000h	Boot PROM Base Address - low word
09h	000Dh	Boot PROM Base Address - high word
0Ah	C000h	Boot PROM Address Mask - low word
0Bh	000Fh	Boot PROM Address Mask high word
THIRD GROUP of WORDS		
0Ch	2158h	Group Header for third group of words. Three words to be loaded, beginning at 0158h in PacketPage memory.
0Dh	0010h	Individual Address - IA bits [39-32], bits [47-40]
0Eh	0000h	Individual Address - IA bits [23-16], bits [31-24]
0Fh	0000h	Individual Address - IA bits [7-0], bits [15-8]
CHECKSUM Value		
10h	2800h	The high byte, 28h, is the Checksum Value. In this example, the checksum includes word addresses 00h through 0Fh. The hexadecimal sum of the bytes is D7h, resulting in a 2's complement of 28h. The low byte, 00h, provides a pad to the word boundary.

*FFFFh is a special code indicating that there are no more words in the EEPROM.

Table 3.6. EEPROM Configuration Block Example

not attempt to read configuration data from the EEPROM.

Determining Number of Bytes in the Reset Configuration Block: The low byte of the Reset Configuration Block header is known as the link byte. The value of the Link Byte represents the number of bytes of configuration data in the Reset Configuration Block. The two bytes used for the header are excluded when calculating the Link Byte value.

For example, a Reset Configuration Block header of A104h indicates a non-sequential EEPROM programmed with a Reset Configuration Block containing 4 bytes of configuration data. This Reset Configuration Block occupies 6 bytes (3 words) of EEPROM space (2 bytes for the header and 4 bytes of configuration data).

Groups of Configuration Data

Configuration data are arranged as groups of words. Each group contains one or more words of data that are to be loaded into PacketPage registers. The first word of each group is referred to as the Group Header. The Group Header indicates the number of words in the group and the address of the PacketPage register into which the first data word in the group is to be loaded. Any remaining words in the group are stored in successive PacketPage registers.

Group Header: Bits F through C of the Group Header specify the number of words in each group that are to be transferred to PacketPage registers (see Figure 3.1). This value is two less than the total number of words in the group, including the Group Header. For example, if bits F through C contain 0001, there are three words in the group (a Group Header and two words of configuration data).

Bits 8 through 0 of the Group Header specify a 9-bit PacketPage Address. This address defines the PacketPage register that will be loaded with the first word of configuration data from the group. Bits B though 9 of the Group Header are forced to 0, restricting the destination address range to the first 512 bytes of PacketPage memory. Figure 3.1 shows the format of the Group header.

Reset Configuration Block Checksum

A checksum is stored in the high byte position of the word immediately following the last group of data in the Reset Configuration Block. (The EEPROM address of the checksum value can be determined by dividing the value stored in the Link Byte by two). The checksum value is the 2's complement of the 8-bit sum (any carry out of eighth bit is ignored) of all the bytes in the Reset Configuration Block, excluding the checksum byte. This sum includes the Reset Configuration Block header at address 00h. Since the checksum value is calculated as the 2 s complement of the sum of all the preceding bytes in the in the Reset Configuration Block, a total of 0 should result when the checksum value is added to the sum of the previous bytes.

EEPROM Example

Table 3.6 shows an example of a Reset Configuration Block stored in a C46 EEPROM. Note that little-endian word ordering is used, i.e., the
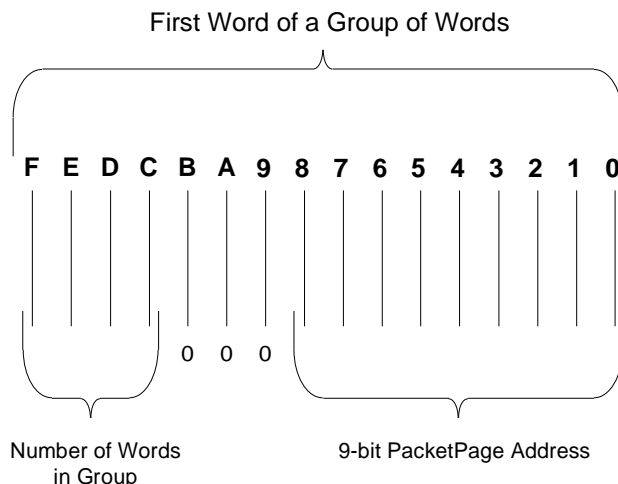
First Word of a Group of Words

F E D C B A 9 8 7 6 5 4 3 2 1 0

0 0 0

Number of Words in Group   9-bit PacketPage Address

Figure 3.1. Group Header

least significant word of a multi-word datum is located at the lowest address.

EEPROM Read-out

If the EEDI pin is asserted high at the end of reset, the CS8900 reads the first word of EEPROM data by:

1. Asserting EECS

2. Clocking out a Read-Register-00h command on EEDO (EESK provides a 1MHz serial clock signal)

3. Clocking the data in on EEDI.

If the EEDI pin is low at the end of the reset signal, the CS8900 does not perform an EEPROM read-out (uses its default configuration).

Determining EEPROM Size: The CS8900 determines the size of the EEPROM by checking the sense of EEDI on the tenth rising edge of EESK. If EEDI is low, the EEPROM is a 'C46 or 'CS46. If EEDI is high, the EEPROM is a 'C56, 'CS56, 'C66, or 'CS66.

Loading Configuration Data: The CS8900 reads in the first word from the EEPROM to determine if configuration data is contained in the EEPROM. If configuration data is not stored in the EEPROM, the CS8900 terminates initialization from EEPROM and operates using its default configuration (See Table 3.3). If configuration data is stored in EEPROM, the CS8900 automatically loads all configuration data stored in the Reset Configuration Block into its internal PacketPage registers.

EEPROM Read-out Completion

Once all the configuration data are transferred to the appropriate PacketPage registers, the CS8900 performs a checksum calculation to verify the

Reset Configuration Blocks data are valid. If the resulting total is 0, the read-out is considered valid. Otherwise, the CS8900 initiates a partial reset to restore the default configuration.

If the read-out is valid, the EEPROMOK bit (Register 16, SelfST, bit A) is set. EEPROMOK is cleared if a checksum error is detected. In this case, the CS8900 performs a partial reset and is restored to its default. Once initialization is complete (configuration loaded from EEPROM or reset to default configuration) the INITD bit is set (Register 16, SelfST, bit 7).

3.5 Programming the EEPROM

After initialization, the host can access the EEPROM through the CS8900 by writing one of seven commands to the EEPROM Command register (PacketPage base + 0040h). Figure 3.2 shows the format of the EEPROM Command register.

EEPROM Commands

The seven commands used to access the EEPROM are: Read, Write, Erase, Erase/Write Enable, Erase/Write Disable, Erase-All, and Write-All. They are described in Table 3.7.

EEPROM Command Execution

During the execution of a command, the two Opcode bits, followed by six bits of address (for a 'C46 or 'CS46) or eighth bits of address (for a 'C56, 'CS56, 'C66 or 'CS66), are shifted out of the CS8900, into the EEPROM. If the command is a Write, the data in the EEPROM Data register (PacketPage base + 0042h) follows. If the command is a Read, the data in the specified EEPROM location is written into the EEPROM Data register. If the command is an Erase or Erase-All, no data is transferred to or from the EEPROM Data register. Before issuing any command, the host must wait for the SI-BUSY bit

(Register 16, SelfST, bit 8) to clear. After each command has been issued, the host must wait again for SI-BUSY to clear.
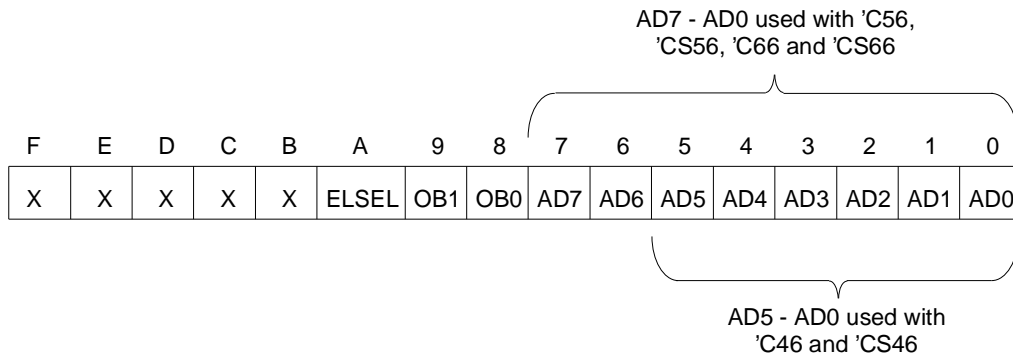
Enabling Access to the EEPROM

The Erase/Write Enable command provides protection from accidental writes to the EEPROM. The host must write an Erase/Write Enable command before it attempts to write to or erase any

EEPROM memory location. Once the host has finished altering the contents of the EEPROM, it must write an Erase/Write Disable command to prevent unwanted modification of the EEPROM.

Writing and Erasing the EEPROM

To write data to the EEPROM, the host must execute the following series of commands:

AD7 - AD0 used with 'C56,
'CS56, 'C66 and 'CS66

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	ELSEL	OB1	OB0	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

AD5 - AD0 used with
'C46 and 'CS46

Bit	Name	Description
[F:B]		Reserved
[A]	ELSEL	**External Logic Select:** When clear, the EECS pin is used to select the EEPROM. When set, the ELCS pin is used to select the external LA decode circuit.
[9:8]	OB1, OB0	**Opcode:** Indicates what command is being executed (see next section).
[7:0]	AD7 to AD0	**EEPROM Address:** Address of EEPROM word being accessed.

Figure 3.2. EEPROM Command Register Format

Command	Opcode (bits 9,8)	EEPROM Address (bits 7 to 0)	Data	EEPROM Type	Execution Time
Read Register	1,0	word address	yes	all	25 µs
Write Register	0,1	word address	yes	all	10 ms
Erase Register	1,1	word address	no	all	10 ms
Erase/Write Enable	0,0	XX11-XXXX	no	'CS46, 'C46	9 µs
		11XX-XXXX	no	'CS56, 'C56, 'CS66, 'C66	9 µs
Erase/Write Disable	0,0	XX00-XXXX	no	'CS46, 'C46	9 µs
	0,0	00XX-XXXX	no	'CS56, 'C56, 'CS66, 'C66	9 µs
Erase-All Registers	0,0	XX10-XXXX	no	'CS46, 'C46	10 ms
	0,0	10XX-XXXX	no	'CS56, 'C56, 'CS66, 'C66	10 ms
Write-All Registers	0,0	XX01-XXXX	yes	'CS46, 'C46	10 ms
	0,0	01XX-XXXX	yes	'CS56, 'C56, 'CS66, 'C66	10 ms

Table 3.7. EEPROM Commands

1. Issue an Erase/Write Enable command.

2. Load the data into the EEPROM Data register.

3. Issue a Write command.

4. Issue an Erase/Write Disable command.

During the Erase command, the CS8900 writes FFh to the specified EEPROM location. During the Erase-All command, the CS8900 writes FFh to all locations.

3.6 Boot PROM Operation

The CS8900 supports an optional Boot PROM used to store code for remote booting from a network server.

Accessing the Boot PROM

To retrieve the data stored in the Boot PROM, the host issues a Read command to the Boot PROM as a Memory space access. The CS8900 decodes the command and drives the CSOUT pin low, causing the data stored in the Boot PROM to be shifted into the bus transceiver. The bus transceiver then drives the data out onto the ISA bus.
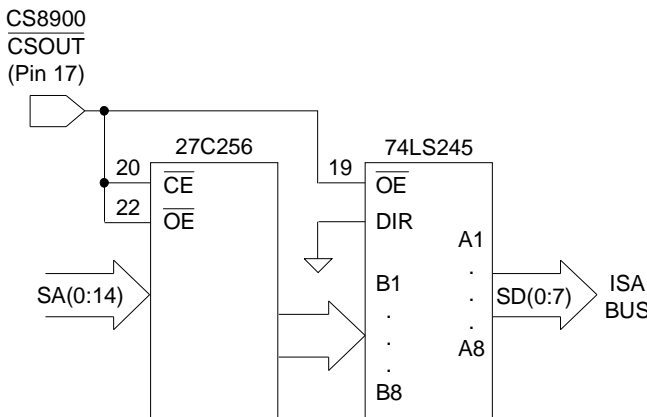

Figure 3.3. Boot PROM Connection Diagram

Configuring the CS8900 for Boot PROM Operation

Figure 3.3 show how the CS8900 should be connected to the Boot PROM and '245 driver. To configure the CS8900's internal registers for Boot PROM operation, the Boot PROM Base Address must be loaded into the Boot PROM Base Address register (PacketPage base + 0030h) and the Boot PROM Address Mask must be loaded into the BootPROM Address Mask register (PacketPage base + 0034h). The Boot PROM Base Address provides the starting location in host memory where the Boot PROM is mapped. The Boot PROM Address Mask indicates the size of the attached Boot PROM and is limited to 4-Kbyte increments. The lower 12 bits of the Address Mask are ignored and should be 000h.

In the EEPROM example shown in Table 3.6, the Boot PROM starting address is D0000h and the Address Mask is FC000h. This configuration describes a 16-Kbyte (128 Kbit) PROM mapped into host memory from D0000h to D3FFFh.

3.7 Low-Power Modes

For power-sensitive applications, the CS8900 supports three low-power modes: Hardware Standby, Hardware Suspend, and Software Suspend. All three low-power modes are controlled through the SelfCTL register (Register 15). See also Section 5.85.8.

An internal reset occurs when the CS8900 comes out of any suspend or standby mode. After a reset (internal or external), the CS8900 goes though a self configuration. This includes calibrating on-chip analog circuitry, and reading EEPROM for validity and configuration. When the calibration is done, bit InitD in Register 16 (Self Status register) is set indicating that initialization is complete, and the SIBusy bit in the same register is cleared (indicating that the EEPROM is no longer being read or pro-

grammed. Time required for the reset calibration is typically 10 ms. Software drivers should not access registers internal to CS8900 during this time.

Hardware Standby

Hardware (HW) Standby is designed for use in systems, such as portable PC's, that may be temporarily disconnected from the 10BASE-T cable. It allows the system to conserve power while the LAN is not in use, and then automatically restore Ethernet operation once the cable is reconnected.

In HW Standby mode, all analog and digital circuitry in the CS8900 is turned off, except for the 10BASE-T receiver which remains active to listen for link activity. If link activity is detected, the $\overline{\text{LANLED}}$ pin is driven low, providing an indication to the host that the network connection is active. The host can then activate the CS8900 by de-asserting the $\overline{\text{SLEEP}}$ pin. During this mode, all ISA bus accesses are ignored.

To enter HW Standby mode, the $\overline{\text{SLEEP}}$ pin must be low *and* the SLEEPE bit (Register 15, SelfCTL, Bit 9) *and* the HWstandbyE bit (Register 15, SelfCTL, Bit A) must be set. When the CS8900 enters HW Standby, all registers and circuits are reset except for the SelfCTL register. Upon exit from HW Standby, the CS8900 performs a complete reset, and then goes through normal initialization.

Hardware Suspend

During Hardware Suspend mode, the CS8900 uses the least amount of current of the three low-power modes. All internal circuits are turned off and the CS8900's core is electronically isolated from the rest of the system. Accesses from the ISA bus and Ethernet activity are both ignored.

HW Suspend mode is entered by driving the $\overline{\text{SLEEP}}$ pin low and setting the SLEEPE bit (Register 15, SelfCTL, bit 9) while the HWstandbyE bit (Register 15, SelfCTL, bit A) is clear. To exit from this mode, the $\overline{\text{SLEEP}}$ pin must be driven high. Upon exit, the CS8900 per-

CS8900 Configuration					CS8900 Operation
$\overline{\text{SLEEP}}$ (Pin 77)	HWstandbyE (SelfCTL, Bit A)	SLEEPE (SelfCTL, Bit 9)	SWSuspend (SelfCTL, Bit 8)	Link Activity	
Low	1	1	N/A	Not Present	HW Standby mode: 10BASE-T receiver listens for link activity
Low	1	1	N/A	Present	HW Standby mode: $\overline{\text{LANLED}}$ low
Low	0	1	N/A	N/A	HW Suspend mode
Low to High	N/A	1	0	N/A	CS8900 resets and goes through Initialization
High	N/A	N/A	0	N/A	Not in low-power mode
High	N/A	N/A	1	N/A	SW Suspend mode
Low	N/A	0	1	N/A	SW Suspend mode
Low	N/A	0	0	N/A	Not in low-power mode

Note: Both HW Standby and HW Suspend take precedence over SW Suspend.

Table 3.8. Low-Power Mode Operation

forms a complete reset, and then goes through a normal initialization procedure.

Software Suspend

Software (SW) Suspend mode can be used to conserve power in applications, like adapter cards, that do not have power management circuitry available. During this mode, all internal circuits are shut off except the I/O Base Address register (PacketPage base + 0020h) and the SelfCTR register (Register 15).

To enter SW Suspend mode, the host must set the SWSuspend bit (Register 15, SelfCTL, bit 8). To exit SW Suspend, the host must write to the CS8900's assigned I/O space (the Write is only used to wake the CS8900, the Write itself is ignored). Upon exit, the CS8900 performs a complete reset, and then goes through a normal initialization procedure.

Any hardware reset takes the chip out of any sleep mode.

Table 3.8 summarizes the operation of the three low-power modes.

3.8   LED Outputs

The CS8900 provides three output pins that can be used to control LEDs or external logic.

HC0E (Bit C)	HCB0 (Bit E)	Pin Function
0	N/A	Pin configured as LINKLED: Output is low when valid 10BASE-T link pulses are detected. Output is high if valid link pulses are not detected
1	0	Pin configured as HC0: Output is high
1	1	Pin configured as HC0: Output is low

Table 3.9. LINKLED/HC0 Pin Operation

LANLED: LANLED goes low whenever the CS8900 transmits or receives a frame, or when it detects a collision. LANLED remains low until there has been no activity for 6 ms (i.e. each transmission, reception, or collision produces a pulse lasting a minimum of 6 ms).

LINKLED or HC0: LINKLED or HC0 can be controlled by either the CS8900 or the host. When controlled by the CS8900, LINKLED is low whenever the CS8900 receives valid 10BASE-T link pulses. To configure this pin for CS8900 control, the HC0E bit (Register 15, SelfCTL, Bit C) must be clear. When controlled by the host, LINKLED is low whenever the HCB0 bit (Register 15, SelfCTL, Bit E) is set. To configure it for host control, the HC0E bit must be set. Table 3.9 summarizes this operation.

HC1E (Bit D)	HCB1 (Bit F)	Pin Function
0	N/A	Pin configured as BSTATUS: Output is low when a receive frame begins transfer across the ISA bus. Output is high otherwise.
1	0	Pin configured as HC1: Output is high
1	1	Pin configured as HC1: Output is low

Table 3.10. BSTATUS/HC1 Pin Operation

BSTATUS or HC1: BSTATUS or HC1 can be controlled by either the CS8900 or the host. When controlled by the CS8900, BSTATUS is low whenever the host reads the RxEvent register (PacketPage base + 0124h), signaling the transfer of a receive frame across the ISA bus. To configure this pin for CS8900 control, the HC1E bit (Register 15, SelfCTL, Bit D) must be clear. When controlled by the host, BSTATUS is low whenever the HCB1 bit (Register 15, SelfCTL, Bit F) is set. To configure it for host control, HC1E must be set. Table 3.10 summarizes this operation.

LED Connection

Each LED output is capable of sinking 10 mA to drive an LED directly through a series resistor. The output voltage of each pin is less than 0.4 V when the pin is low. Figure 3.4 shows a typical LED circuit.
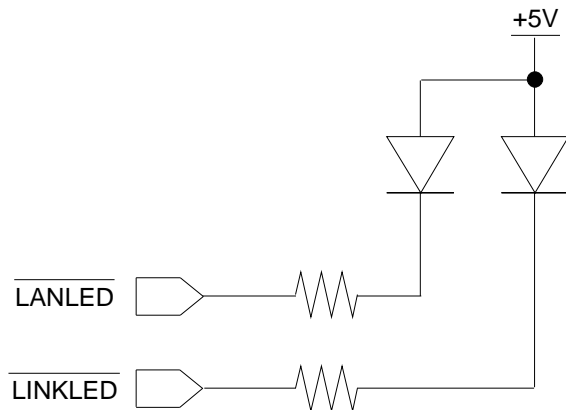

Figure 3.4. LED Connection Diagram

3.9 Media Access Control

Overview

The CS8900's Ethernet Media Access Control (MAC) engine is fully compliant with the IEEE 802.3 Ethernet standard (ISO/IEC 8802-3, 1993). It handles all aspects of Ethernet frame transmission and reception, including: collision detection, preamble generation and detection, and CRC generation and test. Programmable MAC features include automatic retransmission on collision, and padding of transmitted frames.
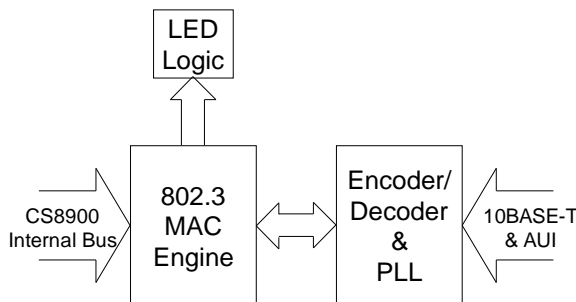
Figure 3.5. MAC Interface

Figure 3.5 shows how the MAC engine interfaces to other CS8900 functions. On the host side, it interfaces to the CS8900's internal data/address/control bus. On the network side, it interfaces to the internal Manchester encoder/decoder (ENDEC). The primary functions of the MAC are: frame encapsulation and decapsulation; error detection and handling; and, media access management.

Frame Encapsulation and Decapsulation

The CS8900's MAC engine automatically assembles transmit packets and disassembles receive packets. It also determines if transmit and receive frames are of legal minimum size.

Transmission: Once the proper number of bytes have been transferred to the CS8900's memory (either 5, 381, 1021 bytes, or full frame), and providing that access to the network is permitted, the MAC automatically transmits the 7-byte preamble (1010101b…), followed by the Start-of-Frame Delimiter (SFD, 10101011b), and then the serialized frame data. It then transmits the Frame Check Sequence (FCS). The data after the SFD and before the FCS (Destination Address, Source Address, Length, and data field) is supplied by the host. FCS generation by the CS8900 may be disabled by setting the InhibitCRC bit (Register 9, TxCMD, bit C).

Figure 3.6 shows the Ethernet frame format.

Reception: The MAC receives the incoming packet as a serial stream of NRZ data from the Manchester encoded/decoder. It begins by checking for the SFD. Once the SFD is detected, the MAC assumes all subsequent bits are frame data. It reads the DA and compares it to the criteria programmed into the address filter (see Section 5.3 for a description of Address Filtering). If the DA passes the address filter, the frame is loaded into the CS8900's memory. If the BufferCRC bit (Register 3, RxCFG, bit B) is set, the received FCS is also loaded into memory. Once the en-

tire packet has been received, the MAC validates the FCS. If an error is detected, the CRCerror bit (Register 4, RxEvent, Bit C) is set.

Enforcing Minimum Frame Size: The MAC provides minimum frame size enforcement of both transmit and receive packets. When the TxPadDis bit (Register 9, TxCMD, Bit D) is clear, transmit frames will be padded with additional bits to ensure that the receiving station receives a legal frame (64 bytes, including CRC). When TxPadDis is set, the CS8900 will not add pad bits and will transmit frames less that 64 bytes. If a frame is received that is less than 64 bytes (including CRC), the Runt bit (Register 4, RxEvent, Bit D) will be set indicating the arrival of an illegal frame.

Transmit Error Detection and Handling

The MAC engine monitors Ethernet activity and reports and recovers from a number of error conditions. For transmission, the MAC reports the following errors in the TxEvent register (Register 8) and BufEvent register (Register C):

Loss of Carrier: Whenever the CS8900 is transmitting on the AUI port, it expects to see its own transmission "looped back" to its receiver. If it is unable to monitor its transmission after the end of the preamble, the MAC reports a loss-of-carrier error by setting the Loss-of-CRS bit (Register 8, TxEvent, Bit 6). If the Loss-of-

CRSiE bit (Register 7, TxCFG, Bit 6) is set, the host will be interrupted.

SQE Error: After the end of transmission on the AUI port, the MAC expects to see a collision within 64 bit times. If no collision is detected, the SQEerror bit (Register 8, TxEvent, Bit 7) is set. If the SQEerroriE bit is set (Register 7, TxCFG, Bit 7), the host is interrupted. An SQE error may indicate a fault on the AUI cable or a faulty transceiver (it is assumed that the attached transceiver supports this function).

Out-of-Window (Late) Collision: If a collision is detected after the first 512 bits have been transmitted, the MAC reports a late collision by setting the Out-of-window bit (Register 8, TxEvent, Bit 9). The MAC then forces a bad CRC and terminates the transmission. If the Out-of-windowiE bit (Register 7, TxCFG, Bit 9) is set, the host is interrupted. A late collision may indicate an illegal network configuration.

Jabber Error: If a transmission continues longer than about 26 ms, the MAC disables the transmitter and sets the Jabber bit (Register 8, TxEvent, Bit A). The output of the transmitter returns to idle and remains there until the host issues a new Transmit Command. If the JabberiE bit (Register 7, TxCFG, Bit A) is set, the host is interrupted. A Jabber condition indicates that there may be something wrong with the CS8900 transmit function. To prevent possible network
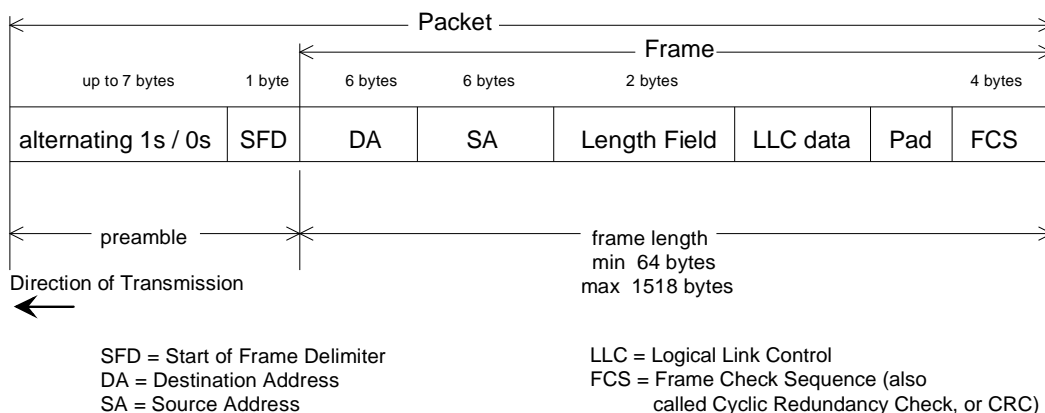

Figure 3.6. Ethernet Frame Format

faults, the host should clear the transmit buffer. Possible options include:

Reset the chip with either software or hardware reset (see Section 3.3).

Issue a Force Transmit Command by setting the Force bit (Register 9, TxCMD, bit 8).

Issue a Transmit Command with the TxLength field set to zero.

Transmit Collision: The MAC counts the number of times an individual packet must be re-transmitted due to network collisions. The collision count is stored in bits B through E of the TxEvent register (Register 8). If the packet collides 16 times, transmission of that packet is terminated and the 16coll bit (Register 8, TxEvent, Bit F) is set. If the 16colliE bit (Register 7, TxCFG, Bit F) is set, the host will be interrupted on the 16th collision. A running count of transmit collisions is recorded in the TxCOL register.

Transmit Underrun: If the CS8900 starts transmission of a packet but runs out of data before reaching the end of frame, the TxUnderrun bit (Register C, BufEvent, Bit 9) is set. The MAC then forces a bad CRC and terminates the transmission. If the TxUnderruniE bit (Register B, BufCFG, Bit 9) is set, the host is interrupted.

Receive Error Detection and Handling

The following receive errors are reported in the RxEvent register (Register 4):

CRC Error: If a frame is received with a bad CRC, the CRCerror bit (Register 4, RxEvent, Bit C) is set. If the CRCerrorA bit (Register 5, RxCTL, Bit C) is set, the frame will be buffered by CS8900. If the CRCerroriE bit (Register 3, RxCFG. Bit C) is set, the host is interrupted.

Runt Frame: If a frame is received that is shorter than 64 bytes, the Runt bit (Register 4, RxEvent, Bit D) is set. If the RuntA bit (Register 5, RxCTL, Bit D) is set, the frame will still be buffered by CS8900. If the RuntiE bit (Register 3, RxCFG. Bit D) is set, the host is interrupted.

Extra Data: If a frame is received that is longer than 1518 bytes, the Extradata bit (Register 4, RxEvent, Bit E) is set. If the ExtradataA bit (Register 5, RxCTL, Bit E) is set, the first 1518 bytes of the frame will still be buffered by CS8900. If the ExtradataiE bit (Register 3, RxCFG. Bit E) is set, the host is interrupted.

Dribble Bits and Alignment Error: Under normal operating conditions, the MAC may detect up to 7 additional bits after the last full byte of a receive packet. These bits, known as dribble bits, are ignored. If dribble bits are detected, the Dribblebit bit (Register 4, RxEvent, Bit 7) is set. If both the Dribblebit bit and CRCerror bit (Register 4, RxEvent, Bit C) are set at the same time, an alignment error has occurred.

Media Access Management

The Ethernet network topology is a single shared medium with several attached stations. The Ethernet protocol is designed to allow each station equal access to the network at any given time. Any node can attempt to gain access to the network by first completing a deferral process (described below) after the last network activity, and then transmitting a packet that will be received by all other stations. If two nodes transmit simultaneously, a collision occurs and the colliding packets are corrupted. Two primary tasks of the MAC are to avoid network collisions, and then recover from them when they occur. In addition, when the CS8900 is using the AUI, the MAC must support the SQE Test function described in section 7.2.4.6 of the Ethernet standard.

Collision Avoidance: The MAC continually monitors network traffic by checking for the presence of carrier activity (carrier activity is indicated by the assertion of the internal Carrier Sense signal generated by the ENDEC). If carrier activity is detected, the network is assumed busy and the MAC must wait until the current packet is finished before attempting transmission. The CS8900 supports two schemes for determining when to initiate transmission: Two-Part Deferral, and Simple Deferral. Selection of the deferral scheme is determined by the 2-part-DefDis bit (Register 13, LineCTL, Bit D). If the 2-partDefDis bit is clear, the MAC uses a two-part deferral process defined in section 4.2.3.2.1 of the Ethernet standard (ISO/IEC 8802-3, 1993). If the 2-partDefDis bit is set, the MAC uses a simplified deferral scheme. Both schemes are described below:

Two-Part Deferral: In the two-part deferral process, the 9.6 µs Inter Packet Gap (IPG) timer is started whenever the internal Carrier Sense signal is de-asserted. If activity is detected during the first 6.4 µs of the IPG timer, the timer is reset and then restarted once the activity has stopped. If there is no activity during the first 6.4 µs of the IPG timer, the IPG timer is allowed to time out (even if network activity is detected during the final 3.2 µs). The MAC then begins transmission if a transmit packet is ready and if it is not in Backoff (Backoff is described later in this section). If no transmit packet is pending, the MAC continues to monitor the network. If activity is detected before a transmit frame is ready, the MAC defers to the transmitting station and resumes monitoring the network.

The two-part deferral scheme was developed to prevent the possibility of the IPG being shortened due to a temporary loss of carrier. Figure 3.7 diagrams the two-part deferral process.

Simple Deferral: In the simple deferral scheme, the IPG timer is started whenever Carrier Sense
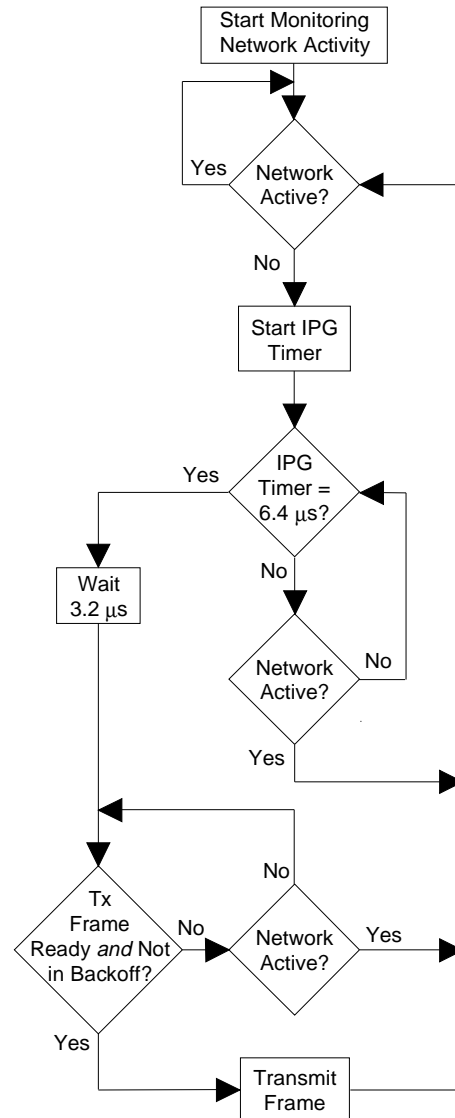
Figure 3.7. Two-Part Deferral

is de-asserted. Once the IPG timer is finished (after 9.6 µs), if a transmit frame is pending and if the MAC is not in Backoff, transmission begins (even if network activity is detected during the 9.6 µs IPG). If no transmit packet is pending, the MAC continues to monitor the network. If activity is detected before a transmit frame is ready, the MAC defers to the transmitting station and resumes monitoring the network. Figure 3.8 diagrams the simple deferral process.

Collision Resolution: If a collision is detected while the CS8900 is transmitting, the MAC re-
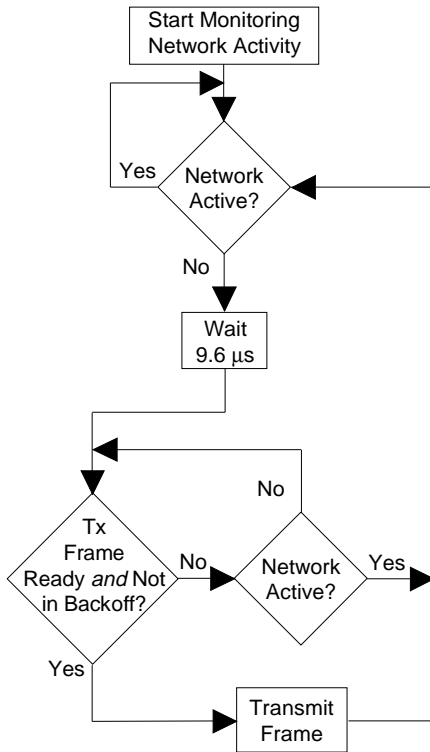
Figure 3.8. Simple Deferral

sponds in one of three ways depending on whether it is a normal collision (within the first 512 bits of transmission) or a late collision (after the first 512 bits of transmission):

Normal Collisions: If a collision is detected before the end of the preamble and SFD, the MAC finishes the preamble and SFD, transmits the jam sequence (32-bit pattern of all 0's), and then initiates Backoff. If a collision is detected after the transmission of the preamble and SFD but before 512 bit times, the MAC immediately terminates transmission, transmits the jam sequence, and then initiates Backoff. In either case, if the Onecoll bit (Register 9, TxCMD, Bit 9) is clear, the MAC will attempt to transmit a packet a total of 16 times (the initial attempt plus 15 retransmissions) due to normal collisions. On the 16th collision, it sets the 16coll bit (Register 8, TxEvent, Bit F) and discards the packet. If the Onecoll bit is set, the MAC discards the packet without attempting any re-transmission.

Late Collisions: If a collision is detected after the first 512 bits have been transmitted, the MAC immediately terminates transmission, transmits the jam sequence, discards the packet, and sets the Out-of-window bit (Register 8, TxEvent, Bit 9). The CS8900 does not initiate backoff or attempt to re-transmit the frame. For additional information about Late Collisions, see *Out-of-Window Error* in this section.

Backoff: After the MAC has completed transmitting the jam sequence, it must wait, or "Back off", before attempting to transmit again. The amount of time it must wait is determined by one of two Backoff algorithms: the Standard Backoff algorithm (ISO/IEC 4.2.3.2.5) or the Modified Backoff algorithm. The host selects which algorithm through the ModBackoffE bit (Register 13, LineCTL, Bit B).

