6. Ethernet Controller
TABLE OF CONTENTS

6. Ethernet Controller

The Ethernet controller manages the process of transmitting and receiving frames over a local area network. This controller thereby relieves the 80186 of much of the task of managing the local area network communications peripheral.

6.1 Overview

The Ethernet controller performs the following transmit functions

- monitors link activity
- obtains the Logical Link Control (LLC) data to be transmitted from the I/O region in the shared main memory
- binds LLC data into packet
- transmits the packeted data out the Ethernet port according to link management protocol
- re-enforces collisions during transmission
- reschedules transmissions after collisions
- provides transmit statistics and error reporting

The Ethernet controller performs the following receive functions:

- monitors link activity
- receives data from the Ethernet port
- analyzes and unbinds packeted data
- places the appropriately addressed and error-free data into the I/O region of the shared main memory
- provides receive statistics and error reporting

The Ethernet controller also handles network link management simultaneously with the transmit function and does the following:

- Watches for collisions
- Handles the backoff algorithm when a collision occurs
- Issues the jam when a collision occurs during transmission

6.1.1 Controller Functional Blocks

Figure 6.1 illustrates the functional blocks of the Ethernet controller and their relationship to other areas on the IOP board.