*Standard Backoff:* The Standard Backoff algorithm, also called the "Truncated Binary Exponential Backoff", is described by the equation:

$$0 \le r \le 2^k$$

where r (a random integer) is the number of slot times the MAC must wait (1 slot time = 512 bit times), and k is the smaller of n or 10, where n is the number of re-transmission attempts.

Modified Backoff: The Modified Backoff is described by the equation:

$$0 \le r \le 2^k$$

where r (a random integer) is the number of slot times the MAC must wait, and k is 3 for n $<$3 and k is the smaller of n or 10 for n$\ge$3, where n is the number of re-transmission attempts.

The advantage of the Modified Backoff algorithm over the Standard Backoff algorithm is that it reduces the possibility of multiple collisions on the first three re-tries. The disadvantage is

that it extends the maximum time needed to gain access to the network for the first three re-tries.

The host may choose to disable the Backoff algorithm altogether by setting the DisableBackoff bit (Register 19, TestCTL, Bit B). When disabled, the CS8900 only waits the 9.6 µs IPG time before starting transmission.

SQE Test: If the CS8900 is transmitting on the AUI, the external transceiver should generate an SQE Test signal on the CI+/CI- pair following each transmission. The SQE Test is a 10 MHz signal lasting 5 to 15 bit times and starting within 0.6 to 1.6 µs after the end of transmission. During this period, the CS8900 ignores receive carrier activity (see SQE Error in this section for more information).

3.10   Encoder/Decoder (ENDEC)

The CS8900's integrated encoder/decoder (ENDEC) circuit is compliant with the relevant portions of section 7 of the Ethernet standard (ISO/IEC 8802-3, 1993). Its primary functions include: Manchester encoding of transmit data; informing the MAC when valid receive data is present (Carrier Detection); and, recovering the

clock and NRZ data from incoming Manchester-encoded data.

Figure 3.9 provides a block diagram of the ENDEC and how it interfaces to the MAC, AUI and 10BASE-T transceiver.

Encoder

The encoder converts NRZ data from the MAC and a 20 MHz Transmit Clock signal into a serial stream of Manchester data. The Transmit Clock is produced by an on-chip oscillator circuit that is driven by either an external 20 MHz quartz crystal or a TTL-level CMOS clock input. If a CMOS input is used, the clock should be 20 MHz ±0.01% with a duty cycle between 40% and 60%. The specifications for the crystal are described in section 13.0 (Quartz Crystal Requirements). The encoded signal is routed to either the 10BASE-T transceiver or AUI, depending on configuration.

Carrier Detection

The internal Carrier Detection circuit informs the MAC that valid receive data is present by asserting the internal Carrier Sense signal as soon it detects a valid bit pattern (1010b or 0101b for
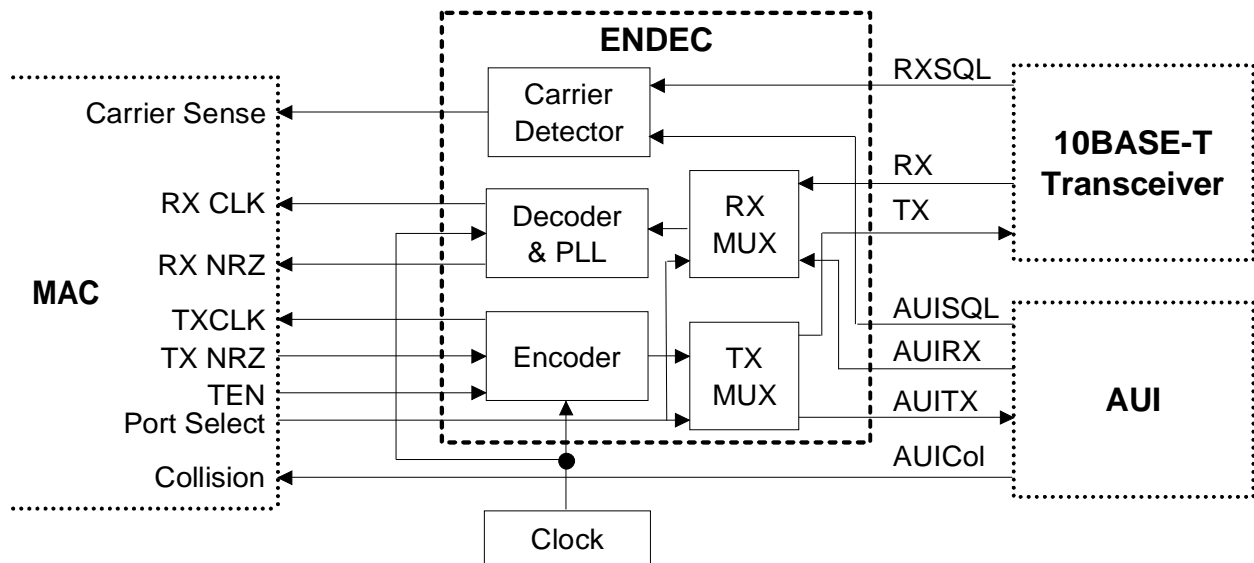

Figure 3.9. ENDEC

10BASE-T, and 1b or 0b for AUI). During normal packet reception, Carrier Sense remains asserted while the frame is being received, and is de-asserted 1.3 to 2.3 bit times after the last low-to-high transition of the End-of-Frame (EOF) sequence. Whenever the receiver is idle (no receive activity), Carrier Sense is de-asserted. The CRS bit (Register 14, LineST, Bit E) reports the state of the Carrier Sense signal.

Clock and Data Recovery

When the receiver is idle, the phase-lock loop (PLL) is locked to the internal clock signal. The assertion of the Carrier Sense signal interrupts the PLL. When it restarts, it locks on the incoming data. The receive clock is then compared to the incoming data at the bit cell center and any phase difference is corrected. The PLL remains locked as long as the receiver input signal is valid. Once the PLL has locked on the incoming data, the ENDEC converts the Manchester data to NRZ and passes the decoded data and the recovered clock to the MAC for further processing.

AUIonly (Bit 8)	AutoAUI/10BT (Bit 9)	Physical Interface
0	0	10BASE-T Only
1	N/A	AUI Only
0	1	Auto-Select

Table 3.11. Interface Selection

Interface Selection

Physical interface selection is determined by the AUIonly bit (Bit 8) and the AutoAUI/10BT bit (Bit 9) in the LineCTL register (Register 13). Table 3.11 describes the possible configurations.

10BASE-T Only: When configured for 10BASE-T-only operation, the 10BASE-T transceiver and its interface to the ENDEC are active, and the AUI is powered down.

AUI Only: When configured for AUI-only operation, the AUI and its interface to the ENDEC are active, and the 10BASE-T transceiver is powered down.

Auto-Select: In Auto-Select mode, the CS8900 automatically selects the 10BASE-T interface and powers down the AUI if valid packets and link pulses are detected by the 10BASE-T receiver. If valid packets and link pulses are not detected, the CS8900 selects the AUI. Whenever the AUI is selected, the 10BASE-T receiver remains active to listen for link pulses or packets. If 10BASE-T activity is detected, the CS8900 switches back to 10BASE-T.

3.11 10BASE-T Transceiver

The CS8900 includes an integrated 10BASE-T transceiver that is compliant with the relevant portions of section 14 of the Ethernet standard (ISO/IEC 8802-3, 1993). It includes all analog and digital circuitry needed to interface the CS8900 directly to a simple isolation transformer (see section 11.0 for a connection diagram). Figure 3.10 provides a block diagram of the 10BASE-T transceiver.

10BASE-T Filters

The CS8900's 10BASE-T transceiver includes integrated low-pass transmit and receive filters, eliminating the need for external filters or a filter/transformer hybrid. On-chip filters are gm/c implementations of fifth-order Butterworth low-pass filters. Internal tuning circuits keep the gm/c ratio tightly controlled, even when large temperature, supply, and IC process variations occur. The nominal 3 dB cutoff frequency of the filters is 16 MHz, and the nominal attenuation at 30 MHz (3rd harmonic) is -27 dB.
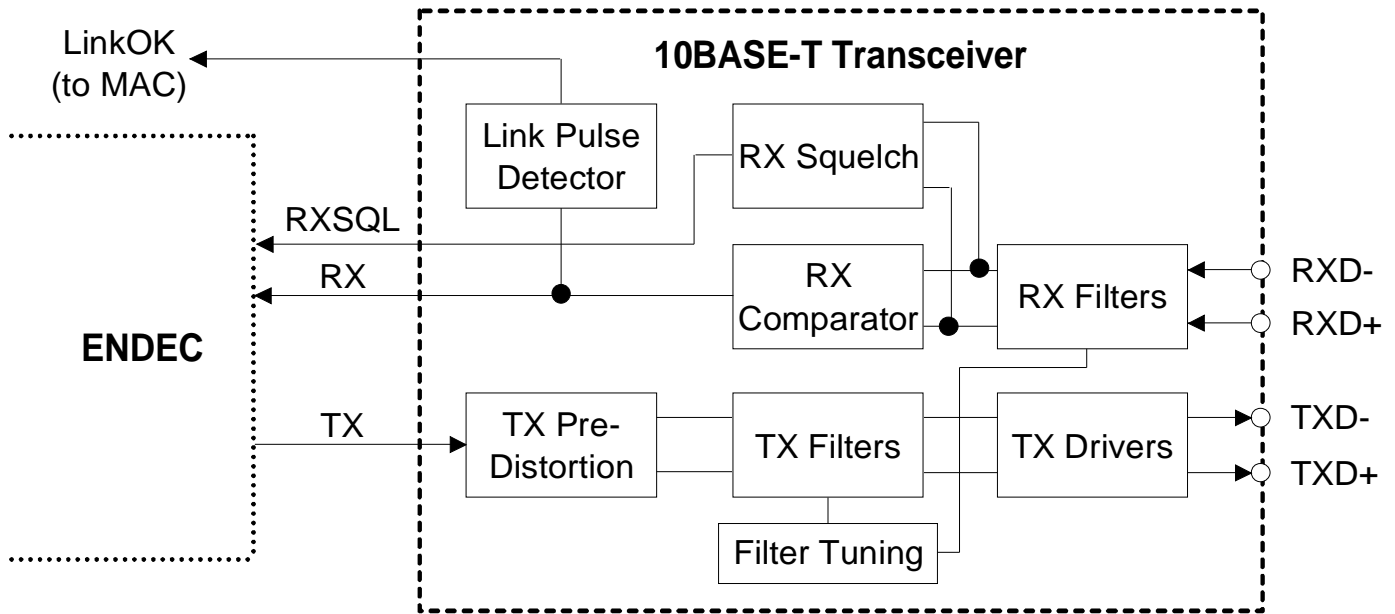
Figure 3.10. 10BASE-T Transceiver

Transmitter

When configured for 10BASE-T operation, Manchester encoded data from the ENDEC is fed into the transmitter's pre-distortion circuit where initial wave shaping and pre-equalization is performed. The output of the pre-distortion circuit is fed into the transmit filter where final wave shaping occurs and unwanted noise is removed. The signal then passes to the differential driver where it is amplified and driven out of the TXD+/TXD- pins.

In the absence of transmit packets, the transmitter generates link pulses in accordance with section 14.2.1.1. of the Ethernet standard. Transmitted link pulses are positive pulses, one bit time wide, typically generated at a rate of one every 16 ms. The 16 ms timer starts whenever the transmitter completes an End-of-Frame (EOF) sequence. Thus, there is a link pulse 16 ms after an EOF unless there is another transmit-

ted packet. Figure 3.11 diagrams the operation of the Link Pulse Generator.

If no link pulses are being received on the receiver, the 10BASE-T transmitter is internally forced to an inactive state unless bit DisableLT in register 19 (Test Control register) is set to one.

Receiver

The 10BASE-T receive section consists of the receive filter, squelch circuit, polarity detection and correction circuit, and link pulse detector.

Squelch Circuit: The 10BASE-T squelch circuit determines when valid data is present on the RXD+/RXD- pair. Incoming signals passing through the receive filter are tested by the squelch circuit. Any signal with amplitude less than the squelch threshold (either positive or negative, depending on polarity) is rejected.

Extended Range: The CS8900 supports an Extended Range feature that reduces the 10BASE-T receive squelch threshold by approximately 6 dB. This allows the CS8900 to operate with 10BASE-T cables that are longer than 100 meters (100 meters is the maximum length specified by the Ethernet standard). The exact additional distance depends on the quality of the cable and the amount of electro-magnetic noise in the surrounding environment. To activate this feature, the host must set the LoRxSquelch bit (Register 13, LineCTL, Bit E).

Link Pulse Detection

To prevent disruption of network operation due to a faulty link segment, the CS8900 continually monitors the 10BASE-T receive pair (RXD+/RXD-) for packets and link pulses. After each packet or link pulse is received, an internal Link-Loss timer is started. As long as a packet or link pulse is received before the Link-Loss timer finishes (between 25 and 150 ms), the CS8900 maintains normal operation. If no receive activity is detected, the CS8900 disables packet transmission to prevent "blind" transmissions onto the network (link pulses are still sent while packet transmission is disabled). To reactivate transmission, the receiver must detect a single packet (the packet itself is ignored), or two link pulses separated by more than 2 to 7 ms and no more than 25 to 150 ms (see section 10.0 for 10BASE-T timing).

The state of the link segment is reported in the LinkOK bit (Register 14, LineST, Bit 7). If the HC0E bit (Register 15, SelfCTL, Bit D) is clear, it is also indicated by the output of the LINKLED pin. If the link is "good", the LinkOK bit is set and the $\overline{\text{LINKLED}}$ pin is driven low. If the link is "bad" the LinkOK bit is clear and the $\overline{\text{LINKLED}}$ pin is high. To disable this feature, the host must set the DisableLT bit (Register 19, TestCTL, Bit 7). If DisableLT is set, the CS8900 will transmit and receive packets independent of the link segment.

Receive Polarity Detection and Correction

The CS8900 automatically checks the polarity of the receive half of the twisted pair cable. If the polarity is correct, the PolarityOK bit (Register 14, LineST, bit C) is set. If the polarity is reversed, the PolarityOK bit is clear. If the PolarityDis bit (Register 19, TestCTL, Bit C) is clear, the CS8900 automatically corrects a reversal. If the PolarityDis bit is set, the CS8900 does not correct a reversal. The PolarityOK bit and the PolarityDis bit are independent.

To detect a reversed pair, the receiver examines received link pulses and the End-of-Frame (EOF) sequence of incoming packets. If it detects at least one reversed link pulse and at least four frames in a row with negative polarity after the EOF, the receive pair is considered reversed. Any data received before the correction of the reversal is ignored.
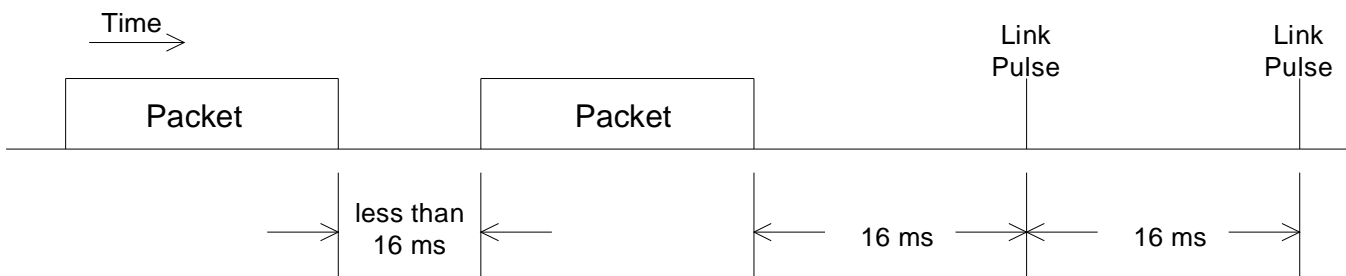
Figure 3.11. Link Pulse Transmission

Collision Detection

If half-duplex operation is selected (Register 19, Bit E, FDX), the CS8900 detects a 10BASE-T collision whenever the receiver and transmitter are active simultaneously. When a collision is present, the Collision Detection circuit informs the MAC by asserting the internal Collision signal (see section 3.10 for collision handling).

3.12   Attachment Unit Interface (AUI)

The CS8900 Attachment Unit Interface (AUI) provides a direct interface to external 10BASE2, 10BASE5, and 10BASE-FL Ethernet transceivers. It is fully compliant with Section 7 of the Ethernet standard (ISO/IEC 8802-3), and as such, is capable of driving a full 50-meter AUI cable.

The AUI consists of three pairs of signals: Data Out (DO+/DO-), Data In (DI+/DI-), and Collision In (CI+/CI-). To select the AUI, the host should set the AUI bit (Register 13, LineCTL, Bit 9). The AUI can also be selected automatically as described in the previous section (Section 3.10, *Interface Selection*). Figure 3.12 provides a block diagram of the AUI. (For a connection diagram, see section 12.0).

AUI Transmitter

The AUI transmitter is a differential driver designed to drive a 78 ohm cable. It accepts data from the ENDEC and transmits it directly on the DO+/DO- pins. After transmission has started, the CS8900 expects to see the packet "looped-back" (or echoed) to the receiver, causing the Carrier Sense signal to be asserted. This Carrier Sense presence indicates that the transmit signal is getting through to the transceiver. If the Carrier Sense signal remains de-asserted throughout the transmission, or if the Carrier Sense signal is de-asserted before the end of the transmission,

there is a Loss-of-Carrier error and the Loss-of-CRS bit (Register 8, TxEvent, Bit 6) is set.

AUI Receiver

The AUI receiver is a differential pair circuit that connects directly to the DI+/DI- pins. It is designed to distinguish between transient noise pulses and incoming Ethernet packets. Incoming packets with proper amplitude and pulse width are passed on to the ENDEC section, while unwanted noise is rejected.

Collision Detection

The AUI collision circuit is a differential pair receiver that detects the presence of collision signals on the CI+/CI- pins. The collision signal is generated by an external Ethernet transceiver whenever a collision is detected on the Ethernet segment. (Section 7.3.1.2 of ISO/IEC 8802-3, 1993, defines the collision signal as a 10 MHz +/- 15% signal with a duty cycle no worse than 60/40). When a collision is present, the AUI Collision circuit informs the MAC by asserting the internal Collision signal.

Figure 3.12. AUI

3.13   External Clock Oscillator

A 20-MHz quartz crystal or CMOS clock input is required by the CS8900. If a CMOS clock input is used, it should be connected the to XTAL1 pin, with the XTAL2 pin left open. The clock signal should be 20 MHz ±0.01% with a duty

cycle between 40% and 60%. The specifications for the crystal are described in section 13.0 (Quartz Crystal Requirements).

4.0 PACKETPAGE ARCHITECTURE

4.1 PacketPage Overview

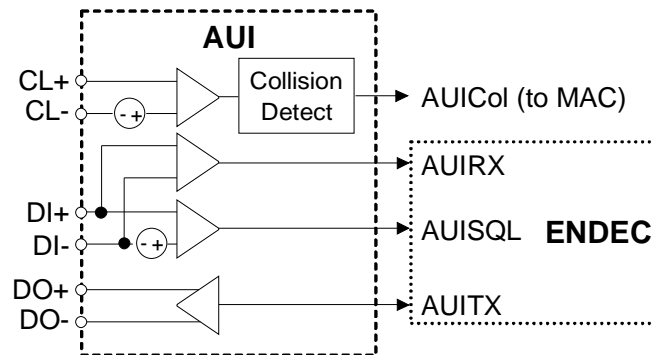The CS8900 architecture is based on a unique, highly-efficient method of accessing internal registers and buffer memory known as PacketPage. PacketPage provides a unified way of controlling the CS8900 in Memory or I/O space that minimizes CPU overhead and simplifies software. It provides a flexible set of performance features and configuration options, allowing designers to develop Ethernet circuits that meet their particular system requirements.

Integrated Memory

Central to the CS8900 architecture is a 4-Kbyte page of integrated RAM known as PacketPage memory. PacketPage memory is used for temporary storage of transmit and receive frames, and for internal registers. Access to this memory is done directly, through Memory space operations (Section 4.9), or indirectly, though I/O space operations (Section 4.10). In most cases, Memory Mode will provide the best overall performance, because ISA Memory operations require fewer cycles than I/O operations. I/O Mode is the CS8900's default configuration and is used when memory space is not available or when special operations are required (e.g. waking the CS8900 from the Software Sleep state requires the host to write to the CS8900's assigned I/O space).

The user-accessible portion of PacketPage memory is organized into the following six sections:

PacketPage Address	Contents
0000h - 0045h	Bus Interface Registers
0100h - 013Fh	Status and Control Registers
0140h - 014Fh	Initiate Transmit Registers
0150h - 015Dh	Address Filter Registers
0400h	Receive Frame Location
0A00h	Transmit Frame Location

Bus Interface Registers

The Bus Interface registers are used to configure the CS8900's ISA-bus interface and to map the CS8900 into the host system's I/O and Memory space. Most of these registers are written only during initialization, remaining unchanged while the CS8900 is in normal operating mode. The exceptions to this are the DMA registers which are modified continually whenever the CS8900 is using DMA. These registers are described in more detail in Section 4.3.

Status and Control Registers

The Status and Control registers are the primary means of controlling and getting status of the CS8900. They are described in more detail in Section 4.4.

Initiate Transmit Registers

The TxCMD/TxLength registers are used to initiate Ethernet frame transmission. These registers are described in more detail in Section 4.5. (See Section 5.7 for a description of frame transmission.)

Address Filter Registers

The Filter registers store the Individual Address filter and Logical Address filter used by the Destination Address (DA) filter. These registers are described in more detail in Section 4.6. For a description of the DA filter, see Section 5.3.

Receive and Transmit Frame Locations

The Receive and Transmit Frame PacketPage locations are used to transfer Ethernet frames to and from the host. The host simply writes to and reads from these locations and internal buffer memory is dynamically allocated between transmit and receive as needed. This provides more efficient use of buffer memory and better overall network performance. As a result of this dynamic

allocation, only one receive frame (starting at PacketPage base + 0400h) and one transmit frame (starting at PacketPage base + 0A00h) are directly accessible. See Section 4.7.

4.2 PacketPage Memory Map

Table 4.1 shows the CS8900 PacketPage memory address map:

PacketPage Address	# of Bytes	Type	Description	Cross Reference
			Bus Interface Registers	
0000h	4	Read-only	Product Identification Code	Section 4.3
0004h	28	-	Reserved	Note 2
0020h	2	Read/Write	I/O Base Address	Sections 4.3, 4.10
0022h	2	Read/Write	Interrupt Number (0, 1, 2, or 3)	Sections 3.2, 4.3
0024h	2	Read/Write	DMA Channel Number (0, 1, or 2)	Sections 3.2, 4.3
0026h	2	Read-only	DMA Start Of Frame	Sections 4.3, 5.4
0028h	2	Read-only	DMA Frame Count (12 bits)	Sections 4.3, 5.4
002Ah	2	Read-only	RxDMA Byte Count	Sections 4.3, 5.4
002Ch	4	Read/Write	Memory Base Address Register (20 bit)	Sections 4.3, 4.9
0030h	4	Read/Write	Boot PROM Base Address	Sections 3.6, 4.3
0034h	4	Read/Write	Boot PROM Address Mask	Sections 3.6, 4.3
0038h	8	-	Reserved	Note 2
0040h	2	Read/Write	EEPROM Command	Sections 3.5, 4.3
0042h	2	Read/Write	EEPROM Data	Sections 3.5, 4.3
0044h	12	-	Reserved	Note 2
0050h	2	Read only	Received Frame Byte Counter	Sections 4.3, 5.2.9
0052h	174	-	Reserved	Note 2
			Status and Control Registers	
0100h	32	Read/Write	Configuration & Control Registers (2 bytes per register)	Section 4.4
0120h	32	Read-only	Status & Event Registers (2 bytes per register)	Section 4.4
0140h	4	-	Reserved	Note 2
			Initiate Transmit Registers	
0144h	2	Write-only	TxCMD (transmit command)	Sections 4.5, 5.7
0146h	2	Write-only	TxLength (transmit length)	Sections 4.5, 5.7
0148h	8	-	Reserved	Note 2
			Address Filter Registers	
0150h	8	Read/Write	Logical Address Filter (hash table)	Sections 4.6, 5.3
0158h	6	Read/Write	Individual Address	Sections 4.6, 5.3
015Eh	674	-	Reserved	Note 2
			Frame Location	
0400h	2	Read-only	RxStatus (receive status)	Sections 4.7, 5.2
0402h	2	Read-only	RxLength (receive length, in bytes)	Sections 4.7, 5.2
0404h	2	Read-only	Receive Frame Location	Sections 4.7, 5.2
0A00h	2	Read/Write	Transmit Frame Location	Sections 4.7, 5.7

NOTES: 1. All registers are accessed as words only. 2. Read operation from the reserved location provides undefined data. Writing to a reserved location or undefined bits may result in unpredictable operation of the CS8900.

Table 4.1 PacketPage Memory Address Map

4.3   Bus Interface Registers

Bus Interface Register:
Product Identification Code (Read only)   *Address: PacketPage base + 0000h*

Address 0000h	Address 0001h	Address 0002h	Address 0003h
First byte of EISA registration number for Crystal Semiconductor	Second byte of EISA registration number for Crystal Semiconductor	First 8 bits of Product ID Number	Last 3 bits of the Product ID number (5 "X" bits are the revision number)

The Product Identification Code Register is located in the first four bytes of the PacketPage (0000h to 0003h). The register contains a unique 32-bit product ID code that identifies the chip as a CS8900. The host can use this number to determine which software driver to load and to check which features are available.

This register's initial state after reset is:

0000 1110	0110 0011	0000 0000	000X XXXX

The X XXXX codes for the CS8900 are:
   Rev E: 0 0011
   Rev F: 0 0101

Bus Interface Register:
I/O Base Address (Read/Write)   *Address: PacketPage base + 0020h*

Address 0020h	Address 0021h
Most significant byte of I/O Base Address	Least significant byte of I/O Base Address

The I/O Base Address Register describes the base address for the sixteen contiguous locations in the host system's I/O space, which are used to access the PacketPage registers. See Section 4.10. The default location is 0300h.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

0000 0011	0000 0000

Bus Interface Register:
Interrupt Number (Read/Write)   *Address: PacketPage base + 0022h*

Address 0023h	Address 0022h
00h	Interrupt number assignment: 0000 0000b= pin INTRQ0 0000 0001b= pin INTRQ1 0000 0010b= pin INTRQ2 0000 0011b= pin INTRQ3 0000 01XXb= All INTRQ pins high-impedance

The Interrupt Number Register defines the interrupt pin selected by the CS8900. In a typical application the following bus signals are tied to the following pins:

Bus signal	Typical pin connection
IRQ5	INTRQ3
IRQ10	INTRQ0
IRQ11	INTRQ1
IRQ12	INTRQ2

See Section 3.2.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state, which corresponds to placing all the INTRQ pins in a high-impedance state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

XXXX XXXX	XXXX X100

Bus Interface Register:
DMA Channel Number (Read/Write)   *Address: PacketPage base + 0024h*

Address 0025h	Address 0024h
00h	DMA channel assignment: 0000 0000b = pins DMRQ0 and $\overline{DMACK0}$ 0000 0001b = pins DMRQ1 and $\overline{DMACK1}$ 0000 0010b = pins DMRQ2 and $\overline{DMACK2}$ 0000 01XXb = All INTRQ pins high-impedance

The DMA Channel register defines the DMA pins selected by the CS8900. In the typical application, the following bus signals are tied to the following pins:

Bus signal	Typical pin connection
$\overline{DRQ5}$ DACK5	$\overline{DMRQ0}$ DMACK0
$\overline{DRQ6}$ DACK6	$\overline{DMRQ1}$ DMACK1
$\overline{DRQ7}$ DACK7	$\overline{DMRQ2}$ DMACK2

See Sections 3.2 and 5.4.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state which corresponds to selecting pins DMRQ0 and DMACK0. If a EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

XXXX XXXX	XXXX XX11

Bus Interface Register:
DMA Start of Frame (Read only)   **Address: PacketPage base + 0026h**

Address 0027h	Address 0026h
Most-significant byte of offset value	Least-significant byte of offset value

The DMA Start of Frame Register contains a 16-bit value which defines the offset from the DMA base address to the start of the most recently transferred received frame. See Section 5.4.

This register's initial state after reset is: | 0000 0000 | 0000 0000 |

Bus Interface Register:
DMA Frame Count (Read only)   **Address: PacketPage base + 0028h**

Address 0029h	Address 0028h
Most-significant byte of frame count (most-significant nibble always 0h)	Least-significant byte of frame count

The lower 12 bits of the DMA Frame Count register define the number of valid frames transferred via DMA since the last readout of this register. The upper 4 bits are reserved. See Section 5.4.

This register's initial state after reset is: | XXXX 0000 | 0000 0000 |

Bus Interface Register:
RxDMA Byte Count (Read only)   *Address: PacketPage base + 002Ah*

Address 002Bh	Address 002Ah
Most-significant byte of byte count.	Least-significant byte of byte count.

The RxDMA Byte Count register describes the valid number of bytes DMAed since the last readout. See Section 5.4.

This register's initial state after reset is: | 0000 0000 | 0000 0000 |

Bus Interface Register:
Memory Base Address   (Read/Write)   *Address: PacketPage base + 002Ch*

Address 002Fh	Address 002Eh	Address 002Dh	Address 002Ch
Reserved	The most-significant nibble of memory base address. The high-order nibble is reserved.	Contains portion of memory base address.	The least-significant byte of the memory base address.

Memory Base Address: The lower three bytes (002Ch, 002Dh, and 002Eh) are used for the 20-bit memory base address. The upper three nibbles are reserved.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

| XXXX XXXX | XXXX 0000 | 0000 0000 | 0000 0000 |

Bus Interface Register:
Boot PROM Base Address (Read/Write)   *Address: PacketPage base + 0030h*

Address 0033h	Address 0032h	Address 0031h	Address 0030h
Reserved	The most-significant nibble of Boot PROM base address. The high-order nibble is reserved.	Contains portion of Boot PROM base address.	The least-significant byte of the Boot PROM base address.

The lower three bytes (0030h, 0031h, and 0032h) of the Boot PROM Base Address register are used for the 20-bit Boot PROM base address. The upper three nibbles are reserved. See Section 3.6.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

XXXX XXXX	XXXX 0000	0000 0000	0000 0000

Bus Interface Register:
Boot PROM Address Mask (Read/Write)   *Address: PacketPage base + 0034h*

Address 0037h	Address 0036h	Address 0035h	Address 0034h
Reserved	The most-significant nibble of Boot PROM mask address. The high-order nibble is reserved.	Contains portion of Boot PROM mask address. The lower-order nibble must be written as 0h.	The least-significant byte of the Boot PROM mask address. Must be written as 00h

The Boot PROM address mask register indicates the size of the attached Boot PROM and is limited to 4K bit increments. The lower 12 bits of the Address Mask are ignored, and should be 000h. The next lowest-order bits describe the size of the PROM. The upper three nibbles are reserved.

For example:

Size of Boot PROM	Register value			
4k bits	XXXX XXXX	XXXX 1111	1111 0000	0000 0000
8k bits	XXXX XXXX	XXXX 1111	1110 0000	0000 0000
16k bits	XXXX XXXX	XXXX 1111	1100 0000	0000 0000

See Section 3.6.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

XXXX XXXX	XXXX 0000	0000 0000	0000 0000

Bus Interface Register:
EEPROM Command (Read/Write)   ***Address: PacketPage base + 0040h***

F-B	A	9	8	7-0
Reserved	ELSEL	OB1	OB0	ADD7 to ADD0

This register is used to control the reading, writing and erasing of the EEPROM. See Section 3.5.

BIT	NAME	DESCRIPTION
7-0	ADD7-ADD0	Address of the EEPROM word being accessed.
9-8	OB1,OB0	Indicates the Opcode of the command being executed. See Table 3.7.
A	ELSEL	External logic select: When clear, the EECS pin is used to select the EEPROM. When set, the ELCS pin is used to select the external LA decode circuit..
F-B	Reserved	Reserved and must be written as 0.

This register's initial state after reset is: | XXXX XXXX | XXXX XXXX |

Bus Interface Register:
EEPROM Data (Read/Write)   ***Address: PacketPage base + 0042h***

Address 0043h	Address 0042h
Most-significant byte of EEPROM data.	Least-significant byte of the EEPROM data.

This register contains the word being written to, or read from, the EEPROM. See Section 3.5.

This register's initial state after reset is: | XXXX XXXX | XXXX XXXX |

Bus Interface Register:
Receive Frame Byte Counter (Read/only)   ***Address: PacketPage base + 0050h***

Address 0051h	Address 0050h
Most-significant byte of byte count.	Least-significant byte of the byte count.

This register contains the count of the total number bytes received in the the current received frame. This count continuously increments as more bytes in this frame are received. See Section 5.2.9.

This register's initial state after reset is: | XXXX XXXX | XXXX XXXX |

4.4 Status and Control Registers

The Status and Control registers are the primary registers used to control and status the CS8900. They are organized into two groups: Configuration/Control Registers and Status/Event Registers. All Status and Control Registers are 16-bit words as shown in Figure 4.1. Bit 0 indicates whether it is a Configuration/Control Register (Bit $0 = 1$) or a Status/Event Register (Bit $0 = 0$). Bits 0 through 5 provide an internal address code that describes the exact function of the register. Bits 6 through F are the actual Configuration/Control and Status/Event bits.

Configuration and Control Registers

Configuration and Control registers are used to setup the following:

- how frames will be transmitted and received;

- which frames will be transmitted and received;

- which events will cause interrupts to the host processor; and,

- how the Ethernet physical interface will be configured.

These registers are read/write and are designated by odd numbers (e.g. Register 1, Register 3, etc.).

The Transmit Command Register (TxCMD) is a special type of register. It appears in two separate locations in the PacketPage memory map. The first location, PacketPage base + 0108h, is within the block of Configuration/Control Registers and is read-only. The second location, PacketPage base + 0144h, is where the actual transmit commands are issued and is write-only. See Section 4.4 (Register 9) and Section 5.7 for a more detailed description of the TxCMD register.

Status and Event Registers

Status and Event registers report the status of transmitted and received frames, as well as information about the configuration of the CS8900. They are read-only and are designated by even numbers (e.g. Register 2, Register 4, etc.).

The Interrupt Status Queue (ISQ) is a special type of Status/Event register. It is located at PacketPage base + 0120h and is the first register the host reads when responding to an Interrupt. A more detailed description of the ISQ can be found in Section 5.1.
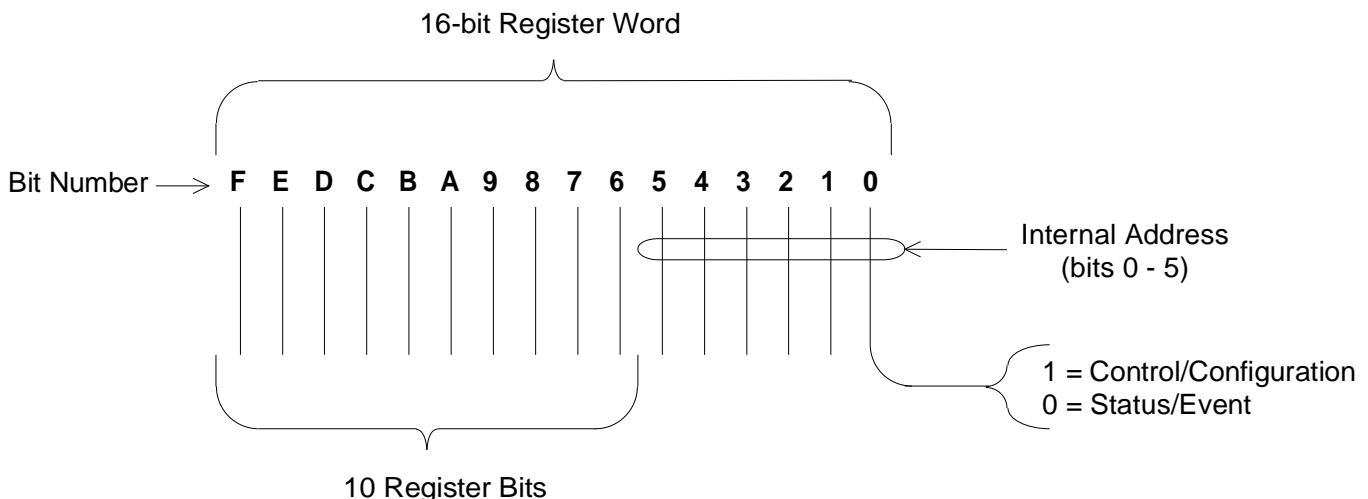

Figure 4.1. Status and Control Register Format

Three 10-bit counters are included with the Status and Event registers. RxMISS counts missed receive frames, TxCOL counts transmit collisions, and TDR is a time domain reflectometer useful in locating cable faults. The following sections contain more information about these counters.

Table 4.2 provides a summary of PacketPage Register types.

4.4.1 Status and Control Bit Definitions

This section provides a description of the special bit types used in the Status and Control registers. Section 4.4 provides a detailed description of the bits in each register.

Act-Once Bits

There are four bits that cause the CS8900 to take a certain action only once when set. These "Act-Once" bits are: Skip_1 (Register 3, RxCFG, Bit 6), RESET (Register 15, SelfCTL, Bit 6), ResetRxDMA (Register 17, BusCTL, Bit 6), and SWint-X (Register B, BufCFG, Bit 6). To cause the action again, the host must set the bit again. Act-Once bits are always read as clear.

Temporal Bits

Temporal bits are bits that are set and cleared by the CS8900 without intervention of the host processor. This includes all status bits in the three status registers (Register 14, LineST; Register 16, SelfST; and, Register 18, BusST), the RxDest bit (Register C, BufEvent, Bit F), and the Rx128 bit (Register C, BufEvent, Bit F). Like all Event bits, RxDest and Rx128 are cleared when read by the host.

Interrupt Enable Bits and Events

Interrupt Enable bits end with the suffix iE and are located in three Configuration registers: RxCFG (Register 3), TxCFG (Register 7), and BufCFG (Register B). Each Interrupt Enable bit corresponds to a specific event. If an Interrupt Enable bit is set and its corresponding event occurs, the CS8900 generates an interrupt to the host processor.

The bits that report when various events occur are located in three Event registers and two counters. The Event registers are RxEvent (Register 4), TxEvent (Register 8), and BufEvent (Register C). The counters are RxMISS (Register

Suffix	Type	Description	Comments
CMD	Read/Write	**Command:** Written once per frame to initiate transmit.	
CFG	Read/Write	**Configuration:** Written at setup and used to determine what frames will be transmitted and received and what events will cause interrupts.	
CTL	Read/Write	**Control:** Written at setup and used to determine what frames will be transmitted and received and how the physical interface will be configured.	
Event	Read-only	**Event:** Reports the status of transmitted and received frames.	cleared when read
ST	Read-only	**Status:** Reports information about the configuration of the CS8900.	
	Read-only	**Counters:** Counts missed receive frames and collisions. Provides time domain reflectometer for locating coax cable faults.	cleared when read

Table 4.2. PacketPage Register Types

10) and TxCOL (Register 12). Each Interrupt Enable bit and its associated Event are identified in Table 4.3.

Interrupt Enable Bit (register name)	Event Bit or Counter (register name)
ExtradataiE (RxCFG)	Extradata (RxEvent)
RuntiE (RxCFG)	Runt (RxEvent)
CRCerroriE (RxCFG)	CRCerror (RxEvent)
RxOKiE (RxCFG)	RxOK (RxEvent)
16colliE (TxCFG)	16coll (TxEvent)
AnycolliE (TxCFG)	"Number-of-Tx-collisions" counter is incremented (TxEvent)
JabberiE (TxCFG)	Jabber (TxEvent)
Out-of-windowiE (TxCFG)	Out-of-window (TxEvent)
TxOKiE (TxCFG)	TxOK (TxEvent)
SQEerroriE (TxCFG)	SQEerror (TxEvent)
Loss-of-CRSiE (TxCFG)	Loss-of-CRS (TxEvent)
MissOvfloiE (BufCFG)	RxMISS counter overflows past 1FFh
TxColOvfloiE (BufCFG)	TxCOL counter overflows past 1FFh
RxDestiE (BufCFG)	RxDest (BufEvent)
Rx128iE (BufCFG)	Rx128 (BufEvent)
RxMissiE (BufCFG)	RxMISS (BufEvent)
TxUnderruniE (BufCFG)	TxUnderrun (BufEvent)
Rdy4TxiE (BufCFG)	Rdy4Tx (BufEvent)
RxDMAiE (BufCFG)	RxDMAFrame (BufEvent)

Table 4.3. Interrupt Enable Bits and Events

An Event bit will be set whenever the specified event happens, whether or not the associated Interrupt Enable bit is set. All Event registers are cleared upon read-out by the host.

Accept Bits

There are nine Accept bits located in the RxCTL register (Register 5), each of which is followed by the suffix A. Accept bits indicate which types of frames will be accepted by the CS8900. (A frame is said to be "accepted" by the CS8900 when the frame data are placed in either on-chip memory, or in host memory by DMA.) Four of these bits have corresponding Interrupt Enable

(iE) bits. An Accept bit and an Interrupt Enable bit are independent operations. It is possible to set either, neither, or both bits. The four corresponding pairs of bits are:

iE Bit in RxCFG	A Bit in RxCTL
ExtradataiE	ExtradataA
RuntiE	RuntA
CRCerroriE	CRCerrorA
RxOKiE	RxOKA

If one of the above Interrupt Enable bits is set and the corresponding Accept bit is clear, the CS8900 generates an interrupt when the associated receive event occurs, but then does not accept the receive frame (the length of the receive frame is set to zero).

The other five Accept bits in RxCTL are used for destination address filtering (see Section 5.3). The Accept mechanism is explained in more detail in Section 5.2.

4.4.2 *Status and Control Register Summary*

The figure on the following page (Figure 4.2) provides a summary of the Status and Control registers. Section 4.4.2 gives a detailed description of each Status and Control register.

F	E	D	C	B	A	9	8	7	6	Register Number (Offset)	Name
Control and Configuration Bits										**Register**	
F	E	D	C	B	A	9	8	7	6	**Number (Offset)**	**Name**
Reserved (register contents undefined)										1	
	Extra dataiE	RuntiE	CRC erroriE	Buffer CRC	AutoRx DMAE	RxDMA only	RxOKiE	StreamE	Skip_1	3 (0102h)	RxCFG
	Extra dataA	RuntA	CRC errorA	Broad castA	Individ ualA	Multi castA	RxOKA	Promis cuousA	IAHashA	5 (0104h)	RxCTL
16colliE				AnycolliE	JabberiE	Out-of-windowiE	TxOKiE	SQEerror iE	Loss-of-CRSiE	7 (0106h)	TxCFG
		TxPadDis	Inhibit CRC			Onecoll	Force	TxStart		9 (0108h)	TxCMD
RxDestiE		Miss OvfloiE	TxCol OvfloiE	Rx128iE	RxmissiE	TxUnder runiE	Rdy4TxiE	RxDMAiE	SWint-X	B (010Ah)	BufCFG
Reserved (register contents undefined)										D - 11	
	LoRx Squelch	2-part DefDis	Polarity Dis	Mod BackoffE		Auto AUI/10BT	AUIonly	SerTxON	SerRxON	13 (0112h)	LineCTL
HCB1	HCB0	HC1E	HC0E		HW StandbyE	HW SleepE	SW Suspend		RESET	15 (0114h)	SelfCTL
Enable IRQ		RxDMA size	IOCH RDYE	DMA Burst	MemoryE	UseSA			Reset RxDMA	17 (0116)	BusCTL
FastTest	FDX			Disable Backoff	AUIloop	ENDEC loop		Disable LT		19 (0118)	TestCTL
Reserved (register contents undefined)										1B - 1F	
Status and Event Bits										**Register**	
F	E	D	C	B	A	9	8	7	6	**Number (Offset)**	**Name**
Interrupt Status Queue										0 (0120h)	ISQ
Reserved (register contents undefined)										2	
	Extra data	Runt	CRC error	Broad cast	Individual Adr	Hashed	RxOK	Dribble bits	IAHash	4 (0124h)	RxEvent
Hash Table Index (alternate RxEvent meaning if Hashed = 1 and RxOK =1)						Hashed	RxOK	Dribble bits	IAHash	4 (0124h)	RxEvent alternate
Reserved (register contents undefined)										6	
16coll	Number-of-Tx-collisions				Jabber	Out-of-window	TxOK	SQE error	Loss-of-CRS	8 (0128h)	TxEvent
Reserved (register contents undefined)										A	
RxDest				Rx128	RxMiss	TxUnder run	Rdy4Tx	RxDMA Frame	SWint	C (012Ch)	BufEvent
Reserved (register contents undefined)										E	
10-bit Receive Miss (RxMISS) counter, cleared when read										10 (0130h)	RxMISS
10-bit Transmit Collision (TxCOL) counter, cleared when read										12 (0132h)	TxCOL
	CRS		Polarity OK			10BT	AUI	LinkOK		14 (0134h)	LineST
		EEsize	EL present	EEPROM OK	EEPROM present	SIBUSY	INITD		3.3V Active	16 (0136h)	SelfST
							Rdy4Tx NOW	TxBid Err		18 (0138h)	BusST
Reserved (register contents undefined)										1A	
10-bit AUI Time Domain Reflectometer (TDR) counter, cleared when read										1C (013Ch)	TDR
Reserved (register contents undefined)										1E	

Figure 4.2. Status and Control Register Summary

4.4.3 Status and Control Register Descriptions

Register 0: Interrupt Status Queue (ISQ, Read-only)   **Address: PacketPage base + 0120h**

F-6	5-0
RegContent	RegNum

The Interrupt Status Queue Register is used in both Memory Mode and I/O Mode to provide the host with interrupt information. Whenever an event occurs that triggers an enabled interrupt, the CS8900 sets the appropriate bit(s) in one of five registers, maps the contents of that register to the ISQ register, and drives an IRQ pin high. Three of the registers mapped to ISQ are event registers: RxEvent (Register 4), TxEvent (Register 8), and BusEvent (Register C). The other two registers are counter-overflow reports: RxMISS (Register 10) and TxCOL (Register 12). In Memory Mode, ISQ located at PacketPage base + 120h. In I/O Mode, ISQ is located at I/O Base + 0008h. See Section 5.1.

BIT	NAME	DESCRIPTION

5-0   RegNum   The lower six bits describe which register (4, 8, C, 10 or 12) is contained in the ISQ.

6   RegContent   The upper ten bits contain the register data contents.

This register's initial state after reset is: | 0000 0000 | 0000 0000 |

Register 3: Receiver Configuration (RxCFG, Read/Write)   *Address: PacketPage base + 0102h*

F	E	D	C	B	A	9	8	7	6	5-0
	Extra dataiE	RuntiE	CRC erroriE	Buffer CRC	AutoRx DMAE	RxDMA only	RxOKiE	StreamE	Skip_1	000011

RxCFG determines how frames will be transferred to the host and what frame types will cause interrupts.

BIT	NAME	DESCRIPTION
5-0	000011	These bits provide an internal address used by the CS8900 to identify this as the Receiver Configuration Register. To write to this register, these bits must be 000011, where the LSB corresponds to Bit 0.
6	Skip_1	When set, this bit causes the last committed received frame to be deleted from the receive buffer. To skip another frame, the host must re-write a "1" to this bit. This bit is not to be used if RxDMAonly (Bit 9) is set. Skip_1 is an Act-Once bit. See Section 5.2.5
7	StreamE	When set, StreamTransfer mode is used to transfer receive frames that are back-to-back *and* that pass the Destination Address filter (see Section 5.3). When StreamE is clear, StreamTransfer mode is not used. This bit must not be set unless either bit AutoRXDMA or bit RXDMA-only is set.
8	RxOKiE	When set, there is an RxOK Interrupt if a frame is received without errors.
9	RxDMAonly	The Receive-DMA mode is used for all receive frames when this bit is set.
A	AutoRxDMAE	When set, the CS8900 will automatically switch to Receive-DMA mode if the conditions specified in Section 5.5 are met. RxDMAonly (Bit 9) has precedence over AutoRxDMAE.
B	BufferCRC	When set, the received CRC is included with the data stored in the receive-frame buffer, and the four CRC bytes are included in the receive-frame length (PacketPage base + 0402h). When clear, neither the receive buffer nor the receive length include the CRC.
C	CRCerroriE	When set, there is a CRCerror Interrupt if a frame is received with a bad CRC.
D	RuntiE	When set, there is a Runt Interrupt if a frame is received that is shorter than 64 bytes. The CS8900 always discards any frame that is shorter than 8 bytes.
E	ExtradataiE	When set, there is an Extradata Interrupt if a frame is received that is longer than 1518 bytes. The operation of this bit is independent of the received packet integrity (good or bad CRC).

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

0000 0000	0000 0011

Register 4: Receiver Event (RxEvent, Read-only)　　　　*Address: PacketPage base + 0124h*

F	E	D	C	B	A	9	8	7	6	5-0
	Extradata	Runt	CRCerror	Broadcast	Individualadr	Hashed	RxOK	Dribblebits	IAHash	000100

Alternate meaning if bits 8 and 9 are both set (see Section 5.3 for exception regarding Broadcast frames).

F	E	D	C	B	A	9	8	7	6	5-0
Hash Table Index (see Section 5.3)						Hashed = 1	RxOK= 1	Dribblebits	IAHash	000100

RxEvent reports the status of the current received frame. All RxEvent bits are cleared upon readout. The host is responsible for processing all event bits. RxStatus register (PacketPage base + 0400h) is the same as the RxEvent register except RxStatus is not cleared when RxEvent is read. See Section 5.2.

BIT	NAME	DESCRIPTION
5-0	000100	These bits identify this as the Receiver Event Register. When reading this register, these bits will be 000100, where the LSB corresponds to Bit 0.
6	IAHash	If the received frame's Destination Address is accepted by the hash filter, then this bit is set if, and only if, RxOK (Bit 8) is set, IAHashA (Register 5, RxCTL, Bit 6) is set, and Hashed (Bit 9) is set. See Section 5.3.
7	Dribblebits	If set, the received frame had from one to seven bits after the last received full byte. An "Alignment Error" occurs when Dribblebits and CRCerror (Bit C) are both set.
8	RxOK	If set, the received frame had a good CRC and valid length (i.e., there is not a CRC error, Runt error, or Extradata error). When RxOK is set, then the length of the received frame is contained at PacketPage base + 0402h. If RxOKiE (Register 3, RxCFG, Bit 8) is set, there is an interrupt.
9	Hashed	If set, the received frame had a Destination Address that was accepted by the hash filter. If Hashed *and* RxOK (Bit 8) are set, Bits F through A of RxEvent become the Hash Table Index for this frame [See Section 5.3 for an exception regarding broadcast frames!]. If Hashed *and* RxOK are not both set, then Bits F through A are individual event bits as defined below.
A	IndividualAdr	If the received frame had a Destination Address which matched the Individual Address found at PacketPage base + 0158h, then this bit is set if, and only if, RxOK (Bit 8) is set *and* IndividualA (Register 5, RxCTL, Bit A) is set.
B	Broadcast	If the received frame had a Broadcast Address (FFFF FFFF FFFFh) as the Destination Address, then this bit is set if, and only if, RxOK is set *and* BroadcastA (Register 5, RxCTL, Bit B) is set.
C	CRCerror	If set, the received frame had a bad CRC. If CRCerroriE (Register 3, RxCFG, Bit C) is set, there is an interrupt.
D	Runt	If set, the received frame was shorter than 64 bytes. If RuntiE (Register 3, RxCFG, Bit D) is set, there is an interrupt.
E	Extradata	If set, the received frame was longer than 1518 bytes. All bytes beyond 1518 are discarded. If ExtradataiE (Register 3, RxCFG, Bit E) is set, there is an interrupt.

This register's initial state after reset is:　　| 0000 0000 | 0000 0100 |

Register 5: Receiver Control (RxCTL, Read/Write)   *Address: PacketPage base + 0104h*

F	E	D	C	B	A	9	8	7	6	5-0
	ExtradataA	RuntA	CRCerrorA	Broad castA	Individ ualA	MulticastA	RxOKA	Promis cuousA	IAHashA	000101

RxCTL has two functions: Bits 8, C, D, and E define what types of frames to accept. Bits 6, 7, 9, A, and B configure the Destination Address filter. See Section 5.3.

BIT	NAME	DESCRIPTION
5-0	000101	These bits provide an internal address used by the CS8900 to identify this as the Receiver Control Register. To write to this register, these bits must be 000101, where the LSB corresponds to Bit 0.
		For a received frame to be accepted, the Destination Address of that frame must pass the filter criteria found in Bits 6, 7, 9, A, and B (see Section 5.3).
6	IAHashA	When set, receive frames are accepted when the Destination Address is an Individual Address that passes the hash filter.
7	PromiscuousA	Frames with any address are accepted when this bit is set.
8	RxOKA	When set, the CS8900 accepts frames with correct CRC and valid length (valid length is: 64 bytes <= length <= 1518 bytes).
9	MulticastA	When set, receive frames are accepted if the Destination Address is an Multicast Address that passes the hash filter.
A	IndividualA	When set, receive frames are accepted if the Destination Address matches the Individual Address found at PacketPage base + 0158h to PacketPage base + 015Dh.
B	BroadcastA	When set, receive frames are accepted if the Destination Address is FFFF FFFF FFFFh.
C	CRCerrorA	When set, receive frames that pass the Destination Address filter, but have a bad CRC, are accepted. When clear, frames with bad CRC are discarded. See Note 1.
D	RuntA	When set, receive frames that are smaller than 64 bytes, and that pass the Destination Address filter are accepted. When clear, received frames less that 64 bytes in length are discarded. The CS8900 discards any frame that is less than 8 bytes. See Note 1.
E	ExtradataA	When set, receive frames longer than 1518 bytes and that pass the Destination Address filter are accepted. The CS8900 accepts only the first 1518 bytes and ignores the rest. When clear, frames longer than 1518 bytes are discarded. See Note 1.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

0000 0000	0000 0101

NOTE:
1. Typically, when bits CRCerrorA, RuntA and ExtradataA are cleared (meaning bad frames are being discarded), then the corresponding bits CRCerroriE, RuntiE and ExtradataiE should be set in register 3 (Receiver Configuration register) to allow the device driver to keep track of discarded frames.

Register 7: Transmit Configuration (TxCFG, Read/Write)   *Address: PacketPage base + 0106h*

F	E	D	C	B	A	9	8	7	6	5-0
16colliE				AnycolliE	JabberiE	Out-of-windowiE	TxOKiE	SQE erroriE	Loss-of-CRSiE	000111

Each bit in TxCFG is an interrupt enable. When set, the interrupt is enabled as described below. When clear, there is no interrupt.

BIT	NAME	DESCRIPTION

5-0   000111   These bits provide an internal address used by the CS8900 to identify this as the Transmit Configuration Register. To write to this register, these bits must be 000111, where the LSB corresponds to Bit 0.

6   Loss-of-CRSiE   If the CS8900 starts transmitting on the AUI and does not see the Carrier Sense signal at the end of the preamble, an interrupt is generated if this bit is set. Carrier Sense activity is reported by the CRS bit (Register 14, LineST, Bit E).

7   SQEerroriE   When set, an interrupt is generated if there is an SQE error. (At the end of a transmission on the AUI, the CS8900 expects to see a collision within 64 bit times. If this does not happen, there is an SQE error.)

8   TxOKiE   When set, an interrupt is generated if a packet is completely transmitted.

9   Out-of-windowiE   When set, an interrupt is generated if a late collision occurs (a late collision is a collision which occurs after the first 512 bit times). When this occurs, the CS8900 forces a bad CRC and terminates the transmission.

A   JabberiE   When set, an interrupt is generated if a transmission is longer than approximately 26 ms.

B   AnycolliE   When set, if one or more collisions occur during the transmission of a packet, an interrupt occurs at the end of the transmission.

F   16colliE   If the CS8900 encounters 16 normal collisions while attempting to transmit a particular packet, the CS8900 stops attempting to transmit that packet. When this bit is set, there is an interrupt upon detecting the 16th collision.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

0000 0000	0000 0111

NOTE:
      Bit 8 (TxOKiE) and Bit B (AnycolliE) are interrupts for normal transmit operation. Bits 6, 7, 9, A, and F are interrupts for abnormal transmit operation.

Register 8: Transmitter Event (TxEvent, Read-only)   *Address: PacketPage base + 0128h*

F	E	D	C	B	A	9	8	7	6	5-0
16coll	Number-of-Tx-collisions				Jabber	Out-of-window	TxOK	SQEerror	Loss-of-CRS	001000

TxEvent gives the event status of the last packet transmitted.

BIT	NAME	DESCRIPTION
5-0	001000	These bits provide an internal address used by the CS8900 to identify this as the Transmitter Event Register. When reading this register, these bits will be 001000, where the LSB corresponds to Bit 0.
6	Loss-of-CRS	If the CS8900 is transmitting on the AUI and doesn't see Carrier Sense (CRS) at the end of the preamble, there is a Loss-of-Carrier error and this bit is set. If Loss-of-CRSiE (Register 7, TxCFG, Bit 6) is set, there is an interrupt.
7	SQEerror	At the end of a transmission on the AUI, the CS8900 expects to see a collision within 64 bit times. If this does not happen, there is an SQE error and this bit is set. If SQEerroriE (Register 7, TxCFG, Bit 7) is set, there is an interrupt.
8	TxOK	This bit is set if the last packet was completely transmitted (Jabber (Bit A), out-of-window-collision (Bit 9), and 16Coll (Bit F) must all be clear). If TxOKiE (Register 7, TxCFG, Bit 8) is set, there is an interrupt.
9	Out-of-window	This bit is set if a collision occurs more than 512 bit times after the first bit of the preamble. When this occurs, the CS8900 forces a bad CRC and terminates the transmission. If Out-of-windowiE (Register 7, TxCFG, Bit 9) is set, there is an interrupt.
A	Jabber	If the last transmission is longer than 26 msec, then the packet output is terminated by the jabber logic and this bit is set. If JabberiE (Register 7, TxCFG, Bit A) is set, there is an interrupt.
E-B	Number-of-Tx-collisions	These bits give the number of transmit collisions that occurred on the last transmitted packet. Bit B is the LSB. If AnycolliE (Register 7, TxCFG, Bit B) is set, there is an interrupt when any collision occurs.
F	16coll	This bit is set if the CS8900 encounters 16 normal collisions while attempting to transmit a particular packet. When this happens, the CS8900 stops further attempts to send that packet. If 16colliE (Register 7, TxCFG, Bit F) is set, there is an interrupt.

This register's initial state after reset is: | 0000 0000 | 0000 1000 |

NOTES:
1. In any event register, like TxEvent, all bits are cleared upon readout. The host is responsible for processing all event bits.
2. TxOK (Bit 8) and the Number-of-Tx-Collisions (Bits E-B) are used in normal packet transmission. All other bits (6, 7, 9, A, and F) give the status of abnormal transmit operation.

Register 9: Transmit Command Status (TxCMD, Read-only)   **Address: PacketPage base + 0108h**

F	E	D	C	B	A	9	8	7	6	5-0
		TxPadDis	InhibitCRC			Onecoll	Force	TxStart		001001

This register contains the latest transmit command which tells the CS8900 how the next packet should be sent. The command must be written to PacketPage base + 0144h in order to initiate a transmission. The host can read the command from register 9 (PacketPage base + 0108h). See Section 5.7.

| BIT | NAME | DESCRIPTION |

5-0   001001   These bits provide an internal address used by the CS8900 to identify this as the Transmit Command Register. When reading this register, these bits will be 001001, where the LSB corresponds to Bit 0.

7, 6   TxStart   This pair of bits determines how many bytes are transferred to the CS8900 before the MAC starts the packet transmit process.

    Bit 7 Bit 6
      0   0   Start transmission after 5 bytes are in the CS8900
      0   1   Start transmission after 381 bytes are in the CS8900
      1   0   Start transmission after 1021 bytes are in the CS8900
      1   1   Start transmission after the entire frame is in the CS8900

8   Force   When set in conjunction with a new transmit command, any transmit frames waiting in the transmit buffer are deleted. If a previous packet has started transmission, that packet is terminated within 64 bit times with a bad CRC.

9   Onecoll   When this bit is set, any transmission will be terminated after only one collision. When clear, the CS8900 allows up to 16 normal collisions before terminating the transmission.

C   InhibitCRC   When set, the CRC is not appended to the transmission.

D   TxPadDis   When TxPadDis is clear, if the host gives a transmit length less than 60 bytes and InhibitCRC is set, then the CS8900 pads to 60 bytes. If the host gives a transmit length less than 60 bytes and InhibitCRC is clear, then the CS8900 pads to 60 bytes and appends the CRC.

                When TxPadDis is set, the CS8900 allows the transmission of runt frames (a frame less than 64 bytes). If InhibitCRC is clear, the CS8900 appends the CRC. If InhibitCRC is set, the CS8900 does not append the CRC.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

| 0000 0000 | 0000 1001 |

NOTE: The CS8900 does not transmit a frame if TxLength < 3.

Register B: Buffer Configuration (BufCFG, Read/Write)   *Address: PacketPage base + 010Ah*

F	E	D	C	B	A	9	8	7	6	5-0
RxDestiE		Miss OvfloiE	TxCol OvfloiE	Rx128iE	RxMissiE	TxUnder runiE	Rdy4TxiE	RxDMAiE	SWint-X	001011

Each bit in BufCFG is an interrupt enable. When set, the interrupt described below is enabled. When clear, there is no interrupt.

BIT	NAME	DESCRIPTION

5-0   001011   These bits provide an internal address used by the CS8900 to identify this as the Buffer Configuration Register. To write to this register, these bits must be 001011, where the LSB corresponds to Bit 0.

6   SWint-X   When set, there is an interrupt requested by the host software. The CS8900 provides the interrupt, and sets the SWint (Register C, BufEvent, Bit 6) bit. The CS8900 acts upon this command at once. SWint-X is an Act-Once bit. To generate another interrupt, re-write a "1" to this bit.

7   RxDMAiE   When set, there is an interrupt when a frame has been received *and* DMA is complete. With this interrupt, the RxDMAFrame bit (Register C, BufEvent, Bit 7) is set.

8   Rdy4TxiE   When set, there is an interrupt when the CS8900 is ready to accept a frame from the host for transmission. (See Section 5.7 for a description of the transmit bid process.)

9   TxUnderruniE   When set, there is an interrupt if the CS8900 runs out of data before it reaches the end of the frame (called a transmit underrun). When this happens, event bit TXUnderrun (Register C, BufEvent, Bit 9) is set and the CS8900 makes no further attempts to transmit that frame. If the host still wants to transmit that particular frame, the host must go through the transmit request process again.

A   RxMissiE   When set, there is an interrupt if one or more received frames is lost due to slow movement of receive data out of the receive buffer (called a receive miss). When this happens, the RxMiss bit (Register C, BufEvent, Bit A) is set.

B   Rx128iE   When set, there is an interrupt after the first 128 bytes of a frame have been received. This allows a host processor to examine the Destination Address, Source Address, Length, Sequence Number, and other information before the entire frame is received. This interrupt should not be used with DMA. Thus, if either AutoRxDMA (Register 3, RxCFG, Bit A) or RxDMAonly (Register 3, RxCFG, Bit 9) is set, the Rx128iE bit must be clear.

C   TxColOvfiE   If set, there is an interrupt when the TxCOL counter increments from 1FFh to 200h. (The TxCOL counter (Register 18) is incremented whenever the CS8900 sees that the RXD+/RXD- pins (10BASE-T) or the CI+/CI- pins (AUI) go active while a packet is being transmitted.)

Register B: Buffer Configuration (BufCFG) continued

| BIT | NAME | DESCRIPTION |

D MissOvfloiE If MissOvfloiE is set, there is an interrupt when the RxMISS counter increments from 1FFh to 200h. (A receive miss is said to have occurred if packets are lost due to slow movement of receive data out of the receive buffers. When this happens, the RxMiss bit (Register C, BufEvent, Bit A) is set, and the RxMISS counter (Register 10) is incremented.)

F RxDestiE When set, there is an interrupt when a receive frame passes the Destination Address filter criteria defined in the RxCTL register (Register 5). This bit provides an early indication of an incoming frame. It is earlier than Rx128 (Register C, BufEvent, Bit B).

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state after reset. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

| 0000 0000 | 0000 1011 |

Register C: Buffer Event (BufEvent, Read-only)   *Address: PacketPage base + 012Ch*

F	E	D	C	B	A	9	8	7	6	5-0
RxDest				Rx128	RxMiss	TxUnder run	Rdy4Tx	RxDMA Frame	SWint	001100

BufEvent gives the status of the transmit and receive buffers.

BIT	NAME	DESCRIPTION
5-0	001100	These bits provide an internal address used by the CS8900 to identify this as the Buffer Event Register. When reading this register, these bits will be 001100, where the LSB corresponds to Bit 0.
6	SWint	If set, there has been a software initiated interrupt. This bit is used in conjunction with the SWint-X bit (Register B, BufCFG, Bit 6).
7	RxDMAFrame	If set, one or more received frames have been transferred by slave DMA. If RxDMAiE (Register B, BufCFG, Bit 7) is set, there is an interrupt.
8	Rdy4Tx	If set, the CS8900 is ready to accept a frame from the host for transmission. If Rdy4TxiE (Register B, BufCFG, Bit 8) is set, there is an interrupt. (See Section 5.7 for a description of the transmit bid process.)
9	TxUnderrun	This bit is set if CS8900 runs out of data before it reaches the end of the frame (called a transmit underrun). If TxUnderruniE (Register B, BufCFG, Bit 9) is set, there is an interrupt.
A	RxMiss	If set, one or more receive frames have been lost due to slow movement of data out of the receive buffers. If RxMissiE (Register B, BufCFG, Bit A) is set, there is an interrupt.
B	Rx128	This bit is set after the first 128 bytes of an incoming frame have been received. This bit will allow the host the option of pre-processing frame data before the entire frame is received. If Rx128iE (Register B, BufCFG, Bit B) is set, there is an interrupt.
F	RxDest	When set, this bit shows that a receive frame has passed the Destination Address Filter criteria as defined in the RxCTL register (Register 5). This bit is useful as an early indication of an incoming frame. It will be earlier than Rx128 (Register C, BufEvent, Bit B). If RxDestiE (Register B, BufCFG, Bit F) is set, there is an interrupt.

This register's initial state after reset is: | 0000 0000 | 0000 1100 |

NOTE:

With any event register, like BufEvent, all bits are cleared upon readout. The host is responsible for processing all event bits.

Register 10: Receiver Miss Counter (RxMISS, Read-only)   *Address: PacketPage base + 0130h*

F-6	5-0
MissCount	010000

The RxMISS counter (Bits 6 through F) records the number of receive frames that are lost (missed) due to the lack of available buffer space. If the MissOvfloiE bit (Register B, BufCFG, Bit D) is set, there is an interrupt when RxMISS increments from 1FFh to 200h. This interrupt provides the host with an early warning that the RxMISS counter should be read before it reaches 3FFh and starts over (by interrupting at 200h, the host has an additional 512 counts before RxMISS actually overflows). The RxMISS counter is cleared when read.

BIT   NAME   DESCRIPTION

5-0   010000   These bits provide an internal address used by the CS8900 to identify this as the Bus Status Register. When reading this register, these bits will be 010000, where the LSB corresponds to Bit 0.

F-6   MissCount   The upper ten bits contain the number of missed frames.

This register's initial state after reset is: | 0000 0000 | 0001 0000 |

Register 12: Transmit Collision Counter (TxCOL, Read-only)   *Address: PacketPage base + 0132h*

F-6	5-0
ColCount	010010

The TxCOL counter (Bits 6 through F) is incremented whenever the 10BASE-T Receive Pair (RXD+ / RXD-) or AUI Collision Pair (CI+ / CI-) becomes active while a packet is being transmitted. If the TxColOvfiE bit (Register B, BufCFG, Bit C) is set, there is an interrupt when TxCOL increments from 1FFh to 200h. This interrupt provides the host with an early warning that the TxCOL counter should be read before it reaches 3FFh and starts over (by interrupting at 200h, the host has an additional 512 counts before TxCOL actually overflows). The TxCOL counter is cleared when read.

BIT   NAME   DESCRIPTION

5-0   010010   These bits provide an internal address used by the CS8900 to identify this as the Bus Status Register. When reading this register, these bits will be 010010, where the LSB corresponds to Bit 0.

F-6   ColCount   The upper ten bits contain the number of collisions.

This register's initial state after reset is: | 0000 0000 | 0001 0010 |

Register 13: Line Control (LineCTL, Read/Write)   ***Address: PacketPage base + 0112h***

F	E	D	C	B	A	9	8	7	6	5-0
	LoRx Squelch	2-part DefDis	PolarityDis	Mod BackoffE		Auto AUI/10BT	AUIonly	SerTxON	SerRxON	010011

LineCTL determines the configuration of the MAC engine and physical interface.

BIT	NAME	DESCRIPTION

5-0   010011   These bits provide an internal address used by the CS8900 to identify this as the Line Control Register. To write to this register, these bits must be 010011, where the LSB corresponds to Bit 0.

6   SerRxON   When set, the receiver is enabled. When clear, no incoming packets pass through the receiver. If SerRxON is cleared while a packet is being received, reception is completed and no subsequent receive packets are allowed until SerRxON is set again.

7   SerTxON   When set, the transmitter is enabled. When clear, no transmissions are allowed. If SerTxON is cleared while a packet is being transmitted, transmission is completed and no subsequent packets are transmitted until SerTxON is set again.

8   AUIonly   Bits 8 and 9 are used to select either the AUI or the 10BASE-T interface according to the following: [Note: 10BASE-T transmitter will be inactive even when selected unless link pulses are detected or bit DisableLT (register 19) is set.]

AUIonly (Bit 8)	AutoAUI/10BT (Bit 9)	Physical Interface
1	N/A	AUI
0	0	10BASE-T
0	1	Auto-Select

9   AutoAUI/10BT   See AUIonly (Bit 8) description above.

B   ModBackoffE   When clear, the ISO/IEC standard backoff algorithm is used (see Section 3.9). When set, the Modified Backoff algorithm is used. (The Modified Backoff algorithm extends the backoff delay after each of the first three Tx collisions.)

C   PolarityDis   The 10BASE-T receiver automatically determines the polarity of the received signal at the RXD+/RXD- input (see Section 3.11). When this bit is clear, the polarity is corrected, if necessary. When set, no effort is made to correct the polarity. This bit is independent of the PolarityOK bit (Register 14, LineST, Bit C), which reports whether the polarity is normal or reversed.

D   2-partDefDis   Before a transmission can begin, the CS8900 follows a deferral procedure. With the 2-partDefDis bit clear, the CS8900 uses the standard two-part deferral as defined in ISO/IEC 8802-3 paragraph 4.2.3.2.1. With the 2-partDefDis bit set, the two-part deferral is disabled.

E   LoRxSquelch   When clear, the 10BASE-T receiver squelch thresholds are set to levels defined by the ISO/IEC 8802-3 specification. When set, the thresholds are reduced by approximately 6 dB. This is useful for operating with "quiet" cables that are longer than 100 meters.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

0000 0000	0001 0011

Register 14: Line Status (LineST, Read-only)   *Address: PacketPage base + 0134h*

F	E	D	C	B	A	9	8	7	6	5-0
	CRS		PolarityOK			10BT	AUI	LinkOK		010100

LineST reports the status of the Ethernet physical interface.

BIT   NAME   DESCRIPTION

5-0   010100   These bits provide an internal address used by the CS8900 to identify this as the Line Status Register. When reading this register, these bits will be 010100, where the LSB corresponds to Bit 0.

7   LinkOK   If set, the 10BASE-T link has not failed. When clear, the link has failed, either because the CS8900 has just come out of reset, or because the receiver has not detected any activity (link pulses or received packets) for at least 50 ms.

8   AUI   If set, the CS8900 is using the AUI.

9   10BT   If set, the CS8900 is using the 10BASE-T interface.

C   PolarityOK   If set, the polarity of the 10BASE-T receive signal (at the RXD+ / RXD- inputs) is correct. If clear, the polarity is reversed. If PolarityDis (Register 19, TestCTL, Bit C) is set, then the polarity is automatically corrected, if needed. The PolarityOK status bit shows the true state of the incoming polarity independent of the PolarityDis control bit. Thus, if PolarityDis is clear and PolarityOK is clear, then the receive polarity is inverted, and corrected.

E   CRS   This bit tells the host the status of an incoming frame. If CRS is set, a frame is currently being received. CRS remains asserted until the end of frame (EOF). At EOF, CRS goes inactive in about 1.3 to 2.3 bit times after the last low-to-high transition of the recovered data.

This register's initial state after reset is: 0X0X 00XX | X001 0100

Register 15: Self Control (SelfCTL, Read/Write)   ***Address: PacketPage base + 0114h***

F	E	D	C	B	A	9	8	7	6	5-0
HCB1	HCB0	HC1E	HC0E		HW StandbyE	HWSleepE	SW Suspend		RESET	010101

SelfCTL controls the operation of the LED outputs and the low-power modes.

BIT	NAME	DESCRIPTION
5-0	010101	These bits provide an internal address used by the CS8900 to identify this as the Chip Self Control Register. To write to this register, these bits must be 010101, where the LSB corresponds to Bit 0.
6	RESET	When set, a chip-wide reset is initiated immediately. RESET is an Act-Once bit. This bit is cleared as a result of the reset.
8	SWSuspend	When set, the CS8900 enters the software initiated Suspend mode. Upon entering this mode, there is a partial reset. All registers and circuits are reset except for the ISA I/O Base Address Register and the SelfCTL Register. There is no transmit nor receive activity in this mode. To come out of software Suspend, the host issues an I/O Write within the CS8900's assigned I/O space (see Section 3.7 for a complete description of the CS8900's low-power modes).
9	HWSleepE	When set, the $\overline{\text{SLEEP}}$ input pin is enabled. If $\overline{\text{SLEEP}}$ is high, the CS8900 is "awake", or operative (unless in SWSuspend mode, as shown above). If $\overline{\text{SLEEP}}$ is low, the CS8900 enters either the Hardware Standby or Hardware Suspend mode. When clear, the CS8900 ignores the $\overline{\text{SLEEP}}$ input pin (see Section 3.7 for a complete description of the CS8900's low-power modes).
A	HWStandbyE	If HWSleepE is set *and* the $\overline{\text{SLEEP}}$ input pin is low, then when HWStandbyE is set, the CS8900 enters the Hardware Standby mode. When clear, the CS8900 enters the Hardware Suspend mode (see Section 3.7 for a complete description of the CS8900's low-power modes).
C	HC0E	The $\overline{\text{LINKLED}}$ or $\overline{\text{HC0}}$ output pin is selected with this control bit. When HC0E is clear, the output pin is $\overline{\text{LINKLED}}$. When HC0E is set, the output pin is $\overline{\text{HC0}}$ and the HCB0 bit (Bit E) controls the pin.
D	HC1E	The $\overline{\text{BSTATUS}}$ or $\overline{\text{HC1}}$ output pin is selected with this control bit. When HC1E is clear, the output pin is $\overline{\text{BSTATUS}}$ and indicates receiver ISA Bus activity. When HC1E is set, the output pin is $\overline{\text{HC1}}$ and the HCB1 bit (Bit F) controls the pin.
E	HCB0	When HC0E (Bit C) is set, this bit controls the $\overline{\text{HC0}}$ pin. If HCB0 is set, $\overline{\text{HC0}}$ is low. If HCB0 is clear, $\overline{\text{HC0}}$ is high. $\overline{\text{HC0}}$ may drive an LED or a logic gate. When HC0E (Bit C) is clear, this control bit is ignored.
F	HCB1	When HC1E (Bit D) is set, this bit controls the $\overline{\text{HC1}}$ pin. If HCB1 is set, $\overline{\text{HC1}}$ is low. If HCB1 is clear, $\overline{\text{HC1}}$ is high. $\overline{\text{HC1}}$ may drive an LED or a logic gate. When HC1E (Bit D) is clear, this control bit is ignored.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

0000 0000	0001 0101

Register 16: Self Status (SelfST, Read-only)   *Address: PacketPage base + 0136h*

F	E	D	C	B	A	9	8	7	6	5-0
			EEsize	ELpresent	EEPROM OK	EEPROM present	SIBUSY	INITD	3.3V Active	010110

SelfST reports the status of the EEPROM interface and the initialization process.

BIT	NAME	DESCRIPTION
5-0	010110	These bits provide an internal address used by the CS8900 to identify this as the Chip Self Status Register. When reading this register, these bits will be 010110, where the LSB corresponds to Bit 0.
6	3.3VActive	If the CS8900 is operating on a 3.3V supply, this bit is set. If the CS8900 is operating on a 5V supply, this bit is clear.
7	INITD	If set, the CS8900 initialization, including read-in of the EEPROM, is complete.
8	SIBUSY	If set, the EECS output pin is high indicating that the EEPROM is currently being read or programmed. The host must not write to PacketPage base + 0040h nor 0042h until SIBUSY is clear.
9	EEPROM present	If the EEDataIn pin is low after reset, there is no EEPROM present, and the EEPROMpresent bit is clear. If the EEDataIn pin is high after reset, the CS8900 "assumes" that an EEPROM is present, and this bit is set.
A	EEPROMOK	If set, the checksum of the EEPROM readout was OK.
B	ELpresent	If set, external logic for Latchable Address bus decode is present.
C	EEsize	This bit shows the size of the attached EEPROM and is valid only if the EEPROMpresent bit (Bit 9) *and* EEPROMOK bit (Bit A) are both set. If clear, the EEPROM size is either 128 words ('C56 or 'CS56) or 256 words (C66 or 'CS66). If set, the EEPROM size is 64 words ('C46 or 'CS46).

This register's initial state after reset is: (X = Depends on Configuration.) | 000X XXXX | XX01 0110 |

Register 17: Bus Control (BusCTL, Read/Write)   **Address: PacketPage base + 0116h**

F	E	D	C	B	A	9	8	7	6	5-0
EnableIRQ		RxDMA size	IOCH RDYE	DMABurst	MemoryE	UseSA			Reset RxDMA	010111

BusCTL controls the operation of the ISA-bus interface.

BIT	NAME	DESCRIPTION
5-0	010111	These bits provide an internal address used by the CS8900 to identify this as the Bus Control Register. To write to this register, these bits must be 010111, where the LSB corresponds to Bit 0.
6	ResetRxDMA	When set, the RxDMA offset pointer at PacketPage base + 0026h is reset to zero. When the host sets this bit, the CS8900 does the following: 1. Terminates the current receive DMA activity, if any. 2. Clears all internal receive buffers. 3. Zeroes the RxDMA offset pointer. The CS8900 acts upon this command only once when this bit is set. ResetRxDMA is an Act-Once bit. To cause the pointer to reset again, the host must re-write a "1".
9	UseSA	When set, the $\overline{\text{MEMCS16}}$ pin goes low whenever the address on the SA bus [13..19] match the CS8900's assigned Memory base address *and* the $\overline{\text{CHIPSEL}}$ pin is low (internal address decode). When clear, $\overline{\text{MEMCS16}}$ is driven low whenever $\overline{\text{CHIPSEL}}$ goes low. (external address decode). see Section 4.9. For $\overline{\text{MEMCS16}}$ pin to be enabled, the CS8900 must be in Memory Mode with the MemoryE bit (Register 17, BusCTL, Bit A) set.
A	MemoryE	When set, the CS8900 may operate in Memory Mode. When clear, Memory Mode is disabled. I/O Mode is always enabled.
B	DMABurst	When clear, the CS8900 performs continuous DMA until the receive frame is completely transferred from the CS8900 to host memory. When set, each DMA access is limited to 28 µs, after which time the CS8900 gives up the bus for 1.3 µs before making a new DMA request.
C	IOCHRDYE	When set, the CS8900 does not use the IOCHRDY output pin, and the pin is always high-impedance. This allows external pull-up to force the output high. When clear, the CS8900 drives IOCHRDY low to request additional time during I/O Read and Memory Read cycles. IOCHRDY does not affect I/O Write, Memory Write, nor DMA Read.
D	RxDMAsize	This bit determines the size of the receive DMA buffer (located in host memory). When set, the DMA buffer size is 64 Kbytes. When clear, it is 16 Kbytes.
F	EnableIRQ	When set, the CS8900 will generate an interrupt in response to an interrupt event (Section 5.1). When cleared, the CS8900 will not generate any interrupts.

After reset, if no EEPROM is found by the CS8900, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 3.3.

0000 0000	0001 0111

Register 18: Bus Status (BusST, Read-only)   *Address: PacketPage base + 0138h*

F	E	D	C	B	A	9	8	7	6	5-0
							Rdy4Tx NOW	TxBidErr		011000

BusST describes the status of the current transmit operation.

BIT	NAME	DESCRIPTION
5-0	011000	These bits provide an internal address used by the CS8900 to identify this as the Bus Status Register. When reading this register, these bits will be 011000, where the LSB corresponds to Bit 0.
7	TxBidErr	If set, the host has commanded the CS8900 to transmit a frame that the CS8900 will not send. Frames that the CS8900 will not send are:

       1)   Any frame greater than 1514 bytes, provided that InhibitCRC (Register 9, TxCMD, Bit C) is clear.

       2)   Any frame greater than 1518 bytes.

Note that this bit is not set when transmit frames are too short.

| 8 | Rdy4TxNOW | Rdy4TxNOW signals the host that the CS8900 is ready to accept a frame from the host for transmission. This bit is similar to Rdy4Tx (Register C, BufEvent, Bit 8) except that there is no interrupt associated with Rdy4TxNOW. The host can poll the CS8900 and check Rdy4TxNOW to determine if the CS8900 is ready for transmit. (See Section 5.7 for a description of the transmit bid process.) |

This register's initial state after reset is: | 0000 0000 | XX01 1000 |

F-6	5-0
AUI_Delay	011100

The TDR counter (Bits 6 through F) is a time domain reflectometer useful in locating cable faults in 10BASE-2 and 10BASE-5 coax networks. It counts at a 10 MHz rate from the beginning of transmission on the AUI to when a collision or Loss-of-Carrier error occurs. The TDR counter is cleared when read.

BIT	NAME	DESCRIPTION
5-0	011100	These bits provide an internal address used by the CS8900 to identify this as the Bus Status Register. When reading this register, these bits will be 011100, where the LSB corresponds to Bit 0.
F-6	AUI-Delay	The upper ten bits contains the number of 10 MHz clock periods between the beginning of transmission on the AUI to when a collision or Loss-of-Carrier error occurs.

This register's initial state after reset is: | 0000 0000 | 0001 1100 |

4.5   Initiate Transmit Registers

Initiate Transmit Register
Transmit Command Request - TxCMD (Write-only)   *Address: PacketPage base + 0144h*

F	E	D	C	B	A	9	8	7	6	5-0
		TxPadDis	InhibitCRC			Onecoll	Force	TxStart		001001

The word written to PacketPage base + 0144h tells the CS8900 how the next packet should be transmitted. This PacketPage location is write-only, and the written word can be read from Register 9, at PacketPage base + 0108h. The CS8900 does not transmit a frame if TxLength (at PacketPage location base + 0146h) is less than 3. See Section 5.7.

BIT	NAME	DESCRIPTION

5-0   001001   These bits provide an internal address used by the CS8900 to identify this as the Transmit Command Register. When reading this register, these bits will be 001001, where the LSB corresponds to Bit 0.

7, 6   TxStart   This pair of bits determines how many bytes are transferred to the CS8900 before the MAC starts the packet transmit process.

     Bit 7 Bit 6
       0   0   Start transmission after 5 bytes are in the CS8900
       0   1   Start transmission after 381 bytes are in the CS8900
       1   0   Start transmission after 1021 bytes are in the CS8900
       1   1   Start transmission after the entire frame is in the CS8900

8   Force   When set in conjunction with a new transmit command, any transmit frames waiting in the transmit buffer are deleted. If a previous packet has started transmission, that packet is terminated within 64 bit times with a bad CRC.

9   Onecoll   When this bit is set, any transmission will be terminated after only one collision. When clear, the CS8900 allows up to 16 normal collisions before terminating the transmission.

C   InhibitCRC   When set, the CRC is not appended to the transmission.

D   TxPadDis   When TxPadDis is clear, if the host gives a transmit length less than 60 bytes and InhibitCRC is set, then the CS8900 pads to 60 bytes. If the host gives a transmit length less than 60 bytes and InhibitCRC is clear, then the CS8900 pads to 60 bytes and appends the CRC.

    When TxPadDis is set, the CS8900 allows the transmission of runt frames (a frame less than 64 bytes). If InhibitCRC is clear, the CS8900 appends the CRC. If InhibitCRC is set, the CS8900 does not append the CRC.

Since this register is write-only, it's initial state after reset is undefined.

Initiate Transmit Register:
Transmit Length (Write-only)   *Address: PacketPage base + 0146h*

Address 0147h	Address 0146h
Most-significant byte of Transmit Frame Length.	Least-significant byte of Transmit Frame Length.

This register is used in conjunction with register 9, TxCMD. When a transmission is initiated via a command in TxCMD, the the length of the transmitted frame is written into this register. The length of the transmitted frame may be modified by the configuration of the TxPadDis and InhibitCRC bits in the TxCMD register. See Table 5.17, and Section 5.7. TxLength must be >3 and < 1519.

Since this register is write-only, it's initial state after reset is undefined.

4.6 Address Filter Registers

Address Filter Register:
Logical Address Filter (hash table) (Read//Write)   *Address: PacketPage base + 0150h*

Address 0157h	Address 0156h	Address 0155h	Address 0154h	Address 0153h	Address 0152h	Address 0151h	Address 0150h
Most-significant byte of hash filter.							Least-significant byte of hash filter.

The CS8900 hashing decoder circuitry compares its output with one bit of the Logical Address Filter Register. If the decoder output and the Logical Address Filter bit match, the frame passes the hash filter and the Hashed bit (Register 4, RxEvent, Bit 9) is set. See Section 5.3.

This register's initial state after reset is:

0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000

Address Filter Register:
Individual Address (IEEE address) (Read//Write)   *Address: PacketPage base + 0158h*

Address 0015Dh	Address 0015Ch	Address 0015Bh	Address 0015Ah	Address 0159h	Address 00158h
High-order byte of IA					Low-order byte of IA.

The unique, IEEE 48-bit Individual Address (IA) begins at 0158h. The first bit of the IA (Bit IA[00]) must be "0". See Section 5.3. The bit ordering for the IA is:

PacketPage Address	Most-Significant Bit of IA in this byte	Most-Significant Bit of IA in this byte
0158h	IA[47]	IA[40]
0159h	IA[39]	IA[32]
015Ah	IA[31]	IA[24]
015Bh	IA[23]	IA[16]
015Ch	IA[15]	IA[08]
015Dh	IA[07]	IA[00]

The value of this register must be loaded from external storage, for example, from the EEPROM. See Section 3.3. If the CS8900 is not able to load the IA from the EEPROM, then after a reset this register is undefined, and the driver must write an address to this register.

4.7 Receive and Transmit Frame Locations

The Receive and Transmit Frame PacketPage locations are used to transfer Ethernet frames to and from the host. The host sequentially writes to and reads from these locations, and internal buffer memory is dynamically allocated between transmit and receive as needed. One receive frame and one transmit frame are accessible at a time.

Receive PacketPage Locations

In IO mode, the receive status/length/frame locations are read through repetitive reads from one IO port at the IO base address. See Section 4.10.

In memory mode, the receive status/length/frame locations are read using memory reads of a block of memory starting at memory base address + 0400h. Typically the memory locations are read sequentially using repetitive Move instructions (REP MOVS). See Section 4.9.

Random access is not needed. However, the first 118 bytes of the receive frame can be accessed randomly if word reads, on even word boundaries, are used. Beyond 118 bytes, the memory reads must be sequential. Byte reads, or reads on odd-word boundaries, can be performed only in sequential read mode. See Section 4.8.

The RxStatus word reports the status of the current received frame. RxEvent register 4 (PacketPage base + 0124h) has the same contents as the RxStatus register, except RxEvent is cleared when RxEvent is read.

The RxLength (receive length) word is the length, in bytes, of the data to be transferred to the host across the ISA bus. The register describes the length from the start of Source Address to the end of CRC, assuming that CRC has been selected (via Register 3 RxCFG, bit

BufferCRC). If CRC has not been selected, then the length does not include the CRC, and the CRC is not present in the receive buffer.

After the RxLength has been read, the receive frame can be read. When some portion of the frame is read, the entire frame should be read before reading the RxEvent register either directly or through the ISQ register. Reading the RxEvent register signals to the CS8900 that the host is finished with the current frame, and wants to start processing the next frame. In this case, the current frame will no longer be accessible to the host. The current frame will also become inaccessible if a Skip command is issued, or if the entire frame has been read. See Section 5.2.

Transmit Locations

The host can write frames into the CS8900 buffer using Memory writes using REP MOVS to the TxFrame location. See Section 5.7.

4.8 Eight and Sixteen Bit Transfers

A data transfer to or from the CS8900 can be done in either I/O or Memory space, and can be either 16 bits wide (word transfers) or 8 bits wide (byte transfers). Because the CS8900's internal architecture is based on a 16-bit data bus, word transfers are the most efficient.

The CS8900 does not support connection to 8-bit buses.

To transfer transmit frames to the CS8900 and receive frames from the CS8900, the host may mix word and byte transfers, provided it follows three rules:

1. The primary method used to access CS8900 memory is word access.

2. Word accesses to the CS8900's internal memory are kept on even-byte boundaries.

3. When switching from byte accesses to word accesses, a byte access to an even byte address must be followed by a byte access to an odd-byte address before the host may execute a word access (this will re-align the word transfers to even-byte boundaries). On the other hand, a byte access to an odd-byte address may be followed by a word access.

Failure to observe these three rules may cause data corruption.

Transferring Odd-Byte-Aligned Data

Some applications gather transmit data from more than one section of host memory. The boundary between the various memory locations may be either even- or odd-byte aligned. When such a boundary is odd-byte aligned, the host should transfer the last byte of the first block to an even address, followed by the first byte of the second block to the following odd address. It can then resume word transfers. An example of this is shown in Figure 4.3.
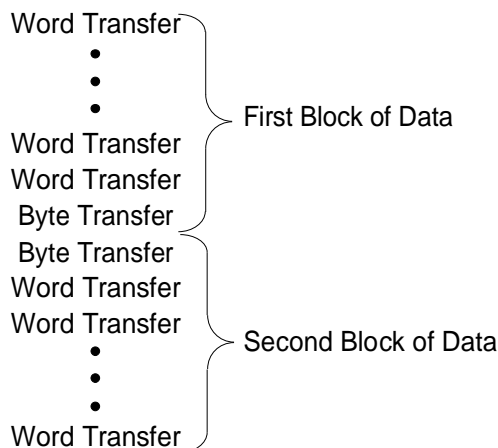
Figure 4.3. Odd-Byte Aligned Data

Random Access to CS8900 Memory

The first 118 bytes of a receive frame held in the CS8900's on-chip memory may be randomly accessed in Memory mode. After the first 118 bytes, only sequential access of received data is allowed. Either byte or word access is permitted, as long as all word accesses are executed to even-byte boundaries.

4.9   Memory Mode Operation

To configure the CS8900 for Memory Mode, the PacketPage memory must be mapped into a contiguous 4-kbyte block of host memory. The block must start at an X000h boundary, with the PacketPage base address mapped to X000h. When the CS8900 comes out of reset, its default configuration is I/O Mode. Once Memory Mode is selected, all of the CS8900's registers can be accessed directly.

In Memory Mode, the CS8900 supports Standard or Ready Bus cycles without introducing additional wait states (i.e., IOCHRDY is not deasserted).

Memory moves can use MOVD (double-word transfers) as long as the CS8900's memory base address is on a double word boundary. Since 286 processors don't support the MOVD instruction, word and byte transfers must be used with a 286.

Description	Mnemonic	Read/Write	Location: PacketPage base +
Receive Status	RxStatus	Read-only	0400h-0401h
Receive Length	RxLength	Read-only	0402h-0403h
Receive Frame	RxFrame	Read-only	starts at 0404h
Transmit Frame	TxFrame	Write-only	starts at 0A00h

Table 4.4. Receive/Transmit Memory Locations

4.9.1 Accesses in Memory Mode

The CS8900 allows Read/Write access to the internal PacketPage memory, and Read access of the optional Boot PROM. (See Section 3.7 for a description of the optional Boot PROM.)

A memory access occurs when all of the following are true:

- The address on the ISA System Address bus (SA0 - SA19) is within the Memory space range of the CS8900 or Boot PROM.

- The $\overline{CHIPSEL}$ input pin is low.

- Either the $\overline{MEMR}$ pin or the $\overline{MEMW}$ pin is low.

4.9.2 Configuring the CS8900 for Memory Mode

There are two different methods of configuring the CS8900 for Memory Mode operation. One method allows the CS8900's internal memory to be mapped anywhere within the host system's 24-bit memory space. The other method limits memory mapping to the first 1 Mbyte of host memory space.

General Memory Mode Operation: Configuring the CS8900 so that its internal memory can be mapped anywhere within host Memory space requires the following:

- a simple circuit must be added to decode the Latchable Address bus (LA20 - LA23) and the BALE signal.

- the host must configure the external logic with the correct address range as follows:

  1. Check to see if the INITD bit (Register 16, SelfST, bit 7) is set, indicating that initialization is complete.

  2. Check to see if the ELpresent bit (Register 16, SelfST, bit B) is set. This bit indicates that external logic for the LA bus decode is present.

  3. Set the ELSEL bit of the EEPROM Command Register to activate the $\overline{ELCS}$ pin for use with the external decode circuit.

  4. Configure the external logic serially.

- the host must write the memory base address into the Memory Base Address register (PacketPage base + 002Ch);

- the host must set the MemoryE bit (Register 17, BusCTL, Bit A); and

- the host must set the UseSA bit (Register 17, BusCTL, Bit 9).

Limiting Memory Mode to the First 1 Mbyte of Host Memory Space: Configuring the CS8900 so that its internal memory can be mapped on;y within the first 1 Mbyte of host memory space requires the following:

- the $\overline{CHIPSEL}$ pin must be tied low;

- the ISA-bus $\overline{SMEMR}$ signal must be connected to the $\overline{MEMR}$ pin;

- the ISA-bus $\overline{SMEMW}$ signal must be connected to the $\overline{MEMW}$ pin;

- the host must write the memory base address into the Memory Base Address register (PacketPage base + 002Ch);

- the host must set the MemoryE bit (Register 17, BusCTL, Bit A); and

- the host must clear the UseSA bit (Register 17, BusCTL, Bit 9).

4.9.3 Basic Memory Mode Transmit

Memory Mode transmit operations occur in the following order (using interrupts):

1. The host bids for storage of the frame by writing the Transmit Command to the TxCMD register (memory base + 0144h) and the transmit frame length to the TxLength register (memory base + 0146h). If the transmit length is erroneous, the command is discarded and the TxBidErr bit (Register 18, BusST, Bit 7) is set.

2. The host reads the BusST register (Register 18, memory base + 0138h). If the Rdy4TxNOW bit (Bit 8) is set, the frame can be written. If clear, the host must wait for CS8900 buffer memory to become available. If Rdy4TxiE (Register B, BufCFG, Bit 8) is set, the host will be interrupted when Rdy4Tx (Register C, BufEvent, Bit 8) becomes set.

3. Once the CS8900 is ready to accept the frame, the host executes repetitive memory-to-memory move instructions (REP MOVS) to memory base + 0A00h to transfer the entire frame from host memory to CS8900 memory.

For a more detailed description of transmit, see Section 5.7.

4.9.4 Basic Memory Mode Receive

Memory Mode receive operations occur in the following order (interrupts used to signal the presence of a valid receive frame):

1. A frame is received by the CS8900, triggering an enabled interrupt.

2. The host reads the Interrupt Status Queue (memory base + 0120h) and is informed of the receive frame.

3. The host reads RxStatus (memory base + 0400h) to learn the status of the receive frame.

4. The host reads RxLength (memory base + 0402h) to learn the frame's length.

5. The host reads the frame data by executing repetitive memory-to-memory move instructions (REP MOVS) from memory base + 0404h to transfer the entire frame from CS8900 memory to host memory.

For a more detailed description of receive, see Section 5.2.

4.9.5 Polling the CS8900 in Memory Mode

If interrupts are not used, the host can poll the CS8900 to check if receive frames are present and if memory space is available for transmit. However, this is beyond the scope of this data sheet.

4.10 I/O Space Operation

In I/O Mode, PacketPage memory is accessed through eight 16-bit I/O ports that are mapped into 16 contiguous I/O locations in the host system's I/O space. I/O Mode is the default configuration for the CS8900 and is always enabled. On power up, the default value of the I/O base address is set at 300h. (Note that 300h is typically assigned to LAN peripherals). The I/O base address may be changed to any available XXX0h location, either by loading configuration data from the EEPROM, or during system setup. Table 4.5 shows the CS8900 I/O Mode mapping:

Receive/Transmit Data Ports 0 and 1

These two ports are used when transferring transmit data to the CS8900 and receive data from the CS8900. Port 0 is used for 16-bit operations and

Offset	Type	Description
0000h	Read/Write	Receive/Transmit Data (Port 0)
0002h	Read/Write	Receive/Transmit Data (Port 1)
0004h	Write-only	TxCMD (Transmit Command)
0006h	Write-only	TxLength (Transmit Length)
0008h	Read-only	Interrupt Status Queue
000Ah	Read/Write	PacketPage Pointer
000Ch	Read/Write	PacketPage Data (Port 0)
000Eh	Read/Write	PacketPage Data (Port 1)

Table 4.5. I/O Mode Mapping

Ports 0 and 1 are used for 32-bit operations (lower-order word in Port 0).

TxCMD Port

The host writes the Transmit Command (TxCMD) to this port at the start of each transmit operation. The Transmit Command tells the CS8900 that the host has a frame to be transmitted, as well as how that frame should be transmitted. This port is mapped into PacketPage base + 0144h. See Register 9 in Section 4.4 for more information.

TxLength Port

The length of the frame to be transmitted is written here immediately after the Transmit Command is written. This port is mapped into PacketPage base + 0146h.

Interrupt Status Queue Port

This port contains the current value of the Interrupt Status Queue (ISQ). The ISQ is located at PacketPage base + 0120h. For a more detailed description of the ISQ, see Section 5.1.

PacketPage Pointer Port

The PacketPage Pointer Port is written whenever the host wishes to access any of the CS8900's internal registers. The first 12 bits (bits 0 through B) provide the internal address of the target reg-

ister to be accessed during the current operation. The next three bits (C, D, and E) must be 0. The last bit (Bit F) indicates whether or not the PacketPage Pointer should be auto-incremented to the next word location. Figure 4.4 shows the structure of the PacketPage Pointer.

PacketPage Data Ports 0 and 1

The PacketPage Data Ports are used to transfer data to and from any of the CS8900's internal registers. Port 0 is used for 16-bit operations and Port 0 and 1 are used for 32-bit operations (lower-order word in Port 0).

I/O Mode Operation

For an I/O Read or Write operation, the AEN pin must be low, and the 16-bit I/O address on the ISA System Address bus (SA0 - SA15) must match the address space of the CS8900. For a Read, the $\overline{IOR}$ pin must be low, and for a Write, the $\overline{IOW}$ pin must be low.

Note: The ISA Latchable Address Bus (LA17 - LA23) is not needed for applications that use only I/O Mode and Receive DMA operation. For these applications, the $\overline{ELCS}$ pin is tied low.
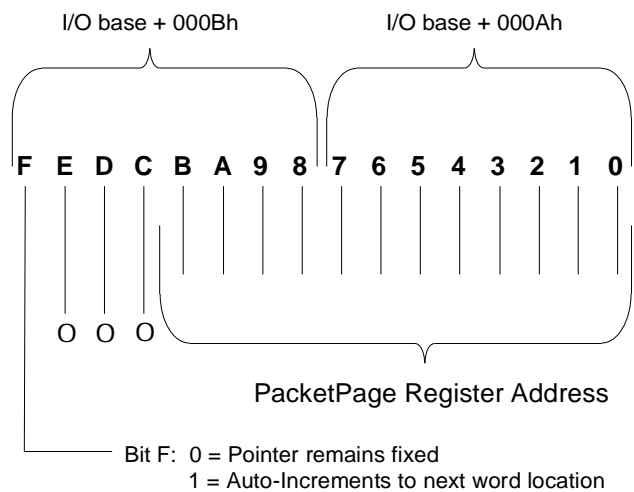
Figure 4.4. PackagePage Pointer

Basic I/O Mode Transmit

I/O Mode transmit operations occur in the following order (using interrupts):

1. The host bids for storage of the frame by writing the Transmit Command to the TxCMD Port (I/O base + 0004h) and the transmit frame length to the TxLength Port (I/O base + 0006h).

2. The host reads the BusST register (Register 18) to see if the Rdy4TxNOW bit (Bit 8) is set. To read the BusST register, the host must first set the PacketPage Pointer at the correct location by writing 0138h to the PacketPage Pointer Port (I/O base + 000Ah). It can then read the BusST register from the PacketPage Data Port (I/O base + 000Ch). If Rdy4TxNOW is set, the frame can be written. If clear, the host must wait for CS8900 buffer memory to become available. If Rdy4TxiE (Register B, BufCFG, Bit 8) is set, the host will be interrupted when Rdy4Tx (Register C, BufEvent, Bit 8) becomes set. If the TxBidErr bit (Register 18, BusST, Bit 7) is set, the transmit length is not valid.

3. Once the CS8900 is ready to accept the frame, the host executes repetitive write instructions (REP OUT) to the Receive/Transmit Data Port (I/O base + 0000h) to transfer the entire frame from host memory to CS8900 memory.

For a more detailed description of transmit, see Section 5.7.

Basic I/O Mode Receive

I/O Mode receive operations occur in the following order (In this example, interrupts are enabled to signal the presence of a valid receive frame):

1. A frame is received by the CS8900, triggering an enabled interrupt.

2. The host reads the Interrupt Status Queue Port (I/O base + 0008h) and is informed of the receive frame.

3. The host reads the frame data by executing repetitive read instructions (REP IN) from the Receive/Transmit Data Port (I/O base + 0000h) to transfer the frame from CS8900 memory to host memory. Preceding the frame data are the contents of the RxStatus register (PacketPage base + 0400h) and the RxLength register (PacketPage base + 0402h).

For a more detailed description of receive, see Section 5.2.

Accessing Internal Registers

To access any of the CS8900's internal registers in I/O Mode, the host must first setup the Packet-Page Pointer. It does this by writing the PacketPage address of the target register to the PacketPage Pointer Port (I/O base + 000Ah). The contents of the target register is then mapped into the PacketPage Data Port (I/O base + 000Ch).

If the host needs to access a sequential block of registers, the MSB of the PacketPage address of the first word to be accessed should be set to "1". The PacketPage Pointer will then move to the next word location automatically, eliminating the need to setup the PacketPage Pointer between successive accesses (see Figure 4.4).

Polling the CS8900 in I/O Mode

If interrupts are not used, the host can poll the CS8900 to check if receive frames are present and if memory space is available for transmit.

5.0 OPERATION

5.1 Managing Interrupts and Servicing the Interrupt Status Queue

The Interrupt Status Queue (ISQ) is used by the CS8900 to communicate Event reports to the host processor. Whenever an event occurs that triggers an enabled interrupt, the CS8900 sets the appropriate bit(s) in one of five registers, maps the contents of that register to the ISQ, and drives the selected interrupt request pin high (if an earlier interrupt is waiting in the queue, the interrupt request pin will already be high). When the host services the interrupt, it must first read the ISQ to learn the nature of the interrupt. It can then process the interrupt (the first read to the ISQ causes the interrupt request pin to go low.

Three of the registers mapped to the ISQ are event registers: RxEvent (Register 4), TxEvent (Register 8), and BufEvent (Register C). The other two registers are counter-overflow reports: RxMISS (Register 10) and TxCOL (Register 12). There may be more than one RxEvent report and/or more than one TxEvent report in the ISQ at a time. However, there may be only one BufEvent report, one RxMISS report and one TxCOL report in the ISQ at a time.

Event reports stored in the ISQ are read out in the order of priority, with RxEvent first, followed by TxEvent, BufEvent, RxMiss, and then TxCOL. The host only needs to read from one location to get the interrupt currently at the front of the queue. In Memory Mode, the ISQ is located at PacketPage base + 0120h. In I/O Mode, it is located at I/O base + 0008h. Each time the host reads the ISQ, the bits in the corresponding register are cleared and the next report in the queue moves to the front.

When the host starts reading the ISQ, it must read and process all Event reports in the queue.

A read-out of a null word (0000h) indicates that all interrupts have been read.

The ISQ is read as a 16-bit word. The lower six bits (0 through 5) contain the register number (4, 8, C, 10, or 12). The upper ten bits (6 through F) contain the register contents. The host must always read the entire 16-bit word, because the CS8900 does not support 8-bit access to its internal registers. Figure 5.1 shows the operation of the ISQ.

The active interrupt pin (INTRQx) is selected via the Interrupt Number register (PacketPage base + 22h). As an additional option, all of the interrupt pins can be tri-stated using the same register. see Section 4.3.

An event triggers an interrupt only when the EnableIRQ bit of the Bus Control register (bit F of register 17) is set. After the CS8900 has generated an interrupt, the first read of the ISQ makes the INTRQ output pin go low (inactive). INTRQ remains low until the null word (0000h) is read from the ISQ, or for 1.6 μs, whichever is longer.

Typically, when interrupts have been enabled for various error conditions (runt, CRC error, frame > 1518 bytes) via register 3, the software will also select that those frames be discarded (register 5).
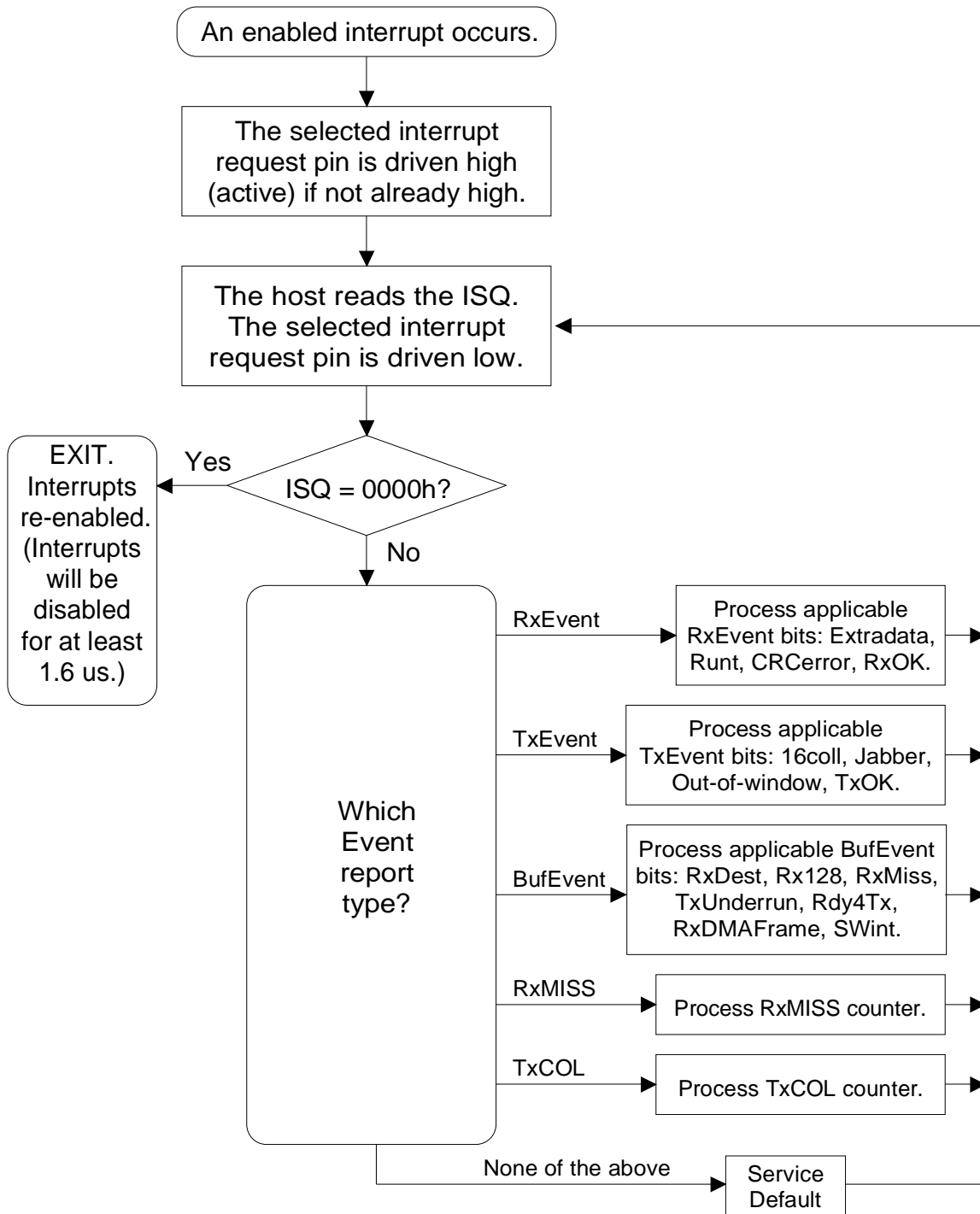
Figure 5.1. Interrupt Status Queue

5.2 Basic Receive Operation

Overview

Once an incoming packet has passed through the analog front end and Manchester decoder, it goes through the following three-step receive process:

1. Pre-Processing

2. Temporary Buffering

3. Transfer to Host

Figure 5.2 shows the steps in frame reception.

As shown in the figure, all receive frames go through the same pre-processing and temporary buffing phases, regardless of transfer method.
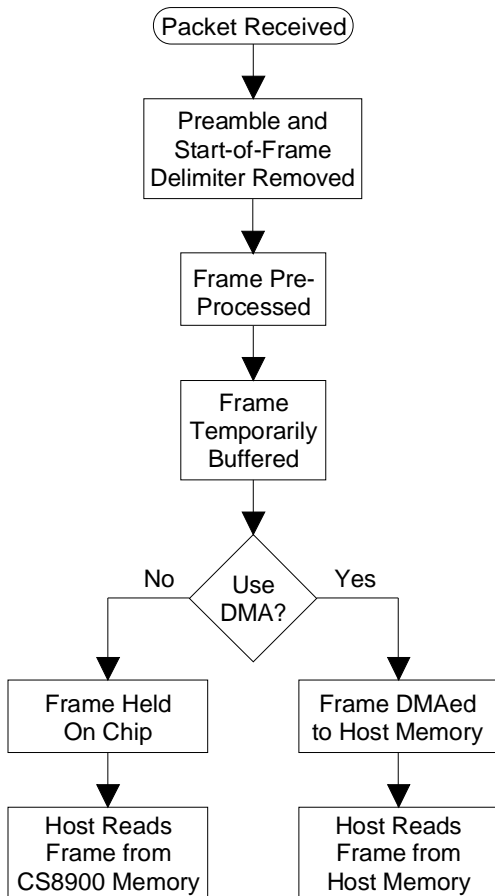
Figure 5.2. Frame Reception

Once a frame has been pre-processed and buffered, it can be accessed by the host in either Memory or I/O space. In addition, the CS8900 can transfer receive frames to host memory via host DMA. This section describes receive frame pre-processing and Memory and I/O space receive operation. Sections 5.4 through 5.5 describe DMA operation.

5.2.1 Terminology: Packet, Frame, and Transfer

The terms Packet, Frame, and Transfer are used extensively in the following sections. They are defined below for clarity:

Packet: The term "packet" refers to the entire serial string of bits transmitted over an Ethernet network. This includes the preamble, Start-of-Frame Delimiter (SFD), Destination Address (DA), Source Address (SA), Length field, Data field, pad bits (if necessary), and Frame Check Sequence (FCS, also called CRC). Figure 3.6 shows the format of a packet.

Frame: The term "frame" refers to the portion of a packet from the DA to the FCS. This includes the Destination Address (DA), Source Address (SA), Length field, Data field, pad bits (if necessary), and Frame Check Sequence (FCS, also called CRC). Figure 3.6 shows the format of a frame. The term "frame data" refers to all the data from the DA to the FCS that is to be transmitted, or that has been received.

Transfer: The term "transfer" refers to moving data across the ISA bus, to and from the CS8900. During receive operations, only frame data are transferred from the CS8900 to the host (the preamble and SFD are stripped off by the CS8900's MAC engine). The FCS may or may not be transferred, depending on the configuration. All transfers to and from the CS8900 are counted in bytes, but may be padded for double word alignment.

5.2.2 Receive Configuration

After each reset, the CS8900 must be configured for receive operation. This can be done automatically using an attached EEPROM or by writing configuration commands to the CS8900's internal registers (see Section 3.1). The items that must be configured include:

- which physical interface to use;

- which types of frames to accept;

- which receive events cause interrupts; and,

- how received frames are transferred.

Configuring the Physical Interface: Configuring the physical interface consists of determining which Ethernet interface should be active, and enabling the receive logic for serial reception. This is done via the LineCTL register (Register 13) and is described in Table 5.1.

Register 13, LineCTL		
Bit	Bit Name	Operation
6	SerRxON	When set, reception enabled.
8	AUIonly	When set, AUI selected (takes precedence over AutoAUI/10BT). When clear, 10BASE-T selected.
9	AutoAUI/ 10BT	When set, automatic interface selection enabled.
E	LoRx Squelch	When set, receiver squelch level reduced by approximately 6 dB.

Table 5.1. Physical Interface Configuration

Choosing which Frame Types to Accept: The RxCTL register (Register 5) is used to determine which frame types will be accepted by the CS8900 (a receive frame is said to be "accepted" when the frame is buffered, either on chip or in host memory via DMA). Table 5.2 describes the configuration bits in this register. Refer to Section 5.3 for a detailed description of Destination Address filtering.

Register 5, RxCTL		
Bit	Bit Name	Operation
6	IAHashA	When set, Individual Address frames that pass the hash filter are accepted*.
7	Promis cuousA	When set, all frames are accepted*.
8	RxOKA	When set, frames with valid length and CRC and that pass the DA filter are accepted.
9	MulticastA	When set, Multicast frames that pass the hash filter accepted*.
A	IndividualA	When set, frames with DA that matches the IA at PacketPage base + 0158h are accepted*.
B	BroadcastA	When set, all broadcast frames are accepted*.
C	CRCerrorA	When set, frames with bad CRC that pass the DA filter are accepted.
D	RuntA	when set, frames shorter than 64 bytes that pass the DA filter are accepted.
E	ExtradataA	When set, frames longer than 1518 bytes that pass the DA filter are accepted (only the first 1518 bytes are buffered).

  * Must also meet the criteria programmed into
    bits 8, C, D, and E.

Table 5.2. Frame Acceptance Criteria

Selecting which Events Cause Interrupts: The RxCFG register (Register 3) and the BufCFG register (Register B) are used to determine which receive events will cause interrupts to the host processor. Table 5.3 describes the interrupt enable (iE) bits in these registers.

Register 3, RxCFG		
Bit	Bit Name	Operation
8	RxOKiE	When set, there is an interrupt if a frame is received with valid length and CRC*.
C	CRCerroriE	When set, there is an interrupt if a frame is received with bad CRC*.
D	RuntiE	When set, there is an interrupt if a frame is received that is shorter than 64 bytes*.
E	ExtradataiE	When set, there is an interrupt if a frame is received that is longer than 1518 bytes*.

  * Must also pass the DA filter before there is an
    interrupt.

Register B, BufCFG		
Bit	Bit Name	Operation
7	RxDMAiE	When set, there is an interrupt if one or more frames are transferred via DMA.
A	RxMissiE	When set, there is an interrupt if a frame is missed due to insufficient receive buffer space.
B	Rx128iE	When set, there is an interrupt after the first 128 bytes of receive data have been buffered.
D	MissOvfloiE	When set, there is an interrupt if the RxMISS counter overflows.
F	RxDestiE	When set, there is an interrupt after the DA of an incoming frame has been buffered.

Table 5.3. Registers 3 & B Interrupt Configuration

Choosing How to Transfer Frames: The RxCFG register (Register 3) and the BusCTL register (Register 17) are used to determine how frames will be transferred to host memory, as described in Table 5.4.

Register 3, RxCFG		
Bit	Bit Name	Operation
7	StreamE	When set, StreamTransfer enabled.
9	RxDMAonly	When set, DMA slave operation used for all receive frames.
A	AutoRx DMAE	When set, Auto-Switch DMA enabled.
B	BufferCRC	When set, the received CRC is buffered.
Register 17, BusCTL		
Bit	Bit Name	Operation
B	DMABurst	When set, DMA operations hold the bus for up to approximately 28 µs. When clear, DMA operations are continuous.
D	RxDMAsize	When set, DMA buffer size is 64 Kbytes. When clear, DMA buffer size is 16 Kbytes.

Table 5.4. Frame Transfer Method

5.2.3 Receive Frame Pre-Processing

The CS8900 pre-processes all receive frames using a four step process:

1. Destination Address filtering;

2. Early Interrupt Generation;

3. Acceptance filtering; and,

4. Normal Interrupt Generation.

Figure 5.3 provides a diagram of frame pre-processing.

*Destination Address Filtering:*All incoming frames are passed through the Destination Address filter (DA filter). If the frame's DA passes the DA filter, the frame is passed on for further pre-processing. If it fails the DA filter, the frame is discarded. See Section 5.3 for a more detailed description of DA filtering.

*Early Interrupt Generation:*The CS8900 support the following two early interrupts that can be used to inform the host that a frame is being received:

• RxDest: The RxDest bit (Register C, BufEvent, Bit F) is set as soon as the Destination Address (DA) of the incoming frame passes the DA filter. If the RxDestiE bit (Register B, BufCFG, bit F) is set, the CS8900 generates a corresponding interrupt. Once RxDest is set, the host is allowed to read the incoming frame's DA (the first 6 bytes of the frame).

• Rx128: The Rx128 bit (Register C, BufEvent, Bit B) is set as soon as the first 128 bytes of the incoming frame have been received. If the Rx128iE bit (Register B, BufCFG, bit B) is set, the CS8900 generates a corresponding interrupt. Once the Rx128 bit is set, the RxDest bit is cleared and the host is allowed to read the first 128 bytes of the incoming frame. The Rx128 bit is cleared by the host reading the BufEvent register (either directly or through the Interrupt Status Queue) or by the CS8900 detecting the in-

CRYSTAL

coming frame's End-of-Frame (EOF) sequence.

Like all Event bits, RxDest and Rx128 are set by the CS8900 whenever the appropriate event occurs. Unlike other Event bits, RxDest and Rx128 may be cleared by the CS8900 without host intervention. All other event bits are cleared only by the host reading the appropriate event register, either directly or through the Interrupt Status Queue (ISQ). (RxDest and Rx128 can also be cleared by the host reading the BufEvent register, either directly or through the Interrupt Status Queue). Figure 5.4 provides a diagram of the Early Interrupt process.

Acceptance Filtering: The third step of pre-processing is to determine whether or not to accept the frame by comparing the frame with the criteria programmed into the RxCTL register (Register 5). If the receive frame passes the Acceptance filter, the frame is buffered, either on chip or in host memory via DMA. If the frame fails the Acceptance filter, it is discarded. The results of the Acceptance filter are reported in the RxEvent register (Register 4).

Normal Interrupt Generation: The final step of pre-processing is to generate any enabled interrupts that are triggered by the incoming frame. Interrupt generation occurs when the entire frame has been buffered (up to the first 1518 bytes). For more information about interrupt generation, see Section 5.1.

5.2.4   Held vs. DMAed Receive Frames

All accepted frames are either held in on-chip RAM until processed by the host, or stored in host memory via DMA. A receive frame that is held in on-chip RAM is referred to as a held receive frame. A frame that is stored in host memory via DMA is a DMAed receive frame. This section describes buffering and transferring held receive frames. Sections 5.4 through 5.6 describe DMAed receive frames.
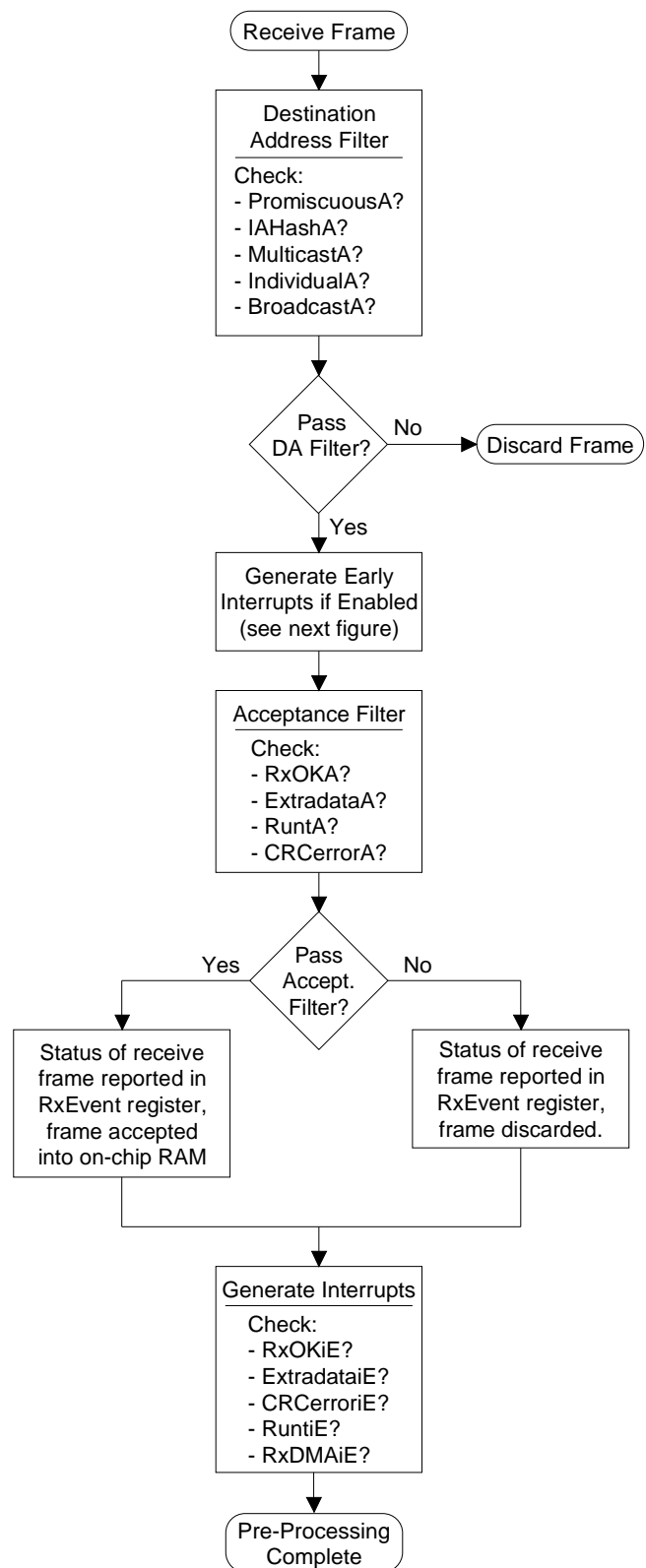
Figure 5.3. Receive Frame Pre-Processing

Receive Frame

DA Filter Passed?

Yes

No → Discard Frame

RxDest set.
Host may read the DA
(first 6 received bytes).

64 bytes Received?

Yes

No

EOF Received?

No

Yes → RxDest cleared
and Runt set.
If RuntA is set,
frame accepted and
Host may read frame.

128 bytes Received?

Yes

No

EOF Received?

No

Yes → RxDest cleared and
RxOK or CRCerror
set, as appropriate.
If RxOKA or CRCerrorA
is set, frame accepted and
Host may read frame.

Rx128 set and
RxDest cleared.
Host may read first
128 received bytes.

EOF Received?

No

Yes → Rx128 cleared and
RxOK, CRCerror or
Extradata set, as
appropriate. If ExtradataA,
RxOKA or CRCerrorA is
set, frame is accepted and
Host may read frame.

Figure 5.4. Early Interrupt Generation

5.2.5 Buffering Held Receive Frames

If space is available, an incoming frame will be temporarily stored in on-chip RAM, where it awaits processing by the host. Although this receive frame now occupies on-chip memory, the CS8900 does not commit the memory space to it until one of the following two conditions is true:

1. The entire frame has been received *and* the host has learned about the frame by reading the RxEvent register (Register 4), either directly or through the ISQ.

Or:

2. The frame has been partially received, causing either the RxDest bit (Register C, BufEvent, Bit F) or the Rx128 bit (Register C, BufEvent, Bit B) to become set, *and* the host has learned about the receive frame by reading the BufEvent register (Register C), either directly or through the ISQ.

When the CS8900 commits buffer space to a particular held receive frame (termed a committed received frame), no data from subsequent frames can be written to that buffer space until the frame is freed from commitment. (The committed received frame may or may not have been received error free.)

A received frame is freed from commitment by any one of the following conditions:

1. The host reads the entire frame sequentially in the order that it was received (first byte in, first byte out).

Or:

2. The host reads part or none of the frame, and then issues a Skip command by setting the Skip_1 bit (Register 3, RxCFG, bit 6).

Or:

3. The host reads part of the frame and then reads the RxEvent register (Register 5), either directly or through the ISQ, and learns of another receive frame. This condition is called an "implied Skip". On implied skips, a minimum of two words (Status and Length) of a frame must be read before rereading the ISQ.

Both early interrupts are disabled whenever there is a committed receive frame waiting to be processed by the host.

5.2.6 Transferring Held Receive Frames

The host can read-out held receive frames in Memory or I/O space. To transfer frames in Memory space, the host executes repetitive Move instructions (REP MOVS) from PacketPage base + 0404h. To transfer frames in I/O space, the host executes repetitive In instructions (REP IN) from I/O base + 0000h, with status and length preceding the frame.

There are three possible ways that the host can learn the status of a particular frame. It can:

1. Read the Interrupt Status Queue;

2. Read the RxEvent register directly (Register 4); or

3. Read the RxStatus register (PacketPage base + 0400h).

5.2.7 Receive Frame Visibility

Only one receive frame is visible to the host at a time. The receive frame's status can be read from the RxStatus register (PacketPage base + 0400h), and its length can be read from the RxLength register (PacketPage base + 0402h). For more information about Memory space operation, see Section 4.9. For more information about I/O space operation, see Section 4.10.

5.2.8 Example of Memory Mode Receive Operation

A common length for short frames is 64 bytes, including the 4-byte CRC. Suppose that such a frame has been received with the CS8900 configured as follows:

- The BufferCRC bit (Register 3, RxCFG, Bit B) is set causing the 4-byte CRC to be buffered with the rest of the receive data.

- The RxOKA bit (Register 5, RxCTL, Bit 8) is set, causing the CS8900 to accept good frames (a good frame is one with legal length and valid CRC).

- The RxOKiE bit (Register 3, RxCFG, Bit 8) is set, causing an interrupt to be generated whenever a good frame is received.

Then the transfer to the host would proceed as follows:

1. The CS8900 generates an RxOK interrupt to the host to signal the arrival of a good frame.

2. The host reads the ISQ (PacketPage base + 0120h) to assess the status of the receive frame and sees the contents of the RxEvent register (Register 4) with the RxOK bit (Bit 8) set.

3. The host reads the receive frame's length from the RxLength register (PacketPage base + 0402h).

4. The host reads the frame data by executing 32 consecutive MOV instructions from Packet-Page base + 0404h.

The memory map of the 64-byte frame is given in Table 5.5.

5.2.9 Receive Frame Byte Counter

The receive frame byte counter describes the number of bytes received for the current frame. The counter is incremented in real time as bytes are received from the Ethernet. The byte counter can be used by the driver to determine how many bytes are available for reading out of the CS8900. Maximum Ethernet throughput can be achieved by using I/O or memory modes, and by dedicating the CPU to reading this counter, and using the count to read the frame out of the CS8900 at the same time it is being received by the CS8900 from the Ethernet (parallel frame-reception and frame-read-out tasks).

The byte count register resides at PacketPage base + 50h.

Following an RxDest or Rx128 interrupt the register contains the number of bytes which are available to be read by the CPU. When the end of frame is reached, the count contains the final count value for the frame, including the allowance for the BufferCRC option. When this final count is read by the CPU the count register is set to zero. Therefore to read a complete frame using the byte count register, the register can be read and the data moved until a count of zero is

Memory Space Word Offset	Description of Data Stored in On-chip RAM
0400h	RxStatus Register (the host may skip reading 0400h since RxEvent was read from the ISQ.)
0402h	RxLength Register (In this example, the length is 40h bytes. The frame starts at 0404h, and runs through 0443h.)
0404h to 0409h	6-byte Source Address.
040Ah to 040Fh	6-byte Destination Address.
0410h to 011h	2-byte Length or Type Field.
0412h to 043Fh	46 bytes of data.
0440h	CRC, bytes 1 and 2
0442h	CRC, bytes 3 and 4

Table 5.5. Example Memory Map

detected. Then the RxEvent register can be read to determine the final frame status.

The sequence is as follows:

1. At the start of a frame, the byte counter matches the incoming character counter. The byte counter will have an *even* value prior to the end of the frame.

2. At the end of the frame, the final count, including the allowance for the CRC (if the BufferCRC option is enabled), is held until the byte counter is read.

3. When a read of the byte counter returns a count of zero, the previous count was the final count. The count may now have an *odd* value.

4. RxEvent should be read to obtain a final status of the frame, followed by a Skip command to complete the operation.

Note that all RxEvent's should be processed before using the byte counter. The byte counter should be used following a BufEvent when RxDest or Rx128 interrupts are enabled.

5.3   Receive Frame Address Filtering

The CS8900 is equipped with a Destination Address (DA) filter used to determine which receive frames will be accepted. (A receive frame is said to be "accepted" by the CS8900 when the frame data are placed in either on-chip memory, or in host memory by DMA). The DA filter can be configured to accept the following frame types:

Individual Address Frames: For all Individual Address frames, the first bit of the DA is a "0" (DA[0] = 0), indicating that the address is a Physical Address. The address filter accepts Individual Address frames whose DA matches the Individual Address (IA) stored at PacketPage base + 0158h, or whose hash-filtered DA matches one of the bits programmed into the

Logical Address Filter (the hash filter is described later in this section).

Multicast Frames: For Multicast Frames, the first bit of the DA is a "1" (DA[0] = 1), indicating that the frame is a Logical Address. The address filter accepts Multicast frames whose hash-filtered DA matches one of the bits programmed into the Logical Address Filter (the hash filter is described later is this section). As shown in Table 5.7, Broadcast Frames can be accepted as Multicast frames under a very specific set of conditions.

Broadcast Frames: Frames with DA equal to FFFF FFFF FFFFh. In addition, the CS8900 can be configured for Promiscuous Mode, in which case it will accept all receive frames, irrespective of DA.

Configuring the Destination Address Filter

The DA filter is configured by programming five DA filter bits in the RxCTL register (Register 5): IAHashA, PromiscuousA, MulticastA, IndividualA, and BroadcastA. Four of these bits are associated with four status bits in the RxEvent register (Register 4): IAHash, Hashed, IndividualAdr, and Broadcast. The RxEvent register reports the results of the DA filter for a given receive frame. The bits associated with DA filtering are summarized below:

Bit #	RxCTL Register 5	RxEvent Register 4
6	IAHashA	IAHash (used only if IAHashA = 1)
7	PromiscuousA	
9	MulticastA	Hashed
A	IndividualA	IndividualAdr (used only if IndividualA = 1)
B	BroadcastA	Broadcast (used only if BroadcastA = 1)

The IAHashA, MulticastA, IndividualA, and BroadcastA bits are used independently. As a result, many DA filter combinations are possible.

IAHashA	PromiscuousA	MulticastA	IndividualA	BroadcastA	Frames Accepted
0	0	0	1	0	Individual Address frames with DA matching the IA at PacketPage base + 0158h
1	0	0	0	0	Individual Address frames with DA that pass the hash filter (DA[0] must be "0")
0	0	1	0	0	Multicast frames with DA that pass the hash filter (DA[0] must be "1")
0	0	0	0	1	Broadcast frames
X	1	X	X	X	All frames

Table 5.6. DA Filtering Options

For example, if MulticastA and IndividualA are set, then all frames that are either Multicast or Individual Address frames are accepted. The PromiscuousA bit, when set, overrides the other four DA bits, and allows all valid frames to be accepted. Table 5.6 summarizes the configuration options available for DA filtering.

It may become necessary for the host to change the Destination Address (DA) filter criteria without resetting the CS8900. This can be done as follows:

1. Clear SerRxON (Register 13, LineCTL, Bit 6) to prevent any additional receive frames while the filter is being changed.

2. Modify the DA filter bits (B, A, 9, 7, and 6) in the RxCTL register.
   Modify the Logical Address Filter at Packet-Page base + 0150h, if necessary.
   Modify the Individual Address at PacketPage base + 0158h, if necessary.

3. Set SerRxON to re-enable the receiver.

Because the receiver has been disabled, the CS8900 will ignore frames while the host is changing the DA filter.

Hash Filter

The hash filter is used to help determine which Multicast frames and which Individual Address frames should be accepted by the CS8900.

Hash Filter Operation: See Figure 5.5. The DA of the incoming frame is passed through the CRC logic, generating a 32-bit CRC value. The six most-significant bits of the CRC are latched into the 6-bit hash register (HR). The contents of the HR are passed through a 6-to-64-bit decoder, asserting one of the decoder's outputs. The asserted output is compared with a corresponding bit in the 64-bit Logical Address Filter, located at PacketPage base + 0150h. If the decoder output and the Logical Address Filter bit match, the frame passes the hash filter and the Hashed bit (Register 4, RxEvent, Bit 9) is set. If the two do not match, the frame fails the filter and the Hashed bit is clear.

Whenever the hash filter is passed by a "good" frame, the RxOK bit (Register 4, RxEvent, Bit 8) is set and the bits in the HR are mapped to the Hash Table Index bits (Register 4, RxEvent, Bits A through F).

Broadcast Frame Hashing Exception

Table 5.7 describes in detail the content of the RxEvent register for each output of the hash and address filters, and describes an exception to nor-
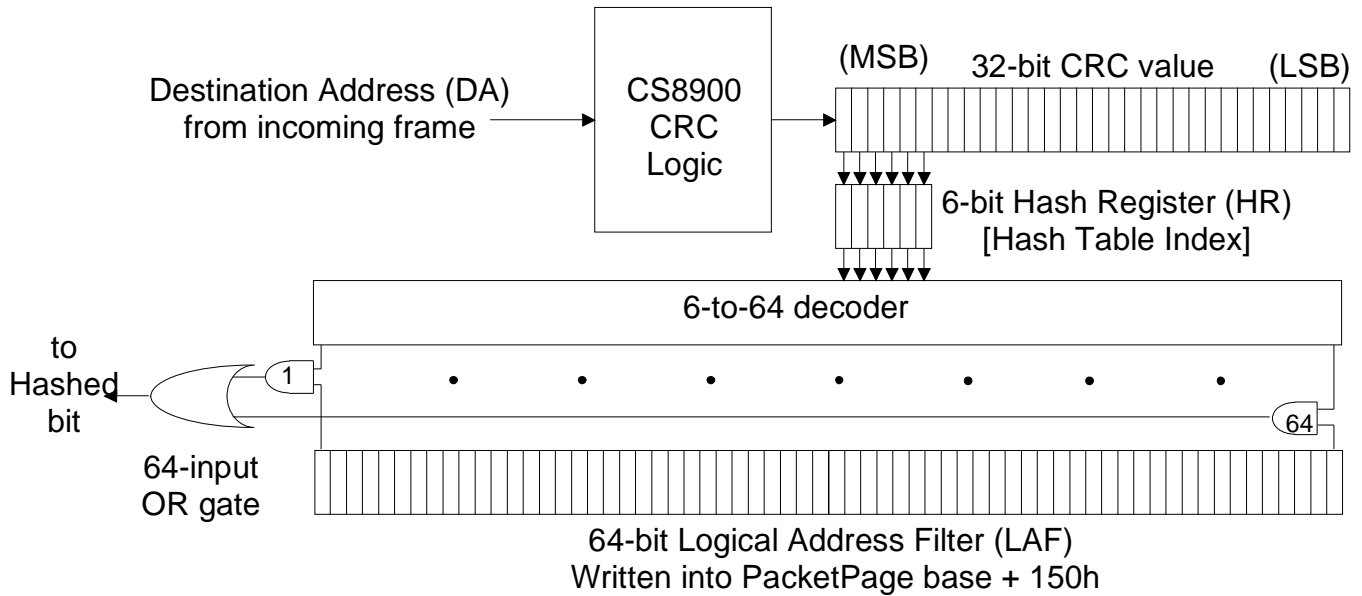
Figure 5.5. Hash Filter Operation

Address Type of Received Frame	Erred Frame?	Passes Hash Filter?	Contents of RxEvent				
			Bits F-A		Bit 9 Hashed	Bit 8 RxOK	Bit 6 IAHash
Individual Address	no	yes	Hash Table Index		1	1	1
	no	no	Extra Data Runt CRC Error Broadcast Individual Adr		0	1	0
	yes	don't care	Extra Data Runt CRC Error Broadcast Individual Adr		0	0	0
Multicast Address	no	yes	Hash table index		1	1	0
	no	no	Extra Data Runt CRC Error Broadcast Individual Adr		0	1	0
	yes	don't care	Extra Data Runt CRC Error Broadcast Individual Adr		0	0	0
Broadcast Address	no	yes (Note 1)	Extra Data Runt CRC Error Broadcast Individual Adr (actual value X00010)		1	1	0
	no	yes (Note 2)	Extra Data Runt CRC Error Broadcast Individual Adr		0	1	0
	no	no	Extra Data Runt CRC Error Broadcast Individual Adr		0	1	0
	yes	don't care	Extra Data Runt CRC Error Broadcast Individual Adr		0	0	0

Notes:
1. Broadcast frames are accepted as Multicast frames if and only if all the following conditions are met simultaneously:
   a) the Logical Address Filter is programmed as: (MSB) 0000 8000 0000 0000h (LSB). Note that this LAF value corresponds to a Multicast Addresses of both all 1s and 03-00-00-00-00-01.
   b) The Rx Control Register (register 5) is programmed to accept IndividualA, MulticastA, RxOK-only, and the following address filters were enabled: IAHashA and BroadcastA.
2. NOT (Note 1).

Table 5.7. Contents of RxEvent Upon Various Conditions

mal processing. That exception can occur when when the hash-filter Broadcast address matches a bit in the Logical Address Filter. To properly account for this exception, the software driver should use the following test to determine if the RxEvent register contains a normal RxEvent (meaning bits E-A are used for Extra data, Runt, CRC Error, Broadcast and IndividualAdr) or a hash-table RxEvent (meaning bits F-A contain the Hash Table Index).

If
  bit Hashed =0, or
  bit RxOK=0, or
  (bits F-A = 02h and the destination
    address is all ones)
then RxEvent contains a normal RxEvent
else RxEvent contained a hash RxEvent.

5.4   Receive DMA

5.4.1   Overview

The CS8900 supports a direct interface to the host DMA controller allowing it to transfer receive frames to host memory via slave DMA. The DMA option applies only to receive frames, and not transmit operation. The CS8900 offers three possible Receive DMA modes:

PacketPage Address	Register Description
0024h	**DMA Channel Number:** DMA channel number (0, 1, or 2) that defines the DMARQ/DMACK pin pair used.
0026h	**DMA Start of Frame:** 16-bit value that defines the offset from the DMA base address to the start of the most recently transferred received frame.
0028h	**DMA Frame Count:** The lower 12 bits define the number of valid frames transferred via DMA since the last read-out of this register. The upper 4 bits are reserved and not applicable.
002Ah	**DMA Byte Count:** Defines the number of bytes that have been transferred via DMA since the last read-out of this register.

Table 5.8. Receive DMA Registers

1. Receive-DMA-only mode: All receive frames are transferred via DMA.

2. Auto-Switch DMA: DMA is used only when needed to help prevent missed frames.

3. StreamTransfer: DMA is used to minimize the number of interrupts to the host.

This section provides a description of Receive-DMA-only mode. Section 5.5 describes Auto-Switch DMA and Section 5.6 describes StreamTransfer.

5.4.2   Configuring the CS8900 for DMA Operation

The CS8900 interfaces to the host DMA controller through one pair of the DMA request/acknowledge pins (see Section 3.2 for a description of the CS8900's DMA interface).

Four 16-bit registers are used for DMA operation. These are described in Table 5.8.

Receive-DMA-only mode is enabled by setting the RxDMAonly bit (Register 3, RxCFG, Bit 9).

Note: If the RxDMAonly bit and the AutoRxD-MAE bit (Register 3, RxCFG, Bit A) are both set, then RxDMAonly takes precedence, and the CS8900 is in DMA mode for all receive frames.

5.4.3   DMA Receive Buffer Size

In receive DMA mode, the CS8900 stores received frames (along with their status and length) in a circular buffer located in host memory space. The size of the circular buffer is determined by the RxDMAsize bit (Register 17, BusCTL, Bit D). When RxDMAsize is clear, the buffer size is 16 Kbytes. When RxDMAsize is set, the buffer is 64 Kbytes. It is the host's task to locate and keep track of the DMA receive buffer's base address. The DMA Start-of-Frame register is the only circuit affected by this bit.

APPLICATION NOTE: As a result of the PC architecture, DMA cannot occur across a 128K boundary in memory. Thus, the DMA buffer reserved for the CS8900 must not cross a 128K boundary in host memory if DMA operation is desired. Requesting a 64K, rather than a 16K buffer, increases the probability of crossing a 128K boundary. After the driver requests a DMA buffer, the driver must check for a boundary crossing. If the boundary is crossed, then the driver must disable DMA functionality.

5.4.4   Receive-DMA-Only Operation

If space is available, an incoming frame is temporarily stored in on-chip RAM. When the entire frame has been received, pre-processed, and accepted, the CS8900 signals the DMA controller that a frame is to be transferred to host memory by driving the selected DMA Request pin high. The DMA controller acknowledges the request by driving the DMA Acknowledge pin low. The CS8900 then transfers the contents of the RxStatus register (PacketPage base + 0400h) and the RxLength register (PacketPage base + 0402h) to host memory, followed by the frame data. If the DMABurst bit (Register 17, BusCTL, Bit B) is clear, the DMA Request pin remains high until the entire frame is transferred. If the DMABurst bit is set, the DMA Request pin (DMARQ) remains high for approximately 28 μs then goes low for approximately 1.3 μs to give the CPU and other peripherals access to the bus.

When the transfer is complete, the CS8900 does the following:

- updates the DMA Start-of-Frame register (PacketPage base + 0026h);

- updates the DMA Frame Count register (PacketPage base + 0028h);

- updates DMA Byte Count register (Packet-Page base + 002Ah);

- sets the RxDMAFrame bit (Register C, BufEvent, Bit 7); and,

- de-allocates the buffer space used by the transferred frame.

In addition, if the RxDMAiE bit (Register B, BufCFG, Bit 7) is set, a corresponding interrupt occurs.

When the host processes DMAed frames, it must read the DMA Frame Count register.

Whenever a receive frame is missed (lost) due to insufficient receive buffer space, the RxMISS counter (Register 10) is incremented. A missed receive frame causes the counter to increment in either DMA or non-DMA modes.

Note that when in DMA mode, reading the contents of the RxEvent register will return 0000h. Status information should be obtained from the DMA buffer.

5.4.5   Committing Buffer Space to a DMAed Frame

Although a receive frame may occupy space in the host memory's circular DMA buffer, the CS8900's Memory Manager does not commit the buffer space to the receive frame until the entire frame has been transferred *and* the host learns of the frame's existence by reading the Frame Count register (PacketPage base + 0028h).

When the CS8900 commits DMA buffer space to a particular DMAed receive frame (termed a committed received frame), no data from subsequent frames can be written to that buffer space until the committed received frame is freed from commitment. (The committed received frame may or may not have been received error free.)

A committed DMAed receive frame is freed from commitment by any one of the following conditions:

1. The host re-reads the DMA Frame Count register (PacketPage base + 0028h).

2. New frames have been transferred via DMA, and the host reads the BufEvent register (either directly or from the ISQ) and sees that the RxDMAFrame bit is set (this condition is termed an "implied Skip").

3. The host issues a Reset-DMA command by setting the ResetRxDMA bit (Register 17, BusCTL, Bit 6).

5.4.6 DMA Buffer Organization

When DMA is used to transfer receive frames, the DMA Start-of-Frame register (PacketPage Base + 0026h) defines the offset from the DMA base to the start of the most recently transferred received frame. Frames stored in the DMA buffer are transferred as words and maintain double-word (32-bit) alignment. Unfilled memory space between successive frames stored in the DMA buffer may result from double-word alignment. These "holes" may be 1, 2, or 3 bytes, depending on the length of the frame preceding the hole.

5.4.7 RxDMAFrame Bit

The RxDMAFrame bit (Register C, BufEvent, bit 7) is controlled by the CS8900 and is set whenever the value in the DMA Frame Count register is non-zero. The host cannot clear RxDMAFrame by reading the BufEvent register (Register C). Table 5.9 summarizes the criteria used to set and clear RxDMAFrame.

5.4.8 Receive DMA Example Without Wrap-Around

Figure 5.6 shows three frames stored in host memory by DMA without wrap-around.

5.4.9 Receive DMA Operation for RxDMA-Only Mode

In an RxDMAOnly mode, a system DMA moves all the received frames from the on-chip memory to an external 16- or 64-Kbyte buffer memory. The received frame must have passed the destination address filter, and must be completely received. Usually, the DMA receive frame interrupt (RxDMAiE, bit 7, Register B, BufCFG) is set so that the CS8900 generates an interrupt when a frame is transferred by DMA. Figure 5.7 shows how a DMA Receive Frame interrupt is processed.

	Non-StreamTransfer Mode	StreamTransfer Mode (see Section 5.6)
To Set RxDMAFrame	The RxDMAFrame bit is set whenever the DMA Frame Count register (PacketPage base + 0028h) transitions to non-zero.	The RxDMAFrame bit is set at the end of a Stream Transfer cycle.
To Clear RxDMAFrame	The DMA Frame Count is zero.	The DMA Frame Count is zero.

Table 5.9. RxDMAFrame Hit

In the interrupt service routine, the BufEvent register (register C), bit RxDMA Frame (bit 7) indicates that one or more receive frames were transferred using DMA. The software driver should maintain a pointer (e.g. P_{DMA_START}) that will point to the beginning of a new frame. After the CS8900 is initialized and before any frame is received, pointer P_{DMA_START} points to the beginning of the DMA buffer memory area. The first read of the DMA Frame Count, C_{DMA}, commits the memory covered by the C_{DMA} count, and the DMA cannot overwrite this committed space until the space is freed. The driver then processes the frames described by the C_{DMA} count and makes a second read of the DMA frame count. This second read frees the buffer memory space described by the C_{DMA} counter.
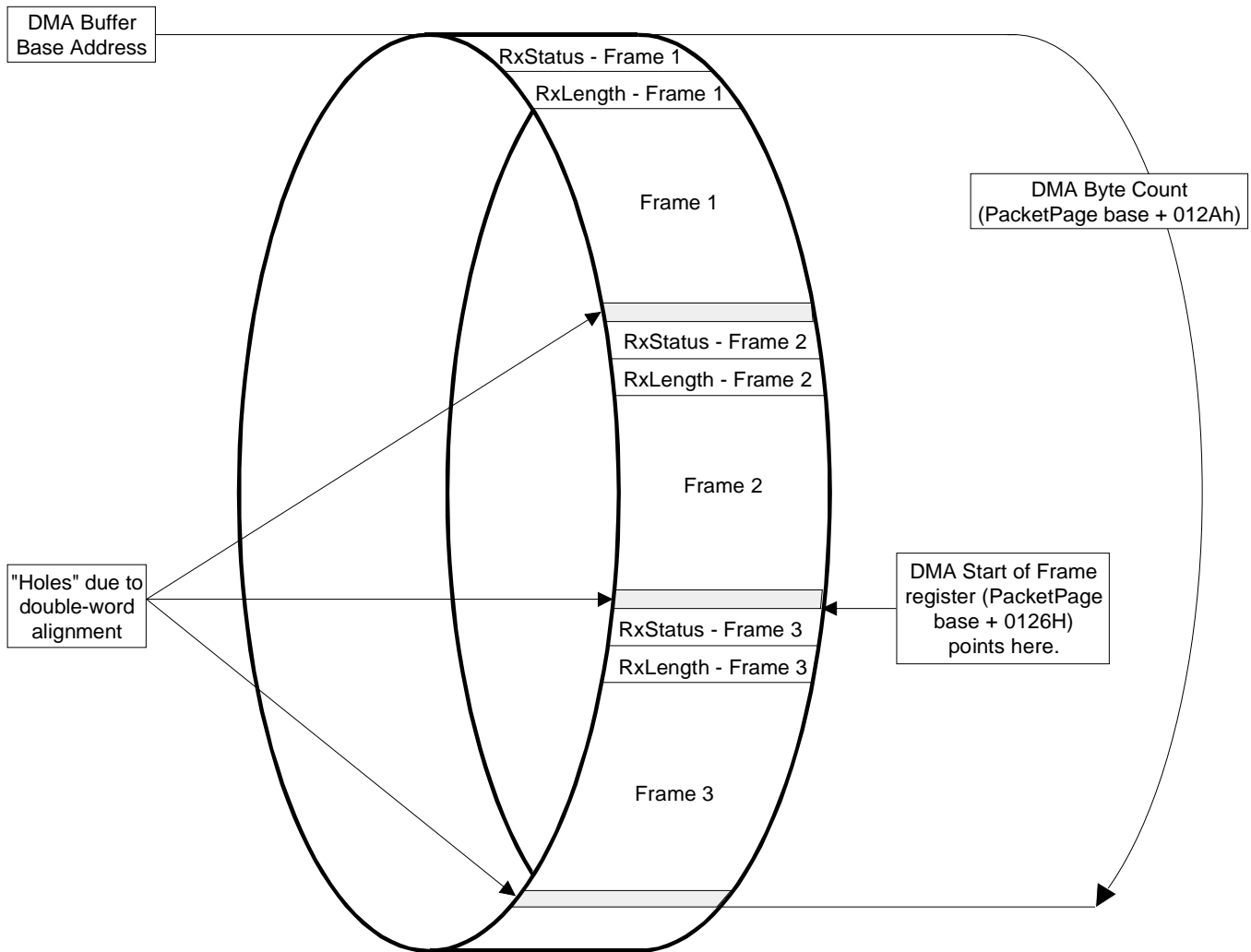
Figure 5.6. Example of Frames Stored in DMA Buffer

During the frame processing, the software should advance the P$_{DMA\_START}$ pointer. At the end of processing a frame, pointer P$_{DMA\_START}$ should be made to align with a double-word boundary. The software remains in the loop until the DMA frame count read is zero.

```
        Host Enters Interrupt Routine

                    RxDMA            No    Process other events
                    Frame          ──────▶ that caused interrupt
                    bit set?

           Yes

        Read the DMA frame Count (CDMA)
        (PacketPage base + 0028h)

                    CDMA             Yes   Process other events
                    = 0 ?          ──────▶ that caused interrupt

           No

             Process the
             CDMA Frames
```
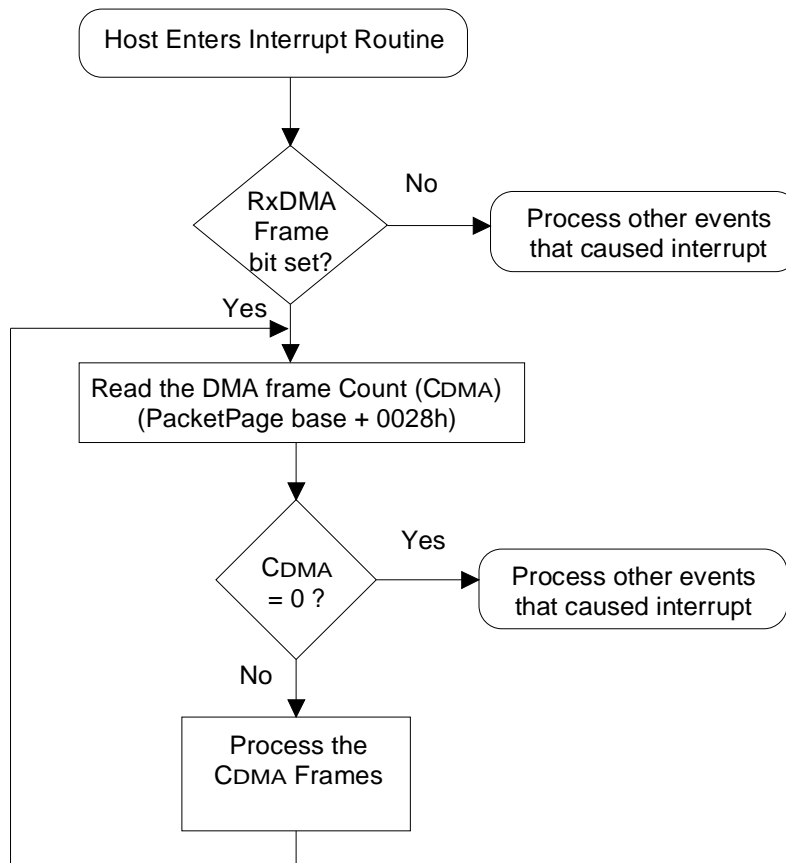
Figure 5.7. RxDMA Only Operation

5.5 Auto-Switch DMA

Overview

The CS8900 supports a unique feature, Auto-Switch DMA, that allows it to switch between Memory or I/O mode and Receive DMA automatically. Auto-Switch DMA allows the CS8900 to realize the performance advantages of Memory or I/O mode while minimizing the number of missed frames that could result due to slow processing by the host.

Configuring the CS8900 for Auto-Switch DMA

Auto-Switch DMA mode requires the same configuration as Receive-DMA-only mode, with one exception: the AutoRxDMAE bit (Register 3, RxCFG, Bit A) must be set, and the RxDMAonly bit (Register 3, RxCFG, Bit 9) must be clear (see Section 5.4, Configuring the CS8900 for DMA Operation). In Auto-Switch DMA mode, the CS8900 operates in non-DMA mode if possible, only switching to slave DMA if necessary.

Note that if the AutoRxDMAE bit and the RxDMAonly bit (Register 3, RxCFG, bit 9) are both set, the CS8900 uses DMA for all receive frames.

Auto-Switch DMA Operation

Whenever a frame begins to be received in Auto-Switch DMA mode, the CS8900 checks to see if there is enough on-chip buffer space to store a maximum length frame. If there is, the incoming frame is pre-processed and buffered as normal. If there isn't, the CS8900's MAC engine compares the frame's Destination Address (DA) to the criteria programmed into the DA filter. If the incoming DA fails the DA filter, the frame is discarded. If the DA passes the DA filter, the CS8900 automatically switches to DMA mode and starts transferring the frame(s) currently being held in the on-chip buffer into host memory.

This frees up buffer space for the incoming frame.

Figure 5.8 shows the steps the CS8900 goes through in determining when to automatically switch to DMA.

Whenever the CS8900 automatically enters DMA, at least one complete frame is already stored in the on-chip buffer. Because frames are transferred to the host in the same order as received (first in, first out), the beginning of the received frame that triggered the switch to DMA is not the first frame to be transferred. Instead, the oldest non-committed frame in the on-chip buffer is the first frame to use DMA. When DMA begins, any pending RxEvent reports in
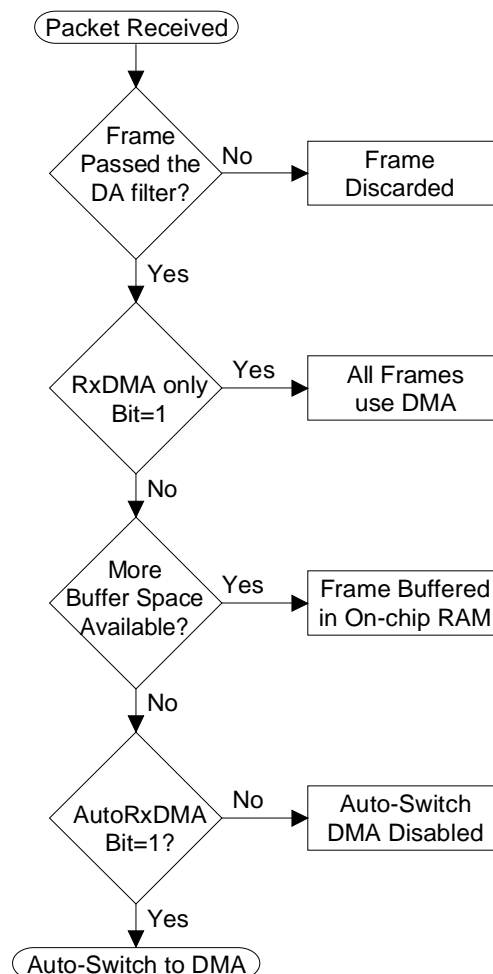
Figure 5.8. Conditions for Switching to DMA

the Interrupt Status Queue are discarded because the host cannot process those events until the corresponding frames have been completely DMAed.

Auto-Switch DMA works only on entire received frames. The CS8900 does not use Auto-Switch DMA to transfer partial frames. Also, when a frame has been committed (see Section 5.2.5), the CS8900 will not switch to DMA mode until the committed frame has been transferred completely or skipped.

After a complete frame has been moved to host memory, the CS8900 updates the DMA Start-of-Frame register (PacketPage base + 0126h), the DMA Frame Count register (PacketPage base + 0128h), and the DMA Byte Count register, then sets the RxDMAFrame bit (Register C, BufEvent, bit 7). If RxDMAiE (Register B, BufCFG, bit 7) is set, a corresponding interrupt occurs.

DMA Channel Speed vs. Missed Frames

When the CS8900 starts DMA, the entire oldest, noncommitted frame must be placed in host memory before on-chip buffer space will be freed for the next incoming frame. If the oldest frame is relatively large, and the next incoming frame also large, the incoming frame may be missed, depending on the speed of the DMA channel. If this happens, the CS8900 will increment the RxMiss counter (Register 10) and clear any event reports (RxEvent and BufEvent) associated with the missed frame.

Exit From DMA

When the CS8900 has activated receive DMA, it remains in DMA mode until all of the following are true:

- The host processes all RxEvent and BufEvent reports pending in the ISQ.

- The host reads a zero value from the DMA Frame Count register (PacketPage base + 0028h).

- The CS8900 is not in the process of transferring a frame via DMA.

Auto-Switch DMA Example

Figure 5.9 shows how the CS8900 enters and exits Auto-Switch DMA mode.

Enter Example Here

Entering this example, the receive buffer is empty and the DMA Frame Count (PacketPage base + 0028h) is zero.

Frame 1

Frame 1 received and completely stored in on-chip RAM.

Frame 2

Frame 2 received and completely stored in on-chip RAM.

Time

At this point, the CS8900 does not have sufficient buffer space for another complete large frame (1518 bytes).

Frame 3

Frame 3 starts to be received and passes the DA filter. This activates Auto-Switch DMA.

Frame 1 is placed in host memory via DMA freeing space for the incoming Frame 3. The CS8900 updates the DMA Frame Count, DMA Start of Frame and DMA Byte Count registers. It then sets the RxDMA DMAFrame bit and generates an interrupt.

Receive DMA used during this time.

Frame 2 is placed in host memory via DMA and the CS8900 updates the DMA registers.

The host responds to the RxDMAFrame interrupt, and reads the Frame Count register, which is cleared when read. Since there are no receive interrupts pending, the CS8900 exits DMA (assumes Frame 3 is still coming in).

Frame 3 is completely buffered in on-chip RAM, and awaits processing by the host.

Exit Example

Figure 5.9. Example of Auto-Switch DMA

5.6 StreamTransfer

Overview

The CS8900 supports an optional feature, StreamTransfer, that can reduce the amount of CPU overhead associated with frame reception. StreamTransfer works during periods of high receive activity by grouping multiple receive events into a single interrupt, thereby reducing the number of receive interrupts to the host processor. During periods of peak loading, StreamTransfer will eliminate 7 out of every 8 interrupts, cutting interrupt overhead by up to 87%.

Configuring the CS8900 for StreamTransfer

StreamTransfer is enabled by setting the StreamE bit along with either the AutoRxDMAE bit or the RxDMAonly bit in register Receiver Configuration (register 3). (StreamTransfer must not be selected unless either one of AutoRxDMAE or RxDMA-only is selected.) StreamTransfer only applies to "good" frames (frames of legal length with valid CRC). Therefore, the RxOKA bit and the RxOKiE bit must both be set. Finally, StreamTransfer works on whole packets and is not compatible with early interrupts. This requires that the RxDestiE bit and the Rx128iE bit both be clear.

Register Name	Bit	Bit Name	Value
Register 3, RxCFG	7	StreamE	1
	8	RxOKiE	1
	9 or A	RxDMAonly or AutoRxDMA	1 or 1
Register 5, RxCTL	8	RxOKA	1
Register B, BufCFG	7	RxDMAiE	1
	F	RxDestiE	0
	B	Rx128iE	0

Table 5.10. Stream Transfer Configuration

Table 5.10 summarizes how to configure the CS8900 for StreamTransfer.

StreamTransfer Operation

When StreamTransfer is enabled, the CS8900 will initiate a StreamTransfer cycle whenever two or more frames with the following characteristics are received:

1. pass the Destination Address filter;

2. are of legal length with valid CRC; and,

3. are spaced "back-to-back" (between 9.6 and 52 µs apart).

During a StreamTransfer cycle the CS8900 does the following:

- delays the normal RxOK interrupt associated with the first receive frame;

- switches to receive DMA mode;

- transfers up to eight receive frames into host memory via DMA;

- updates the DMA Start-of-Frame register (PacketPage base + 0026h);

- updates the DMA Frame Count register (PacketPage base + 0028h);

- updates DMA Byte Count register (PacketPage base + 002Ah);

- sets the RxDMAFrame bit (Register C, BufEvent, Bit 7); and,

- generates an RxDMAFrame interrupt.

Keeping StreamTransfer Mode Active

When the CS8900 initiates a StreamTransfer cycle, it will continue to execute cycles as long as the following conditions hold true:

- all packets received are of legal length with valid CRC;

- each packet follows its predecessor by less than 52 µs; and,

- the DA of each packet passes the DA filter.

If any of these conditions are not met, the CS8900 exits StreamTransfer by generating RxOK and RxDMA interrupts. The CS8900 then returns to either Memory, I/O, or DMA mode, depending on configuration.

Example of StreamTransfer

Figure 5.10 shows how four back-to-back frames, followed by five back-to-back frames, would be received without StreamTransfer. Figure 5.11 shows how the same sequence of frames would be received with StreamTransfer.

Receive DMA Summary

Table 5.11 summarize the Receive DMA configuration options supported by the CS8900.
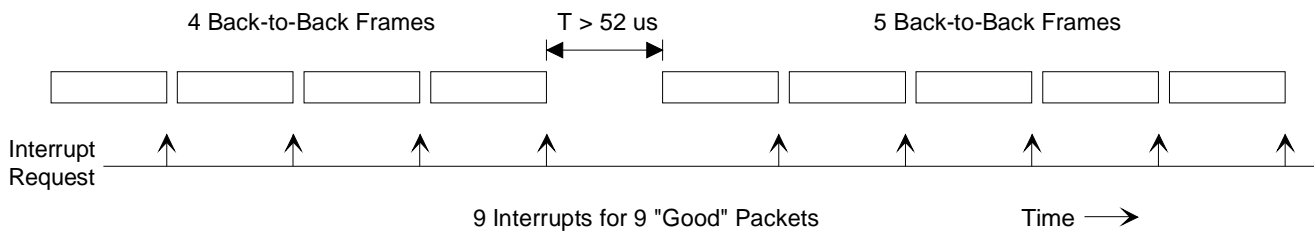
Figure 5.10. Receive Example Without StreamTransfer
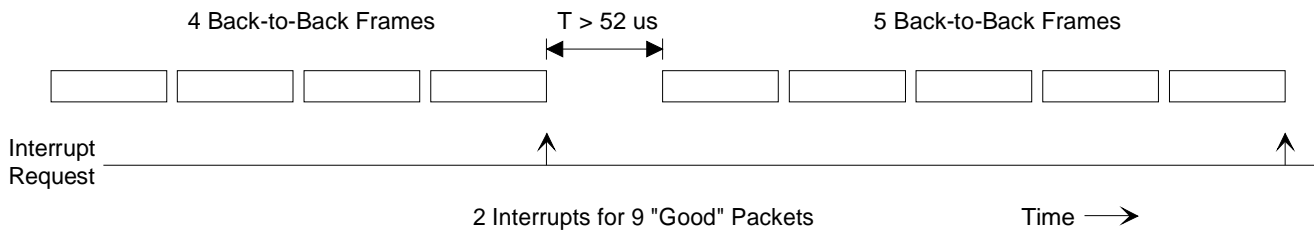
Figure 5.11. Receive Example With StreamTransfer

RxDMAonly (Register 3, RxCFG, Bit 9)	AutoRxDMAiE (Register 3, RxCFG, Bit A)	RxDMAiE (Register B, BufCFG, Bit 7)	RxOKiE (Register 3, RxCFG, Bit 8)	CS8900 Configuration
1	NA	0	NA	Receive DMA used for all receive frames, without interrupts.
1	NA	1	NA	Receive DMA used for all receive frames, with BufEvent interrupts.
0	1	0	0	Auto-Switch DMA used if necessary, without interrupts.
0	1	1	1	Auto-Switch DMA used if necessary, with RxEvent and BufEvent interrupts possible.
0	0	NA	NA	Memory or I/O Mode only.

Table 5.11. Receive DMA Configuration Options

5.7 Transmit Operation

5.7.1 Overview

Packet transmission occurs in two phases. In the first phase, the host moves the Ethernet frame into the CS8900's buffer memory. The first phase begins with the host issuing a Transmit Command. This informs the CS8900 that a frame is to be transmitted and tells the chip when (i.e. after 5, 381, or 1021 bytes have been transferred or after the full frame has been transferred to the CS8900) and how the frame should be sent (i.e. with or without CRC, with or without pad bits, etc.). The host follows the Transmit Command with the Transmit Length, indicating how much buffer space is required. When buffer space is available, the host writes the Ethernet frame into the CS8900's internal memory, using either Memory or I/O space.

In the second phase of transmission, the CS8900 converts the frame into an Ethernet packet then transmits it onto the network. The second phase begins with the CS8900 transmitting the preamble and Start-of-Frame delimiter as soon as the proper number of bytes has been transferred into its transmit buffer (5, 381, 1021 bytes or full frame, depending on configuration). The preamble and Start-of-Frame delimiter are followed by the data transferred into the on-chip buffer by the host (Destination Address, Source Address, Length field and LLC data). If the frame is less than 64 bytes, including CRC, the CS8900 adds pad bits if configured to do so. Finally, the CS8900 appends the proper 32-bit CRC value.

5.7.2 Transmit Configuration

After each reset, the CS8900 must be configured for transmit operation. This can be done automatically using an attached EEPROM, or by writing configuration commands to the CS8900's internal registers (see Section 3.1). The items that must be configured include which physical interface to use and which transmit events cause interrupts.

Configuring the Physical Interface: Configuring the physical interface consists of determining which Ethernet interface should be active (10BASE-T or AUI), and enabling the transmit logic for serial transmission. Configuring the Physical Interface is accomplished via the LineCTL register (Register 13) and is described in Table 5.12.

Note that the CS8900 transmits in 10BASE-T mode when no link pulses are being received only if bit DisableLT is set in register Test Control (Register 19).

Register 13, LineCTL		
Bit	Bit Name	Operation
7	SerTxON	When set, transmission enabled.
8	AUIonly	When set, AUI selected (takes precedence over AutoAUI/10BT). When clear, 10BASE-T selected.
9	AutoAUI/ 10BT	When set, automatic interface selection enabled.
B	Mod BackoffE	When set, the modified backoff algorithm is used. When clear, the standard backoff algorithm is used.
D	2-part DefDis	When set, two-part deferral is disabled.

Table 5.12. Physical Interface Configuration

Selecting which Events Cause Interrupts: The TxCFG register (Register 7) and the BufCFG register (Register B) are used to determine which transmit events will cause interrupts to the host processor. Tables 5.13 and 5.14 describe the interrupt enable (iE) bits in these registers.

Register 7, TxCFG		
Bit	Bit Name	Operation
6	Loss-of-CRSiE	When set, there is an interrupt whenever the CS8900 fails to detect Carrier Sense after transmitting the preamble (applies to the AUI only).
7	SQEerroriE	When set, there is an interrupt whenever there is an SQE error.
8	TxOKiE	When set, there is an interrupt whenever a frame is transmitted successfully.
9	Out-of-windowiE	When set, there is an interrupt whenever a late collision is detected.
A	JabberiE	When set, there is an interrupt whenever there is a jabber condition.
B	AnycolliE	When set, there is an interrupt whenever there is a collision.
F	16colliE	When set, there is an interrupt whenever the CS8900 attempts to transmit a single frame 16 times.

Table 5.13. Transmitting Interrupt Configuration

Register B, BufCFG		
Bit	Bit Name	Operation
8	Rdy4TxiE	When set, there is an interrupt whenever buffer space becomes available for a transmit frame (used with a Transmit Request).
9	TxUnderRuniE	When set, there is an interrupt when the CS8900 runs out of data after transmit has started.
A	TxColOvfloiE	When set, there is an interrupt whenever the TxCol counter overflows.

Table 5.14. Tranmit Interrupt Configuration

5.7.3 Changing the Configuration

When the host configures these registers it does not need to change them for subsequent packet transmissions. If the host does choose to change the TxCFG or BufCFG registers, it may do so at any time. The effects of the change are noticed immediately. That is, any changes in the Interrupt Enable (iE) bits may affect the packet currently being transmitted.

If the host chooses to change bits in the LineCTL register after initialization, the ModBackoffE bit and any receive related bit (LoRxSquelch, SerRxON) may be changed at any time. However, the Auto AUI/10BT and AUIonly bits should not be changed while the SerTxON bit is set. If any of these three bits are to be changed, the host should first clear the SerTxON bit (Register 13, LineCTL, Bit 7), and then set it when the changes are complete.

5.7.4 Enabling CRC Generation and Padding

Whenever the host issues a Transmit Request command, it must indicate whether or not the Cyclic Redundancy Check (CRC) value should be appended to the transmit frame, and whether or not pad bits should be added (if needed). Table 5.15 describes how to configure the CS8900 for CRC generating and padding.

Register 9, TxCMD		
Inhibit CRC (Bit C)	TxPad Dis (Bit D)	Operation
0	0	Pad to 64 bytes if necessary (including CRC).
1	0	Send a runt frame if specified length less than 60 bytes.
0	1	Pad to 60 bytes if necessary (without CRC).
1	1	Send runt if specified length less than 64. The CS8900 will not transmit a frame that is less than 3 bytes.

Table 5.15. CRC and Padding Configuration

5.7.5 Individual Packet Transmission

Whenever the host has a packet to transmit, it must issue a Transmit Request to the CS8900 consisting of the following three operations in the exact order shown:

1. The host must write a Transmit Command to the TxCMD register (PacketPage base + 0144h). The contents of the TxCMD register

may be read back from the TxCMD register (Register 9).

2. The host must write the frame's length to the TxLength register (PacketPage base + 0146h).

3. The host must read the BusST register (Register 18)

The information written to the TxCMD register tells the CS8900 how to transmit the next frame. The bits that must be programmed in the TxCMD register are described in Table 5.16.

Register 9, TxCMD			
Bit		**Bit Name**	**Operation**
6	7	Tx Start	
clear	clear		Start preamble after 5 bytes have been transferred to the CS8900.
clear	set		Start preamble after 381 bytes have been transferred to the CS8900.
set	clear		Start preamble after 1021 bytes have been transferred to the CS8900.
set	set		Start preamble after entire frame has been transferred to the CS8900.
8		Force	When set, the CS8900 discards any frame data currently in the transmit buffer.
9		Onecoll	When set, the CS8900 will not attempt to re-transmit any packet after a collision.
C		InhibitCRC	When set, the CS8900 does not append the 32-bit CRC value to the end of any transmit packet.
D		TxPadDis	When set, the CS8900 will not add pad bits to short frames.

Table 5.16. Tx Command Configuration

For each individual packet transmission, the host must issue a *complete* Transmit Request. Furthermore, the host must write to the TxCMD register before each packet transmission, even if the contents of the TxCMD register do not change. The Transmit Request described above may be in either Memory Space or I/O Space, as follows:

5.7.6 Transmit in Poll Mode

In poll mode, Rdy4TxiE bit (Register B "BufCFG", Bit 8) must be clear (Interrupt Disabled). The transmit operation occurs in the following order and is shown in Figure 5.12

1. The host bids for frame storage by writing the Transmit Command to the TxCMD register (memory base+ 0144h in memory mode and I/O base + 0004h in I/O mode).

2. The host writes the transmit frame length to the TxLength register (memory base + 0146h in memory mode and I/O base + 0006h in I/O mode). If the transmit length is erroneous, the command is discarded and the TxBidErr bit (Register 18, BusST, Bit 7) is set.

3. The host reads the BusST register. This read is performed in memory mode by reading Register 18, at memory base + 0138h. In I/O mode, the host must first set the PacketPage Pointer at the correct location by writing 0138h to the PacketPage Pointer Port (I/O base + 000Ah). The host can then read the BusST register from the PacketPage Data Port (I/O base + 000Ch).

After reading the register, the Rdy4TxNOW bit (Bit 8) is checked. If the bit is set, the frame can be written. If the bit is clear, the host must continue reading the BusST register (Register 18) and checking the Rdy4TxNOW bit (Bit 8) until the bit is set.

When the CS8900 is ready to accept the frame, the host transfers the entire frame from host memory to CS8900 memory using "REP" instruction (REP MOVS to memory base + 0A00h in memory mode, and REP OUT to Receive/Transmit Data Port (I/O base + 0000h) in I/O mode).

Enter Packet Transmit Process

Is TxCMD pending?

Exit: can't Issue command

Yes

Note: Issuing a command at this point will cause previous transmit frame to be lost.

No

Host Writes Transmit Command to the TxCMD Register

Host Writes Transmit Frame Length to the TxLength Register

Transmit Request

Host Reads the BusST Register (Register 18)

Rdy4 TxNOW bit = 1?

No

Polling Loop

Yes

CS8900 Commits Buffer Space to Transmit Frame

Host Writes Transmit Frame to CS8900

CS8900 Transmits Frame

Exit Transmit Process

Figure 5.12. Transmit Operation in Polling Mode

Figure 5.13. Transmit Operation in Interrupt Mode

5.7.7 Transmit in Interrupt Mode

In interrupt mode, Rdy4TxiE bit (Register B "BufCFG", Bit 8) must be set for transmit operation. Transmit operation occurs in the following order and is shown in Figure 5.13

1. The host bids for frame storage by writing the Transmit Command to the TxCMD register (memory base + 0144h in memory mode and I/O base + 0004h in I/O mode).

2. The host writes the transmit frame length to the TxLength register (memory base + 0146h in memory mode and I/O base + 0006h in I/O mode). If the transmit length is erroneous, the command is discarded and the TxBidErr, bit 7, in BusST register is set.

3. The host reads the BusST register. This read is performed in memory mode by reading Register 18, at memory base + 0138h. In I/O mode, the host must first set the PacketPage Pointer at the correct location by writing 0138h to the PacketPage Pointer Port (I/O base + 000Ah), it than can read the BusST register from the PacketPage Data Port (I/O base + 000Ch).

   After reading the register, the Rdy4TxNOW bit is checked. If the bit is set, the frame can be written to CS8900 memory. If Rdy4TxNOW is clear, the host will have to wait for the CS8900 buffer memory to become available at which time the host will be interrupted. On interrupt, the host enters the interrupt service routine and reads ISQ register (Memory base + 0120h in memory mode and I/O base + 0008h in I/O) and checks the Rdy4Tx bit (bit 8). If Rdy4Tx is clear then the CS8900 waits for the next interrupt. If Rdy4Tx is set, then the CS8900 is ready to accept the frame.

4. When the CS8900 is ready to accept the frame, the host transfers the entire frame from host memory to CS8900 memory using

"REP" instruction (REP MOVS to memory base + 0A00h in memory mode, and REP OUT to Receive/Transmit Data Port (I/O base + 0000h) in I/O mode).
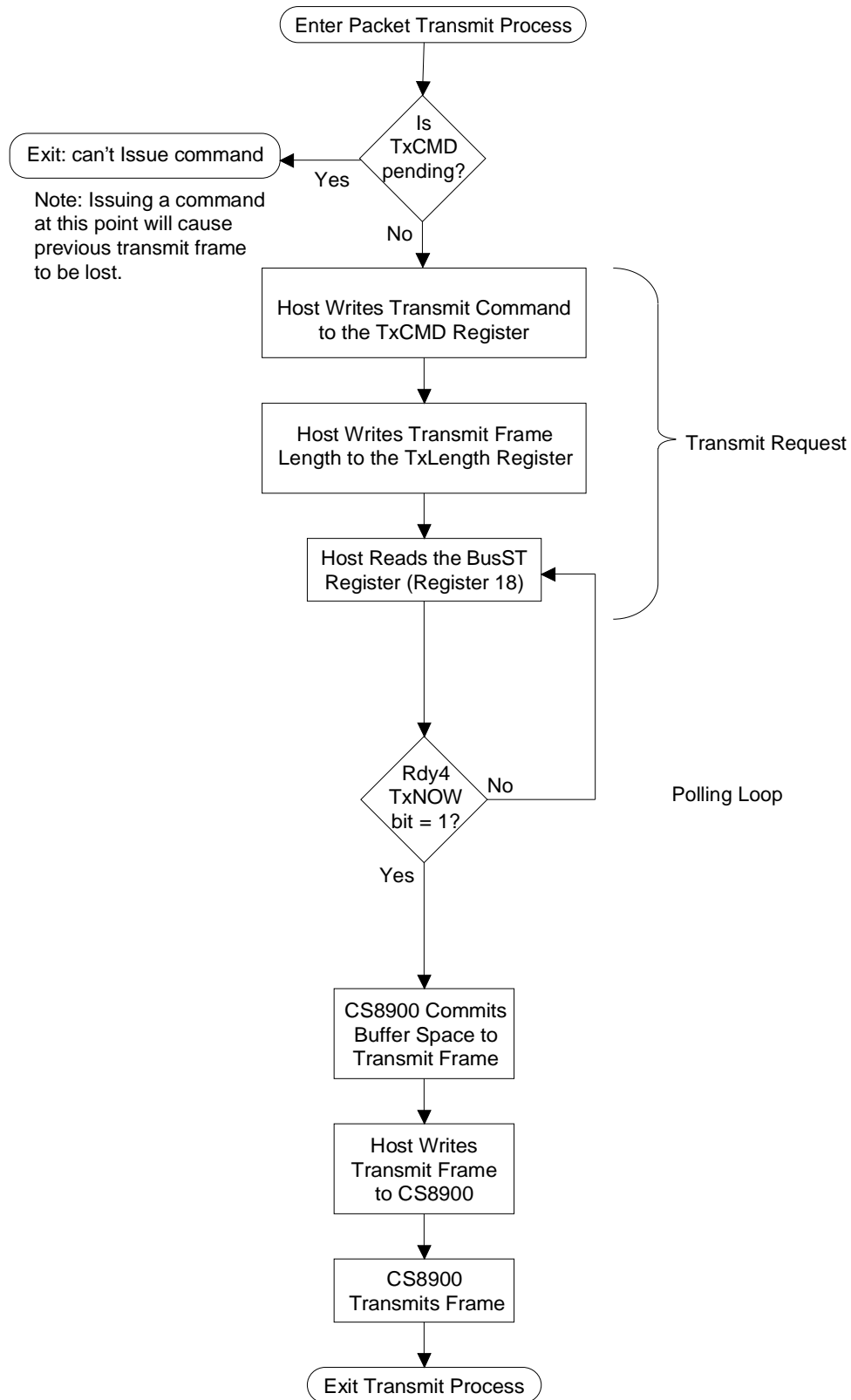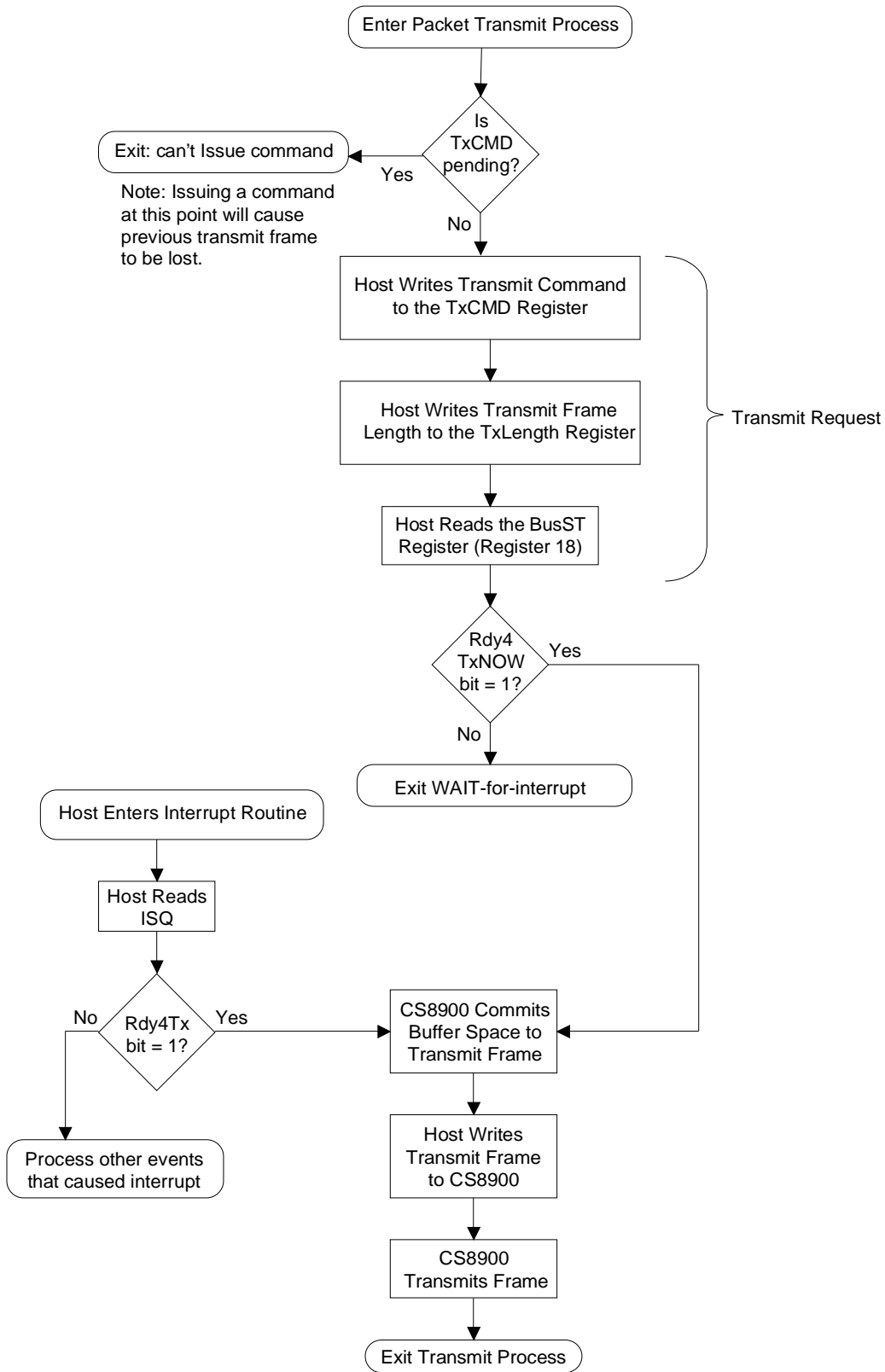
5.7.8 Completing Transmission

When the CS8900 successfully completes transmitting a frame, it sets the TxOK bit (Register 8, TxEvent, Bit 8). If the TxOKiE bit (Register 7, TxCFG, bit 8) is set, the CS8900 generates a corresponding interrupt.

5.7.9 Rdy4TxNOW vs. Rdy4Tx

The Rdy4TxNOW bit (Register 18, BusST, bit 8) is used to tell the host that the CS8900 is ready to accept a frame for transmission. This bit is used during the Transmit Request process or after the Transmit Request process to signal the host that space has become available when interrupts are not being used (i.e. the Rdy4TxiE bit (Register B, BufCFG, Bit 8) is not set). Also, the Rdy4Tx bit is used with interrupts and requires the Rdy4TxiE bit be set.

Figure 5.12 provides a diagram of error free transmission without collision.

5.7.10 Committing Buffer Space to a Transmit Frame

When the host issues a transmit request, the CS8900 checks the length of the transmit frame to see if there is sufficient on-chip buffer space. If there is, the CS8900 sets the Rdy4TxNOW bit. If not, and the Rdy4TxiE bit is set, the CS8900 waits for buffer space to free up and then sets the Rdy4Tx bit. If Rdy4TxiE is not set, the CS8900 sets the Rdy4TxNOW bit when space becomes available.

Even though transmit buffer space may be available, the CS8900 does not commit buffer space to a transmit frame until all of the following are true:

1. The host must issues a Transmit Request;

2. The Transmit Request must be successful; and,

3. Either the host reads that the Rdy4TxNOW bit (Register 18, BusST, Bit 8) is set, or the host reads that the Rdy4Tx bit (Register C, BufEvent, bit 8) is set.

If the CS8900 commits buffer space to a particular transmit frame, it will not allow subsequent frames to be written to that buffer space as long as the transmit frame is committed.

After buffer space is committed, the frame is subsequently transmitted unless any of the following occur:

1. The host completely writes the frame data, but transmission failed on the Ethernet line. There are three such failures, and these are indicated by three transmit error bits in the TxEvent register (Register 8): 16coll, Jabber, or Out-of-Window.

Or:

2. The host aborts the transmission by setting the Force (Register 9, TxCMD, bit 8) bit. In this case, the committed transmit frame, as well as any yet-to-be-transmitted frames queued in the on-chip memory, are cleared and not transmitted. The host should make TxLength = 0 when using the Force bit.

Or:

3. There is a transmit under-run, and the TxUnderrun bit (Register C, BufEvent, Bit 9) is set.

Successful transmission is indicated when the TxOK bit (Register 8, TxEvent, Bit 8) is set.

5.7.11 Transmit Frame Length

The length of the frame transmitted is determined by the value written into the TxLength register (PacketPage base + 0146h) during the Transmit Request. The length of the transmit frame may be modified by the configuration of the TxPadDis bit (Register 9, TxCMD, Bit D) and the InhibitCRC bit (Register 9, TxCMD, Bit C). Table 5.17 defines how these bits affect the length of the transmit frame. In addition, it shows which frames the CS8900 will send.

Register 9, TxCMD		Host specified transmit length at 0146h (in bytes)			
TxPadDis (Bit D)	InhibitCRC (Bit C)	3 < TxLength < 60	60 > TxLength < 1514	1514 > TxLength < 1518	TxLength > 1518
0	0	Pad to 60 and add CRC	Send frame and add CRC [Normal Mode]	Will not send	Will not send
0	1	Pad to 60 and send without CRC	Send frame without CRC	Send frame without CRC	Will not send
1	0	Send without pads, and add CRC	Send frame and add CRC	Will not send	Will not send
1	1	Send without pads and without CRC	Send frame without CRC	Send frame without CRC	Will not send

Notes: 1. If the TxPadDis bit is clear and InhibitCRC is set and the CS8900 is commanded to send a frame of length less than 60 bytes, the CS8900 pads.
2. The CS8900 will not send a frame with TxLength less than 3 bytes.

Table 5.17. Transmit Frame Length

5.8 Full duplex Considerations

The driver should not bid to transmit a long frame (i.e., a frame greater than 118 bytes) if the prior transmit frame is still being transmitted. The end of the transmission of this prior frame is indicated by a TxOK bit being set in the TxEvent register (register 8).

5.9 Auto-Negotiation Considerations

When the CS8900 is connected to an auto negotiation hub, and if auto-media detection is selected (bits 8 and 9 of register 13), then the CS8900 may not auto-select the 10BASE-T media. The cause of this situation is described in the following paragraphs.

The original IEEE 802.3 specification requires the MAC to wait until 4 valid link-pulses are received before asserting Link-OK. Any time an invalid link-pulse is received, the count is restarted. When auto-negotiation occurs, a transmitter sends FLPs (auto-negotiation Fast Link Pulses) bursts instead of the original IEEE 802.3 NLP (Normal Link Pulses).

If the hub is attempting to auto-negotiate with the CS8900, the CS8900 will never get more than 1 "valid" link pulse (valid NLP). This is not a problem if the CS8900 is already sending link-pulses, because when the hub receives NLPs from the CS8900, the hub is required to stop sending FLPs and start sending NLPs. The NLP transmitted by the hub will put the CS8900 into Link-OK.

However, if the CS8900 is in Auto-Switch mode, the CS8900 will never send any link-pulses, and the hub will never change from sending FLPs to sending NLPs.

6.0 TEST MODES

Loopback & Collision Diagnostic Tests

Internal and external Loopback and Collision tests can be used to verify the CS8900's functionality when configured for either 10BASE-T or AUI operation.

Internal Tests

Internal tests allow the major digital functions to be tested, independent of the analog functions. During these tests, the Manchester encoder is connected to the decoder. All digital circuits are operational, and the transmitter and receiver are disabled.

External Tests

External test modes allow the complete chip to be tested without connecting it directly to an Ethernet network.

Loopback Tests

During Loopback tests, the internal Carrier Sense (CRS) signal, used to detect collisions, is ignored, allowing packet reception during packet transmission.

10BASE-T Loopback and Collision Tests

10BASE-T Loopback and Collision Tests are controlled by two bits in the Test Control register: FDX (Register 19, TestCTL, Bit E) and ENDECloop (Register 19, TestCTL, Bit 9). Table 6.1 describes these tests.

AUI Loopback and Collision Tests

AUI Loopback and Collision tests are controlled by two bits in the Test Control register: AUIloop (Register 19, TestCTL, Bit A) and ENDECloop (Register 19, TestCTL, Bit 9). Table 6.2 describes these tests.

Test Mode	FDXforce	ENDECloop	Description of Test
10BASE-T Internal Loopback	1	1	Transmit a frame and verify that the frame is received without error.
10BASE-T Internal Collision	0	1	Transmit frames and verify that collisions are detected and that internal counters function properly. After 16 collisions, verify that 16coll (Register 8, TxEvent, Bit F) is set.
10BASE-T External Loopback	1	0	Connect TXD+ to RXD+ and TXD- to RXD-. Transmit a frame and verify that the frame is received without error.
10BASE-T External Collision	0	0	Connect TXD+ to RXD+ and TXD- to RXD-. Transmit frames and verify that collisions are detected and that internal counters function properly. After 16 collisions, verify that 16coll (Register 8, TxEvent, Bit F) is set.

Table 6.1. 10BASE-T Loopback and Collision Tests

Test Mode	AUIloop	ENDECloop	Description of Test
AUI Internal Loopback	1	1	Transmit a frame and verify that the frame is received without error.
AUI External Loopback	1	0	Connect DO+ to DI+ and D0- to DI-. Transmit a frame and verify that the frame is received without error (since there is no collision signal, an SQE error will occur).
AUI Collision	0	0	Start transmission and observe DO+/DO- activity. Input a 10 MHz sine wave to CI+/CI- pins and observe collisions.

Table 6.2. ADI Loopback and Collision Tests

6.1 Boundary Scan

Boundary Scan test mode provides an easy and efficient board-level test for verifying that the CS8900 has been installed properly. Boundary Scan will check to see if the orientation of the chip is correct, and if there are any open or short circuits.

Boundary Scan is controlled by the $\overline{\text{TESTSEL}}$ pin. When $\overline{\text{TESTSEL}}$ is high, the CS8900 is configured for normal operation. When TEST-SEL is low, the following occurs:

- the CS8900 enters Boundary Scan test mode and stays in this mode as long as $\overline{\text{TESTSEL}}$ is low;

- the CS8900 goes through an internal reset and remains in internal reset as long as $\overline{\text{TESTSEL}}$ is low;

- the AEN pin, normally the ISA bus Address Enable, is redefined to become the Boundary Scan shift clock input; and

- all digital outputs and bi-directional pins are placed in a high-impedance state (this electrically isolates the CS8900 digital outputs from the rest of the circuit board).

For Boundary Scan to be enabled, AEN must be low before $\overline{\text{TESTSEL}}$ is driven low.

A complete Boundary Scan test is made up of two separate cycles. The first cycle, known as the Output Cycle, tests all digital output pins and all bi-directional pins. The second cycle, known as the Input Cycle, tests all digital input pins and all bi-directional pins.

Output Cycle

During the Output Cycle, the falling edge of AEN causes each of the 17 digital output pins and each of the 17 bi-directional pins to be driven low, one at a time. The cycle begins with LINKLED and advances in order counterclockwise around the chip though all 34 pins. This test is referred to as a "walking 0" test.

The following is a list of output pins and bi-directional pins that are tested during the Output Cycle:

Pin Name	Pin #	Pin Name	Pin #
$\overline{\text{ELCS}}$	2	INTRQ1	31
$\overline{\text{EECS}}$	3	INTRQ0	32
EESK	4	$\overline{\text{IOCS16}}$	33
EEDataOut	5	$\overline{\text{MEMCS16}}$	34
DMARQ2	11	INTRQ3	35
DMARQ1	13	IOCHRDY	64
DMARQ0	15	SD0 - SD7	65-68,71-74
$\overline{\text{CSOUT}}$	17	$\overline{\text{BSTATUS}}$	78
SD08-SD15	27-24,21-18	$\overline{\text{LINKLED}}$	99
INTRQ2	30	$\overline{\text{LANLED}}$	100

The output pins not included in this test are:

Pin Name	Pin #	Pin Name	Pin #
DO+	83	TXD-	88
DO-	84	RES	93
TXD+	87	XTAL2	98

Input Cycle

During the Input Cycle, the falling edge of AEN causes the state of each selected pin to be transferred to EEDataOut (that is, EEDataOut will be high or low depending on the input level of the selected pin). This cycle begins with $\overline{\text{SLEEP}}$ and advances clockwise through each of 33 input pins (all digital input pins except for AEN) and each of the 17 bi-directional pins, one pin at a time.

The following is a list of input pins and bi-directional pins that are tested during the Input Cycle:

Pin Name	Pin #	Pin Name	Pin #
$\overline{\text{ELCS}}$	2	$\overline{\text{SBHE}}$	36
EEDataIn	6	SA0 - SA11	37-48
$\overline{\text{CHIPSEL}}$	7	$\overline{\text{REFRESH}}$	49
$\overline{\text{DMACK2}}$	12	SA12- SA19	50-54,58-60
$\overline{\text{DMACK1}}$	14	$\overline{\text{IOR}}$	61
$\overline{\text{DMACK0}}$	16	$\overline{\text{IOW}}$	62
SD08-SD15	27-24,21-18	SD0 - SD7	65-68,71-74
$\overline{\text{MEMW}}$	28	$\overline{\text{RESET}}$	75
MEMR	29	SLEEP	77

The input pins not included in this test are:

Pin Name	Pin #	Pin Name	Pin #
AEN	63	CI-	82
$\overline{\text{TESTSEL}}$	76	RXD+	91
DI+	79	RXD-	92
DI-	80	XTAL1	97
CI+	81		

After the Input Cycle is complete, one more cycle of AEN returns all digital output pins and bi-directional pins to a high-impedance state.

Continuity Cycle

The combination of a complete Output Cycle, a complete Input Cycle, and an additional AEN cycle is called a Continuity Cycle. Each Continuity Cycle lasts for 85 AEN clock cycles. The first Continuity Cycle can be followed by additional Continuity Cycles by keeping $\overline{\text{TESTSEL}}$ low and continuing to cycle AEN. When $\overline{\text{TESTSEL}}$ is driven high, the CS8900 exits Boundary Scan mode and AEN is again used as the ISA-bus Address Enable.

Figure 6.1 shows a complete Boundary Scan Continuity Cycle.

Figure 6.2 shows Boundary Scan timing.

CRYSTAL

```
┌─────────────────────────┐
│   Not in Boundary Scan  │
│       Test Mode         │
└─────────────────────────┘
```

TESTSEL switches low (AEN must be low)

```
┌─────────────────────────┐
│  ENTER BOUNDARY SCAN:   │
│   CS8900 resets, all digital │
│  output pins and bi-directional │
│   pins enter High-Z state, │
│  and AEN becomes shift clock │
└─────────────────────────┘
```

AEN switches high

```
┌─────────────────┐        AEN switches low        ┌─────────────────┐
│                 │   ─────────────────────────    │ Selected output │
│  OUTPUT CYCLE   │         34 cycles              │   goes low      │
│                 │   ─────────────────────────    │                 │
│                 │       AEN switches high        └─────────────────┘
└─────────────────┘
```

```
┌─────────────────┐        AEN switches low        ┌─────────────────┐
│                 │   ─────────────────────────    │ Selected input  │
│  INPUT CYCLE    │         50 cycles              │   copied out    │
│                 │   ─────────────────────────    │    to the       │
│                 │       AEN switches high        │   EEDataOut     │
└─────────────────┘                                │      pin        │
                                                   └─────────────────┘
```

AEN switches low

```
                    ┌─────────────────────────┐
AEN switches high   │ All digital output pins and │
                    │  bi-directional pins enters │
                    │      High-Z state       │
                    └─────────────────────────┘
```

TESTSEL switches high

```
┌─────────────────────────┐
│  EXIT BOUNDARY SCAN:    │
│   AEN becomes ISA bus   │
│     Address Enable      │
└─────────────────────────┘
```
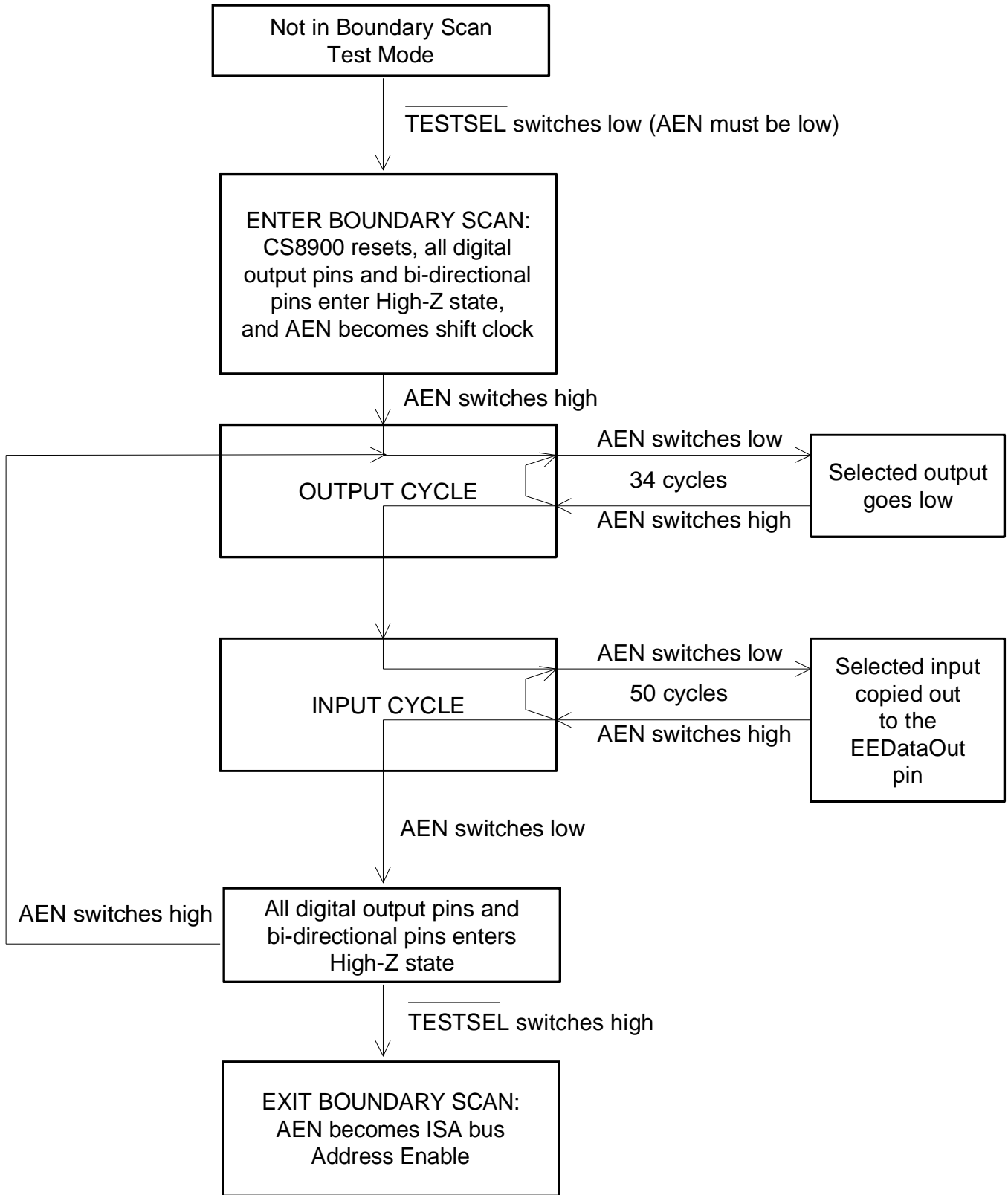
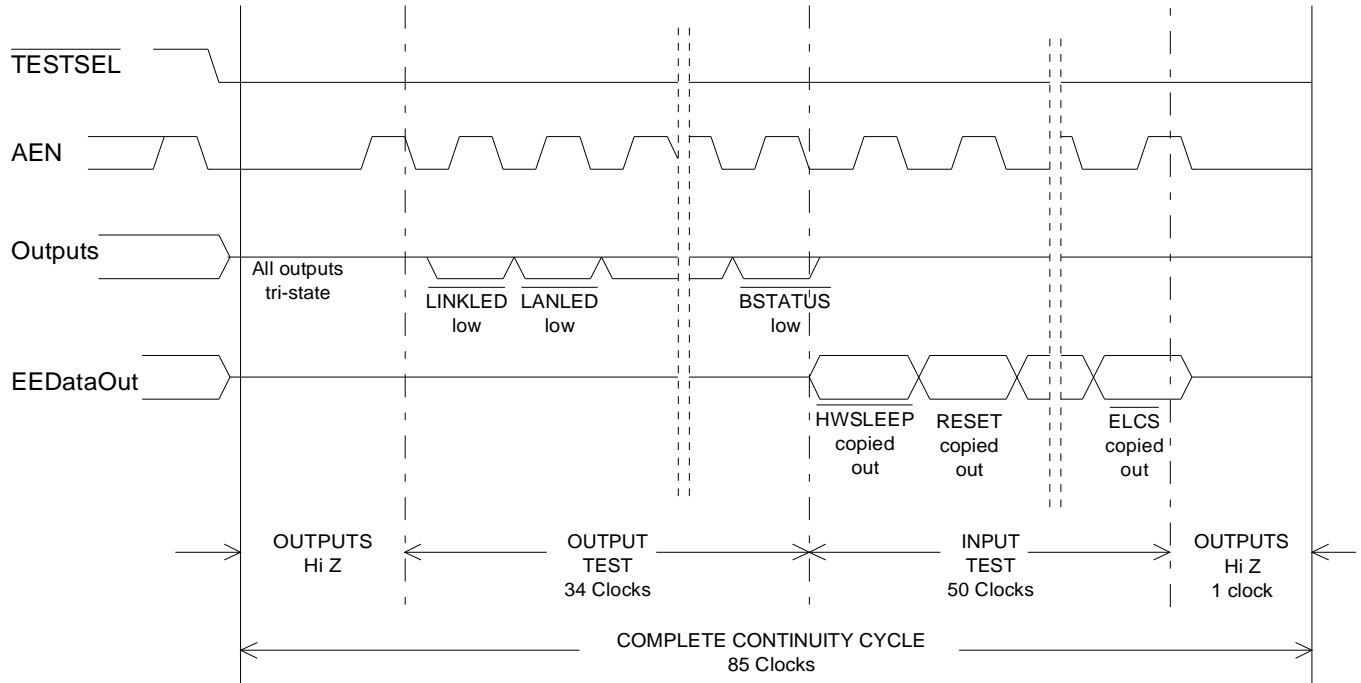**Figure 6.1. Boundary Scan Continuity Cycle**

Figure 6.2. Boundary Scan Timing

7.0 ABSOLUTE MAXIMUM RATINGS

(AVSS, DVSS = 0 V, all voltages with respect to 0 V.)

Parameter		Symbol	Min	Max	Units
Power Supply:	Digital	DV_{DD}	-0.3	6.0	V
	Analog	AV_{DD}	-0.3	6.0	V
Input Current	(Except Supply Pins)			±10.0	mA
Analog Input Voltage			-0.3	(AV_{DD+}) +0.3	V
Digital Input Voltage			-0.3	(DV_{DD+}) +0.3	V
Ambient Temperature	(Power Applied)		-55	+125	°C
Storage Temperature			-65	+150	°C

Warning: Operation at or beyond these limits may result in permanent damage to the device.
Normal operation is not guaranteed at these extremes.

8.0 RECOMMENDED OPERATING CONDITIONS

(AVSS, DVSS = 0 V, all voltages with respect to 0 V.)

Parameter		Symbol	Min	Max	Units
Power Supply:	Digital	DV_{DD}	4.75	5.25	V
	Analog	AV_{DD}	4.75	5.25	V
Operating Ambient Temperature		T_A	0	+70	°C

9.0 DC CHARACTERISTICS

Over Recommended Operating Conditions

9.1 CRYSTAL (when using external clock)				
Parameter	**Symbol**	**Min**	**Max**	**Units**
XTAL1 Input Low Voltage	V_{IXH}	-0.5	0.8	V
XTAL1 Input High Voltage	V_{IXH}	3.5	DV_{DD+} 0.5	V
XTAL1 Input Low Current	I_{IXL}	-40	-20	µA
XTAL1 Input High Current	I_{IXH}	20	40	µA
9.2 POWER SUPPLY				
Parameter	**Symbol**	**Min**	**Max**	**Units**
Power Supply Current while Active	I_{DD}		120	mA
Hardware Standby Mode Current (Note 1)	$I_{DDSTNDBY}$		1.0	mA
Hardware Suspend Mode Current (Note 1)	$I_{DDHWSUS}$		75	µA
Software Suspend Mode Current (Note 1)	$I_{DDSWSUS}$		1.0	mA

Notes: 1. With digital outputs connected to CMOS loads.

9.0 DC CHARACTERISTICS (continued.)

Over Recommended Operating Conditions

9.3 DIGITAL INPUTS and OUTPUTS						
I/O Type	**Parameter**	**Symbol**	**Conditions and Comments**	**Min**	**Max**	**Units**
OD24	Output Low Voltage	V_{OL}	I_{OL} = 24 mA		0.4	V
OD24	Output Leakage Current	I_{LL}	$0 \leq V_{OUT} \leq V_{CC}$	-10	10	μA
OD10	Output Low Voltage	V_{OL}	I_{OL} = 10 mA		0.4	V
OD10	Output Leakage Current	I_{LL}	$0 \leq V_{OUT} \leq V_{CC}$	-10	10	μA
B24	Output Low Voltage	V_{OL}	I_{OL} = 24 mA		0.4	V
B24	Output High Voltage	V_{OH}	I_{OH} = -12 mA	2.4		V
B24	Output Leakage Current	I_{LL}	$0 \leq V_{OUT} \leq V_{CC}$	-10	10	μA
B4	Output Low Voltage	V_{OL}	I_{OL} = 4 mA		0.4	V
B4	Output High Voltage	V_{OH}	I_{OH} = -2 mA	2.4		V
B4	Output Leakage Current	I_{LL}	$0 \leq V_{OUT} \leq V_{CC}$	-10	10	μA
B4w	Output Low Voltage	V_{OL}	I_{OL} = 4 mA		0.4	V
B4w	Output High Voltage	V_{OH}	I_{OH} = -2 mA	2.4		V
B4w	Output Leakage Current	I_{LL}	$0 \leq V_{OUT} \leq V_{CC}$	-20	10	μA
O24ts	Output Low Voltage	V_{OL}	I_{OL} = 24 mA		0.4	V
O24ts	Output High Voltage	V_{OH}	I_{OH} = -2 mA	2.4		V
O24ts	Output Leakage Current	I_{LL}	$0 \leq V_{OUT} \leq V_{CC}$	-10	10	μA
O4	Output Low Voltage	V_{OL}	I_{OL} = 4 mA		0.4	V
O4	Output High Voltage	V_{OH}	I_{OH} = -2 mA	2.4		V
I	Input Low Voltage	V_{IL}			0.8	V
I	Input High Voltage	V_{IH}		2.4		V
I	Input Leakage Current	I_L	$0 \leq V_{IN} \leq V_{CC}$	-10	10	μA
Iw	Input Low Voltage	V_{IL}			0.8	V
Iw	Input High Voltage	V_{IH}		2.4		V
Iw	Input Leakage Current	I_L	$0 \leq V_{IN} \leq V_{CC}$	-20	10	μA

Pin Types:

d	=	Differential Input Pair	I	=	Input	G	=	Ground
dO	=	Differential Output Pair	O	=	Output	ts	=	Tri-State
B	=	Bi-Directional with Tri-State Output	P	=	Power	w	=	Internal Weak Pullup
OD	=	Open Drain Output						

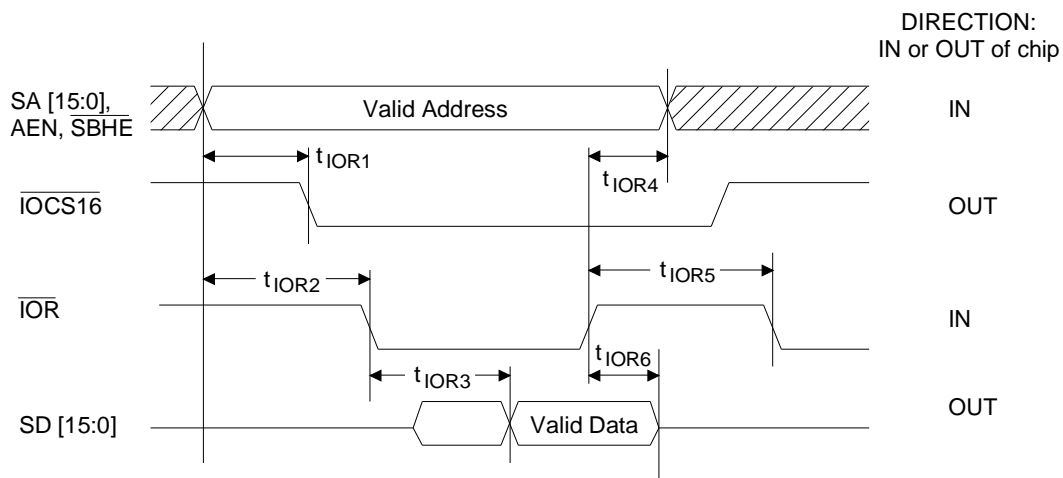Digital outputs are followed by drive in mA (Example: OD24 = Open Drain Output with 24 mA drive).

9.0 DC CHARACTERISTICS (continued)
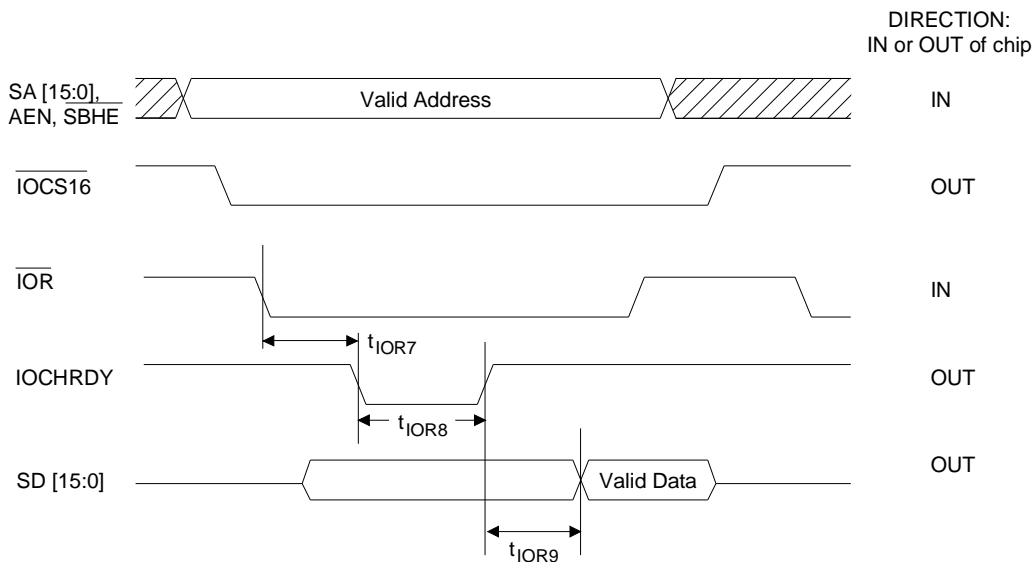
Over Recommended Operating Conditions

9.4 10BASE-T INTERFACE					
Parameter	**Symbol**	**Min**	**Typical**	**Max**	**Units**
Transmitter Differential Output Voltage (Peak)	V_{OD}	2.2		2.8	V
Receiver Normal Squelch Level (Peak)	V_{ISQ}	300		525	mV
Receiver Low Squelch Level (LoRxSquelch bit set)	V_{SQL}	125		290	mV
9.5 AUI INTERFACE					
Parameter	**Symbol**	**Min**	**Typical**	**Max**	**Units**
Transmitter Differential Output Voltage (DO+/DO- Peak)	V_{AOD}	±0.45		±1.2	V
Transmitter Undershoot Voltage	V_{AODU}			100	mV
Transmitter Output Short Circuit Current (V_{DD} or V_{SS})	I_{AOSC}			±150	mA
Transmitter Differential Idle Voltage (DO+/DO- Peak)	V_{IDLE}			40	mV
Receiver Squelch Level (DI+/DI- Peak)	V_{AISQ}	180		300	mV
Receiver Common Mode Voltage	V_{AICM}				V

10.0 SWITCHING CHARACTERISTICS

16-BIT I/O READ, IOCHRDY NOT USED				
Parameter	**Symbol**	**Min**	**Max**	**Units**
Address, AEN, SBHE active to IOCS16 low	t_{IOR1}		35	ns
Address, AEN, SBHE active to IOR active	t_{IOR2}	20		ns
IOR low to SD valid	t_{IOR3}		145	ns
Address, AEN, SBHE hold after IOR inactive	t_{IOR4}	15		ns
IOR inactive to active	t_{IOR5}	60		ns
IOR inactive to SD 3-state	t_{IOR6}		30	ns

DIRECTION:
IN or OUT of chip

SA [15:0], AEN, SBHE — Valid Address — IN

IOCS16 — OUT

t_{IOR1} t_{IOR4}

IOR — IN

t_{IOR2} t_{IOR5}

t_{IOR3} t_{IOR6}

SD [15:0] — Valid Data — OUT

16-BIT I/O READ, WITH IOCHRDY				
Parameter	**Symbol**	**Min**	**Max**	**Units**
IOR active to IOCHRDY inactive	t_{IOR7}		25	ns
IOCHRDY low pulse width	t_{IOR8}	125	175	ns
IOCHRDY active to SD valid	t_{IOR9}		0	ns

DIRECTION:
IN or OUT of chip

SA [15:0], AEN, SBHE — Valid Address — IN

IOCS16 — OUT

IOR — IN

t_{IOR7}

IOCHRDY — OUT

t_{IOR8}

SD [15:0] — Valid Data — OUT

t_{IOR9}

10.0 SWITCHING CHARACTERISTICS (continued)

16-BIT MEMORY READ, IOCHRDY NOT USED				
Parameter	**Symbol**	**Min**	**Max**	**Units**
SA [19:0], SBHE, CHIPSEL, active to MEMCS16 low	t$_{MEMR1}$		30	ns
Address, SBHE, CHIPSEL active to MEMR active	t$_{MEMR2}$	20		ns
MEMR low to SD valid	t$_{MEMR3}$		145	ns
Address, SBHE, CHIPSEL hold after MEMR inactive	t$_{MEMR4}$	0		ns
MEMR inactive to SD 3-state	t$_{MEMR5}$		30	ns
MEMR inactive to active	t$_{MEMR6}$	55		ns
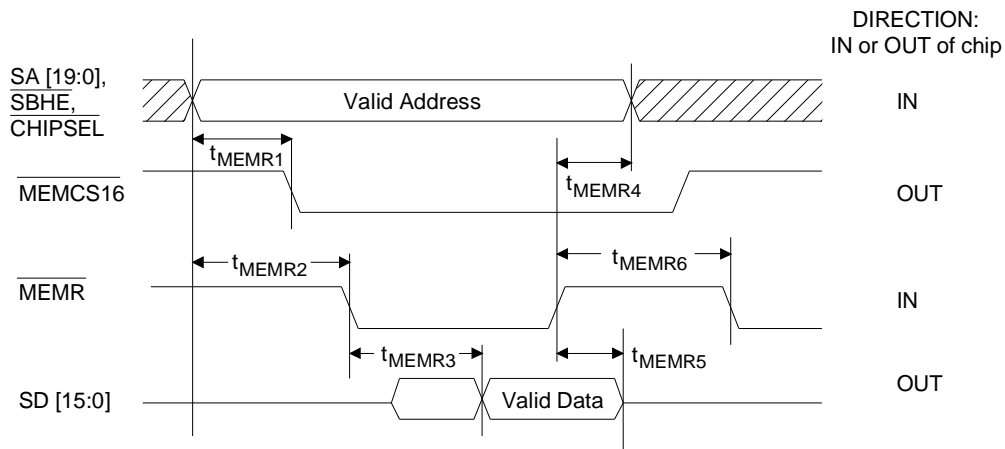
16-BIT MEMORY READ, WITH IOCHRDY				
Parameter	**Symbol**	**Min**	**Max**	**Units**
MEMR low to IOCHRDY inactive	t$_{MEMR7}$		35	ns
IOCHRDY low pulse width	t$_{MEMR8}$	125	175	ns
IOCHRDY active to SD valid	t$_{MEMR9}$		0	ns
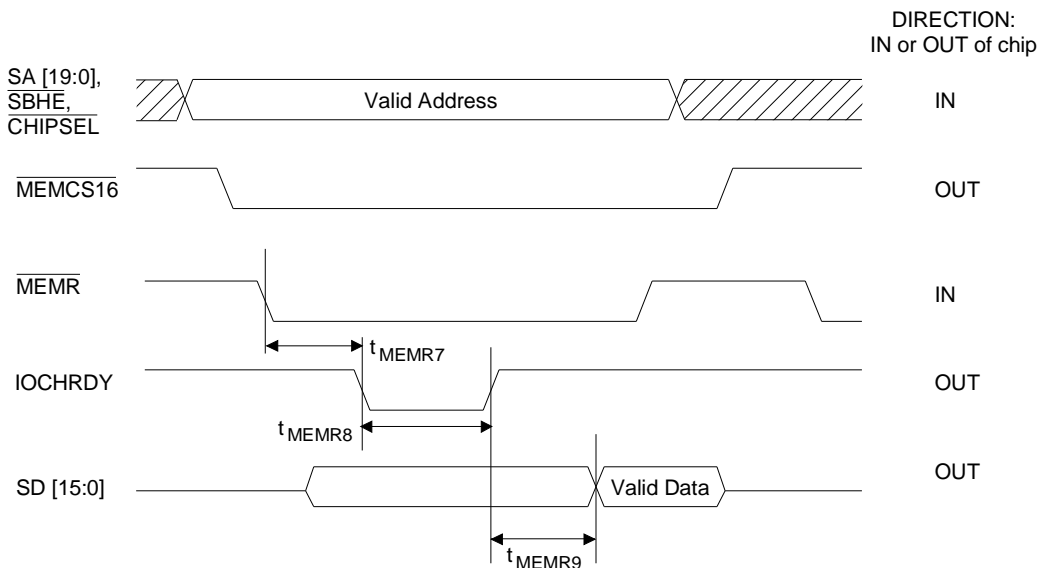
10.0 SWITCHING CHARACTERISTICS (continued)

DMA READ				
Parameter	**Symbol**	**Min**	**Max**	**Units**
AEN active to IOR low	t_{DMAR1}	20		ns
IOR low to DMARQx inactive	t_{DMAR2}		0	ns
IOR low to SD valid	t_{DMAR3}		135	ns
DMACKx, AEN hold after IOR high	t_{DMAR4}	20		ns
IOR inactive to SD 3-state	t_{DMAR5}		30	ns
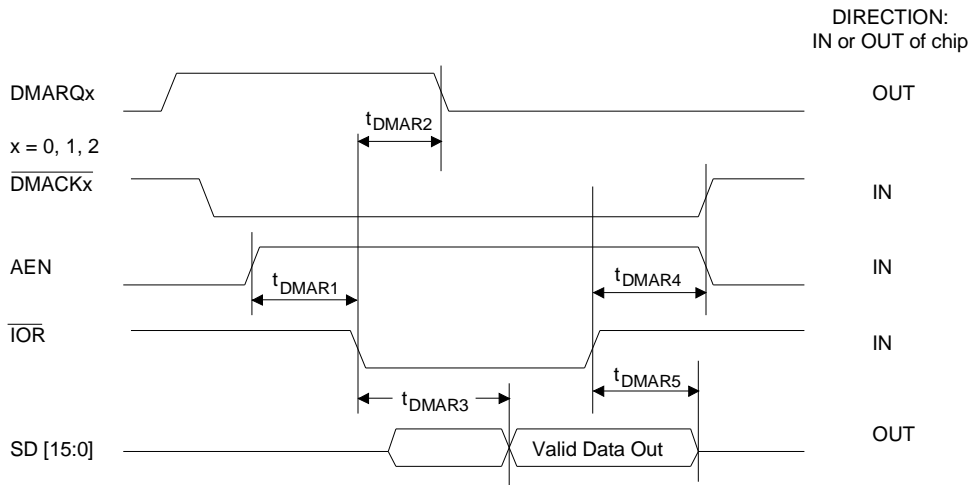
16-BIT I/O WRITE				
Parameter	**Symbol**	**Min**	**Max**	**Units**
Address, AEN, SBHE valid to IOCS16 low	t_{IOW1}		35	ns
Address, AEN, SBHE valid to IOW low	t_{IOW2}	25		ns
IOW pulse width	t_{IOW3}	110		ns
SD hold after IOW high	t_{IOW4}	6		ns
IOW low to SD valid	t_{IOW5}		10	ns
IOW inactive to active	t_{IOW6}	55		ns
Address hold after IOW high	t_{IOW7}	0		ns
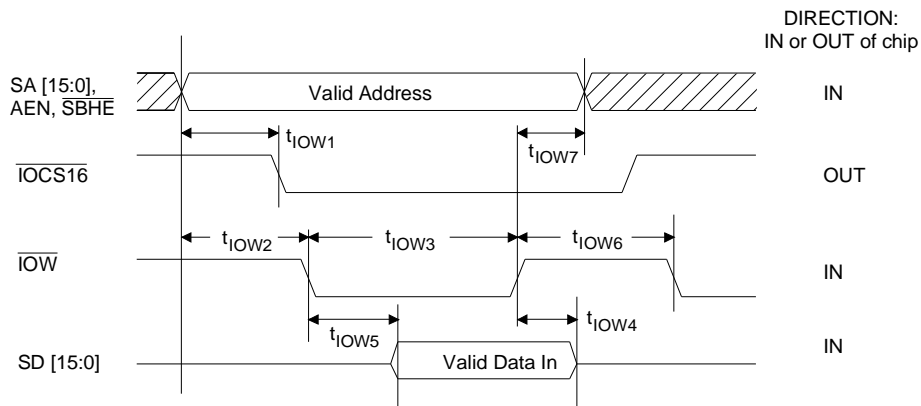
10.0 SWITCHING CHARACTERISTICS (continued)

16-BIT MEMORY WRITE				
Parameter	**Symbol**	**Min**	**Max**	**Units**
Address, SBHE, CHIPSEL valid to MEMCS16 low	t_{MEMW1}		30	ns
Address, SBHE, CHIPSEL valid to MEMW low	t_{MEMW2}	20		ns
MEMW pulse width	t_{MEMW3}	125		ns
MEMW low to SD valid	t_{MEMW4}		40	ns
SD hold after MEMW high	t_{MEMW5}	6		ns
Address hold after MEMW inactive	t_{MEMW6}	0		ns
MEMW inactive to active	t_{MEMW7}	55		ns
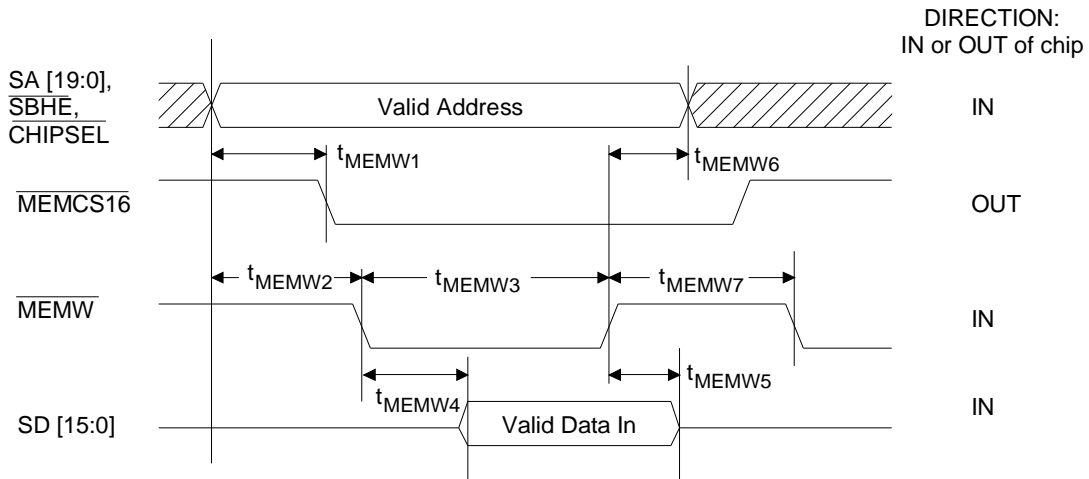
10BASE-T TRANSMIT					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
TXD Pair Jitter into 100 Ohm Load	t_{TTX1}			8	ns
TXD Pair Return to $\leq$ 50 mV after Last Positive Transition	t_{TTX2}			4.5	µs
TXD Pair Positive Hold Time at End of Packet	t_{TTX3}	250			ns
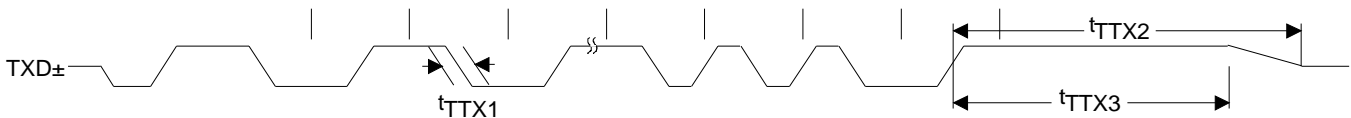
10.0 SWITCHING CHARACTERISTICS (continued)

10BASE-T RECEIVE					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
Allowable Received Jitter at Bit Cell Center	t_{TRX1}			±13.5	ns
Allowable Received Jitter at Bit Cell Boundary	t_{TRX2}			±13.5	ns
Carrier Sense Assertion Delay	t_{TRX3}	300		600	ns
Invalid Preamble Bits after Assertion of Carrier Sense	t_{TRX4}	1		2	bits
Carrier Sense Deassertion Delay	t_{TRX5}	175		230	ns
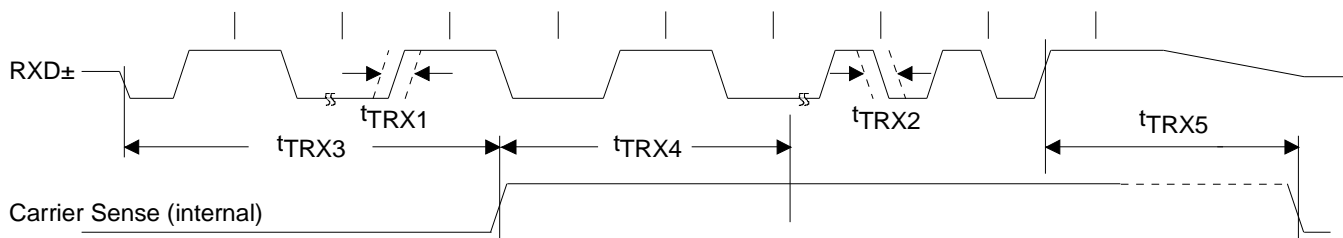
10.0 SWITHING CHARACTERISTICS (continued)

10BASE-T LINK INTEGRITY					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
First Transmitted Link Pulse after Last Transmitted Packet	t_{LN1}	8	16	24	ms
Time Between Transmitted Link Pulses	t_{LN2}	8	16	24	ms
Width of Transmitted Link Pulses	t_{LN3}	60	100	200	ns
Minimum Received Link Pulse Separation	t_{LN4}	2		7	ms
Maximum Received Link Pulse Separation	t_{LN5}	25		150	ms
Last Receive Activity to Link Fail (Link Loss Timer)	t_{LN6}	50		150	ms
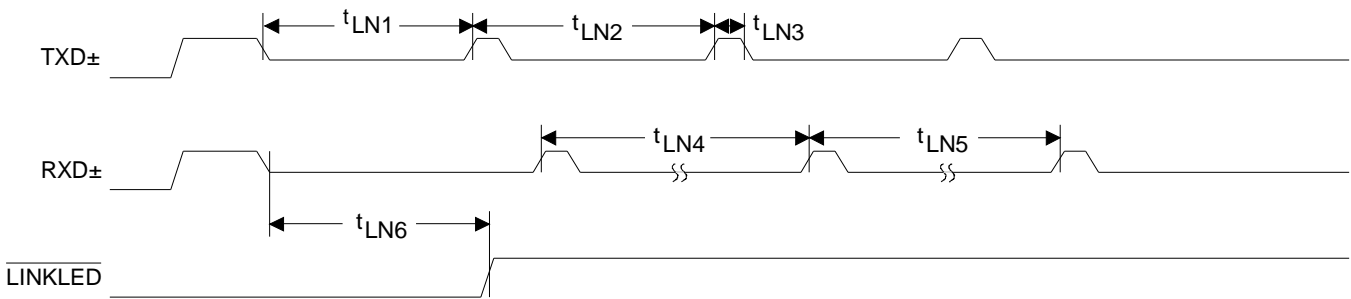
AUI TRANSMIT					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
DO Pair Rise and Fall Times	t_{ATX1}			5	ns
DO Pair Jitter at Bit Cell Center	t_{ATX2}			0.5	ns
DO Pair Positive Hold Time at Start of Idle	t_{ATX3}	200			ns
DO Pair Return to $\leq$ 40 mVp after Last Positive Transition	t_{ATX4}			8.0	μs
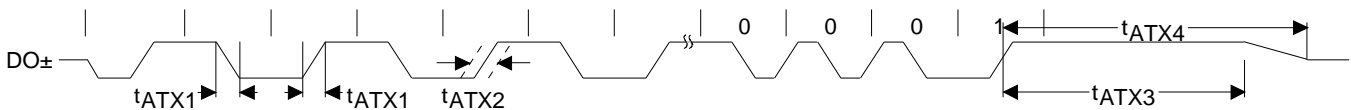
10.0 SWITCHING CHARACTERISTICS (continued)

AUI RECEIVE					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
DI Pair Rise and Fall Time	t_{ARX1}			10	ns
Allowable Bit Cell Center and Boundary Jitter in Data	t_{ARX2}			±18	ns
Carrier Sense Assertion Delay	t_{ARX3}	15		40	ns
Invalid Preamble Bits after Carrier Sense Asserts	t_{ARX4}	1		2	bits
Carrier Sense Deassertion Delay	t_{ARX5}	130	150	160	ns
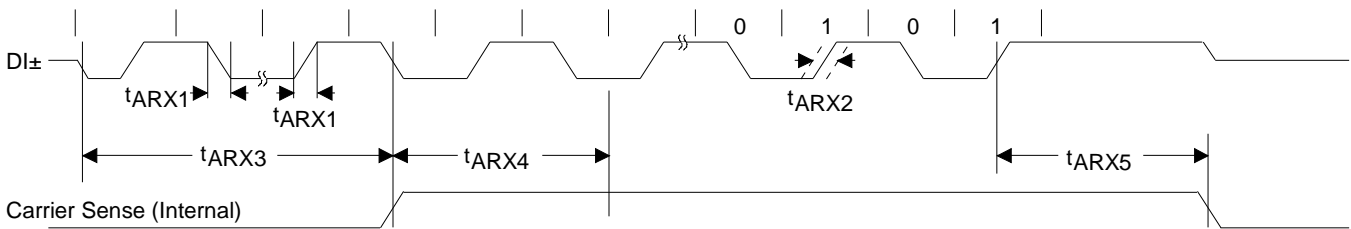
AUI COLLISION					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
CI Pair Cycle Time	t_{ACL1}	85	100	115	ns
CI Pair Rise and Fall Times	t_{ACL2}			10	ns
CI Pair Return to Zero from Last Positive Transition	t_{ACL3}	160			ns
Collision Assertion Delay	t_{ACL4}	10		30	ns
Collision Deassertion Delay	t_{ACL5}	215	225	235	ns

CRYSTAL
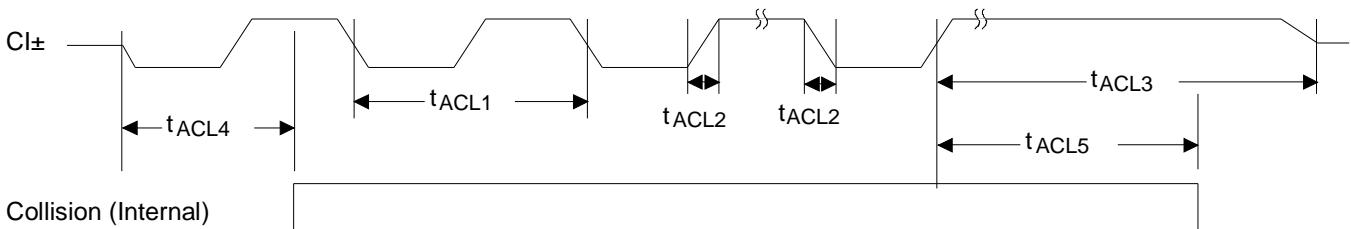
10.0 SWITCHING CHARACTERISTICS (continued)

External Boot PROM Access					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
Address active to MEMR	t_{BPROM1}	20			ns
MEMR active to CSOUT low	t_{BPROM2}			35	ns
MEMR inactive to CSOUT high	t_{BPROM3}			40	ns
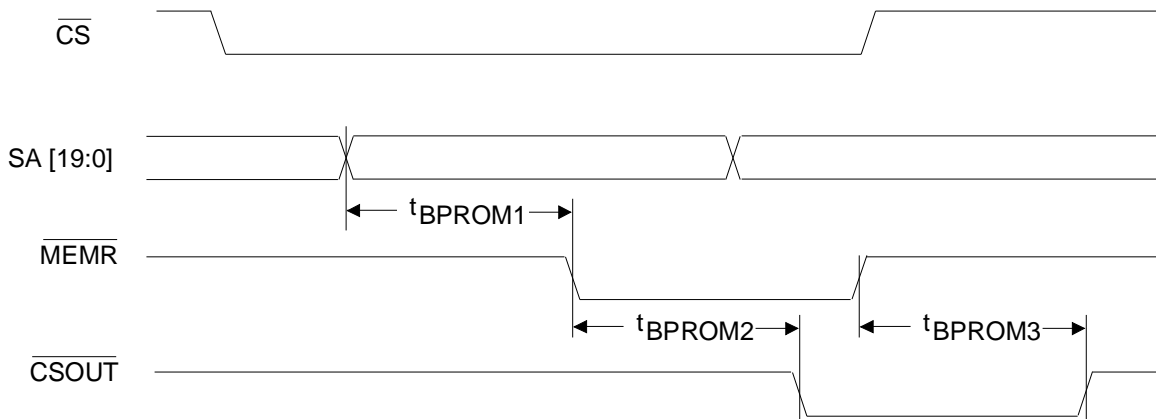
| *EEPROM*					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Units**
EESK Setup time relative to EECS	t_{SKS}	100	-	-	ns
EECS/ELCS_b Setup time wrt ↑ EESK	t_{CSS}	250	-	-	ns
EEDataOut Setup time wrt ↑ EESK	t_{DIS}	250	-	-	ns
EEDataOut Hold time wrt ↑ EESK	t_{DIH}	500	-	-	ns
EEDataIn Hold time wrt ↑ EESK	t_{DH}	10	-	-	ns
EECS Hold time wrt ↓ EESK	t_{CSH}	100	-	-	ns
Min EECS Low time during programming	t_{CS}	1000	-	-	ns
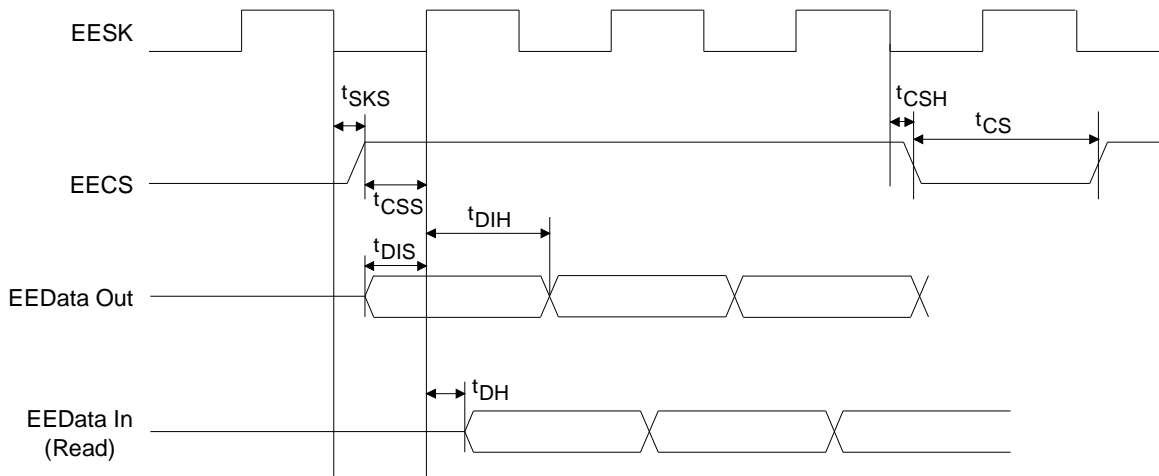
11.0 10BASE-T Wiring
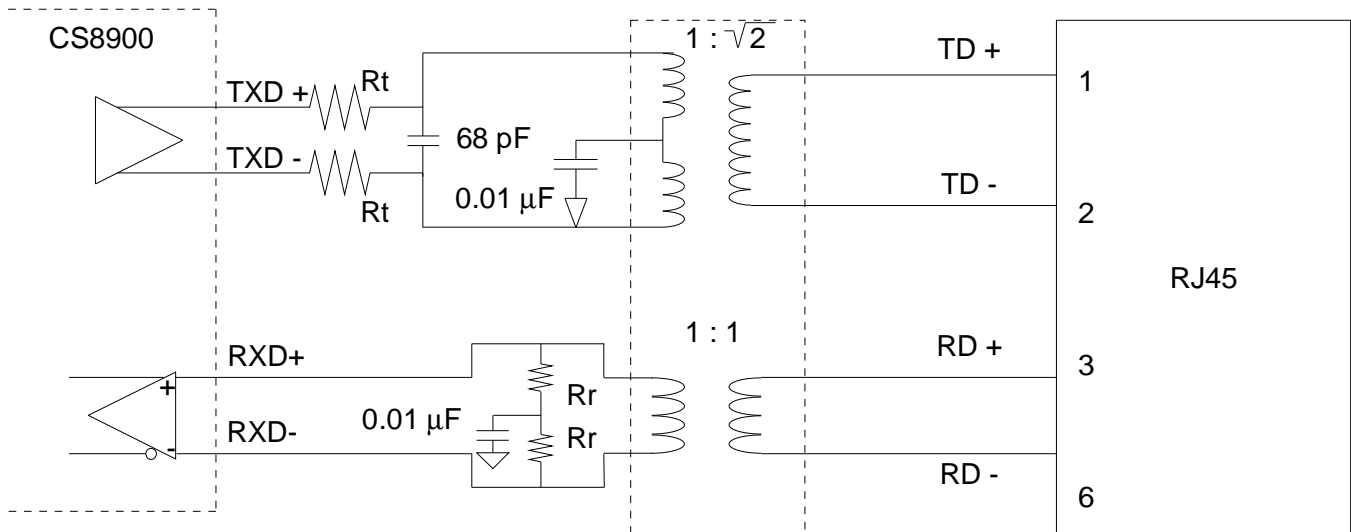
NOTES:

1. If a center tap transformer is used on the RXD+ and RXD- inputs, replace the pair of Rr resistors with a single 2xRr resistor.

2. The Rt and Rr resistors are +/- 1% tolerance.

3. The CS8900 supports 100, 120, and 150Ω unshielded twisted pair cables. The proper values of Rt and Rr, for a given cable impedance, are shown below:

| Cable Impedance (Ω)	Rt (Ω)	Rr (Ω)
100	24.3	49.9
120	30.1	60.4
150	37.4	75

12.0 AUI Wiring
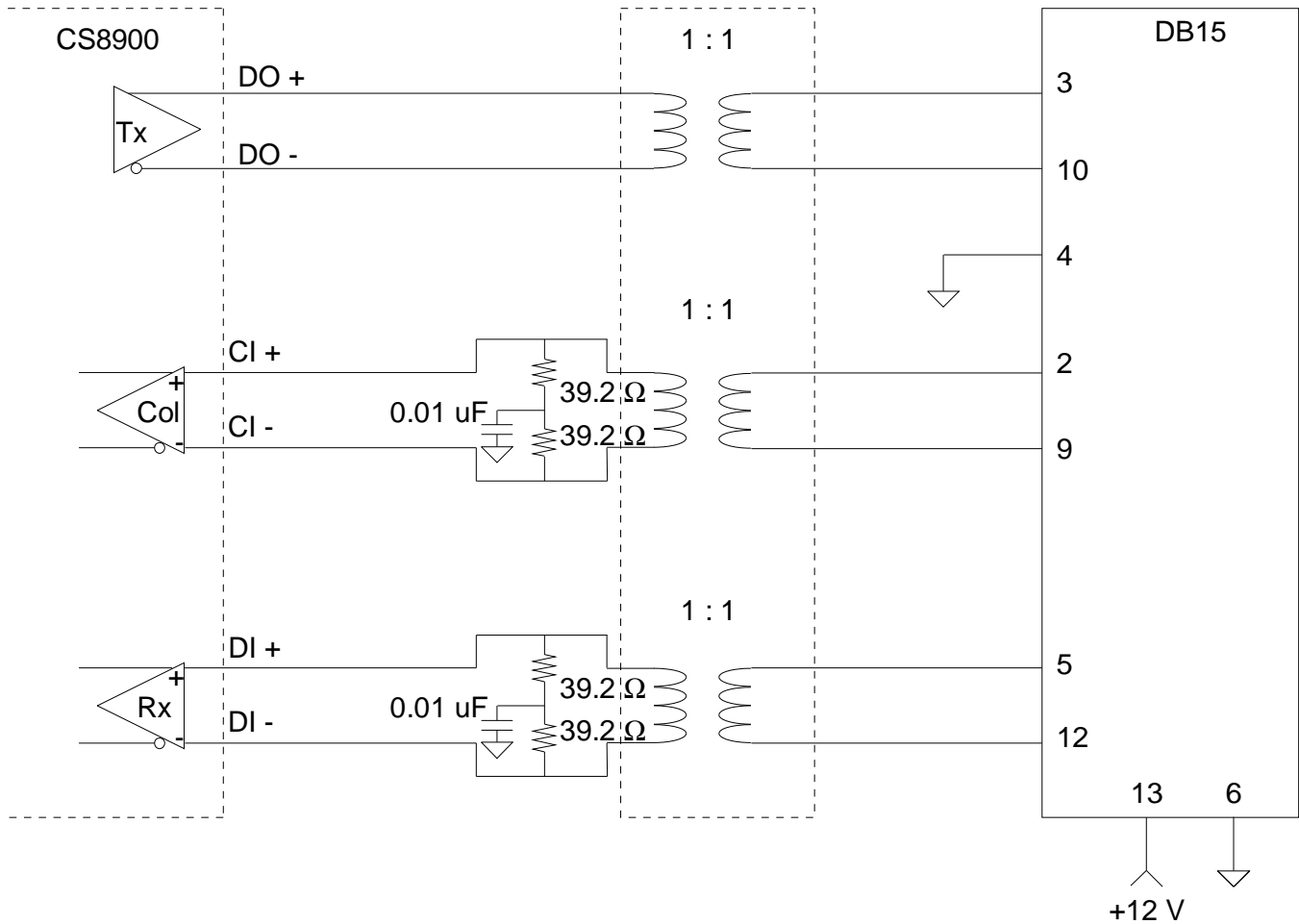
13.0 Quartz Crystal Requirements

If a 20 MHz quartz crystal is used, it must meet the following specifications:

Parameter	min	typ	max	unit
Parallel Resonant Frequency		20		MHz
Resonant Frequency Error (CL = 18 pF)	-50		+50	ppm
Resonant Frequency Change Over Operating Temperature	-40		+40	ppm
Crystal Capacitance			18	pF
Motional Crystal Capacitance		0.022		pF
Series Resistance			50	Ω
Shunt Capacitance			7	pF

14.0 PHYSICAL DIMENSIONS

100 pin TQFP
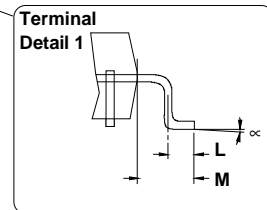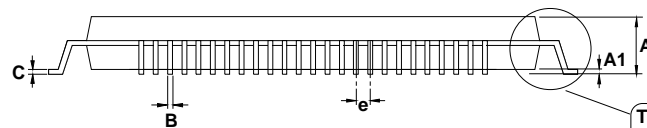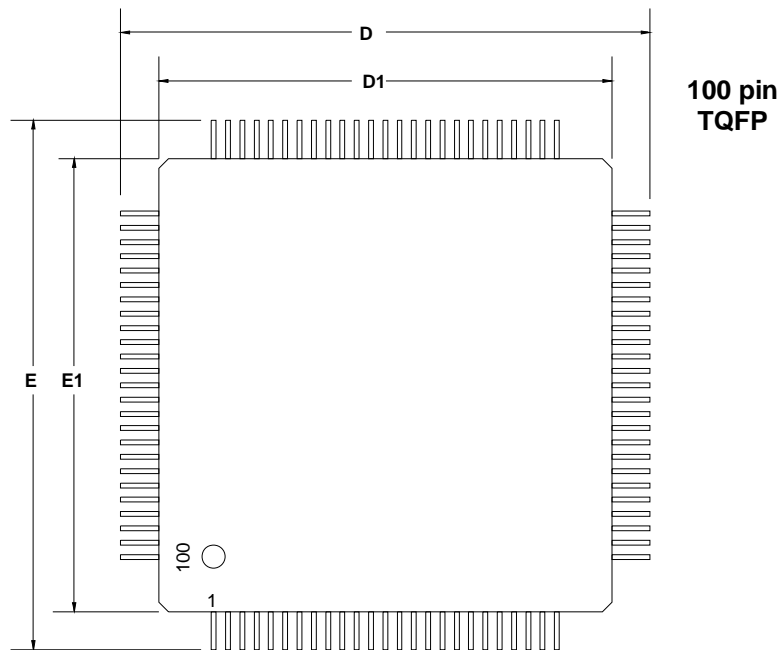
		MILLIMETERS			**INCHES**		
DIM	**MIN**	**NOM**	**MAX**	**MIN**	**NOM**	**MAX**	
A	-	-	1.66	-	-	0.065	
A1	0.00	-	-	0.000	-	-	
B	0.14	0.20	0.26	0.006	0.008	0.010	
C	0.40	0.51	0.60	0.016	0.020	0.024	
D	15.70	16.00	16.30	0.618	0.630	0.642	
D1	13.90	14.00	14.10	0.547	0.551	0.555	
E	15.70	16.00	16.30	0.618	0.630	0.642	
E1	13.90	14.00	14.10	0.547	0.551	0.555	
e	0.375	0.5	0.625	0.015	0.020	0.025	
L	0.30	0.51	0.70	0.012	0.020	0.028	
∝	0°	-	12°	0°	-	12°	
M	1.00 BSC			0.039 BSC			

Terminal Detail 1

15.0 Glossary of Terms

Acronyms

AUI	Attachment Unit Interface
CRC	Cyclic Redundancy Check
CS	Carrier Sense
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DA	Destination Address
EEPROM	Electrically Erasable Programmable Read Only Memory
EOF	End-of-Frame
FCS	Frame Check Sequence
FDX	Full Duplex
IA	Individual Address
IPG	Inter-Packet Gap
ISA	Industry Standard Architecture
LA	ISA Latchable Address Bus (LA17 - LA23)
LLC	Logical Link Control
MAC	Media Access Control
MAU	Medium Attachment Unit
MIB	Management Information Base
RX	Receive
SA	Source Address *or* ISA System Address Bus (SA0 - SA19)
SFD	Start-of-Frame Delimiter
SNMP	Simple Network Management Protocol
SOF	Start-of-Frame
SQE	Signal Quality Error
TDR	Time Domain Reflectometer
TX	Transmit
UTP	Unshielded Twisted Pair

Definitions

Cyclic Redundancy Check
   The method used to compute the 32-bit frame check sequence (FCS).

Frame Check Sequence
   The 32-bit field at the end of a frame that contains the result of the cyclic redundancy check (CRC).

Frame
   An Ethernet string of data bits that includes the Destination Address (DA), Source Address (SA), optional length field, Logical Link Control data (LLC data), pad bits (if needed) and Frame Check Sequence (FCS).

Definitions (continued)

Individual Address

The specific Ethernet address assigned to a device attached to the Ethernet media.

Inter-Packet Gap

Time interval between packets on the Ethernet. Minimum interval is 9.6 μs.

Jabber

A condition that results when a Ethernet node transmits longer than between 20 ms and 150 ms.

Packet

An Ethernet string of data bits that includes the Preamble, Start-of-Frame Delimiter (SFD), Destination Address (DA), Source Address (SA), optional length field, Logical Link Control data (LLC data), pad bits (if needed) and Frame Check Sequence (FCS). A packet is a frame plus the Preamble and SFD.

Receive Collision

A receive collision occurs when the CI+/CI- inputs are active while a packet is being received. Applies only to the AUI.

Signal Quality Error

When transmitting on the AUI, the MAC expects to see a collision signal on the CI+/CI- pair within 64 bit times after the end of a transmission. If no collision occurs, there is said to be an "SQE error". Applies only to the AUI.

Slot Time

Time required for an Ethernet Frame to cross a maximum length Ethernet network. One Slot Time equals 512 bit times.

Transmit Collision

A transmit collision occurs when the receive inputs, RXD+/RXD- (10BASE-T) or CI+/CI- (AUI) are active while a packet is being transmitted.

Acronyms Specific to the CS8900

BufCFG	Buffer Configuration - Register B
BufEvent	Buffer Event - Register C
BusCTL	Bus Control - Register 17
BusST	Bus State - Register 18
ENDEC	Manchester encoder/decoder
ISQ	Interrupt Status Queue - register 0
LineCTL	Ethernet Line Control - Register 13
LineST	Ethernet Line Status - Register 14
RxCFG	Receive Configuration - Register 3
RxCTL	Receive Control - Register 5
RxEvent	Receive Event - Register 4
SelfCTL	Self Control - Register 15
SelfST	Self Status - Register 16
TestCTL	Test Control - Register 19
TxCFG	Transmit Configuration - Register 7
TxCMD	Transmit Command - Register 9
TxEvent	Transmit Event - Register 8

Terms Specific to the CS8900

Act-Once bit

A control bit that causes the CS8900 to take a certain action once when a logic "1" is written to that bit. To cause the action again, the host must rewrite a "1".

Committed Receive Frame

A receive frame is said to be "committed" after the frame has been buffered by the CS8900, and the host has been notified, but the frame has not yet been transferred by the host.

Committed Transmit Frame

A transmit frame is said to be "committed" after the host has issued a Transmit Command, and the CS8900 has reserved buffer space and notified the host that it is ready for transmit.

Event or Interrupt Event

The term "Event" is used in this document to refer to something that can trigger an interrupt. Items that are considered "Events" are reported in the three Event registers (RxEvent, TxEvent, or BufEvent) and in two counter-overflow bits (RxMISS and TxCOL).

StreamTransfer

A method used to significantly reduce the number of interrupts to the host processor during block data transfers (Patent Pending).

Terms Specific to the CS8900 (continued)

PacketPage

A unified, highly-efficient method of controlling and getting status of a peripheral controller in I/O or Memory space.

Standby

A feature of the CS8900 used to conserve power. When in Standby mode, the CS8900 can be awakened either by 10BASE-T activity or host command.

Suspend

A feature of the CS8900 used to conserve power. When in Suspend mode, the CS8900 can be awakened only by host command.

Transfer

The term "transfer" refers to moving frame data across the ISA bus to or from the CS8900.

Transmit Request

A Transmit Request is issued by the host to initiate the start of a new packet transmission. A Transmit Request consists of the following three steps in exactly the order shown:

1. The host writes a Transmit Command to the TxCMD register (PacketPage base + 0144h).

2. The host writes the transmit frame's length to the TxLength register (PacketPage base + 0146h).

3. The host reads BusST (Register 18) to see in the Rdy4TxNOW bit (Bit 8) is set.

Sub-Terms Specific to the CS8900 Used at the End of the Term

These terms have meaning only at the end of a term;

A	Accept
CMD	Command
CFG	Configure
CTL	Control
Dis	Disable
E	Enable
h	Indicates the number is hexadecimal
iE	Interrupt Enable
ST	Status

• Notes •

• Notes •

SMART

Analog™

Smart *Analog*™ is a Trademark of Crystal Semiconductor Corporation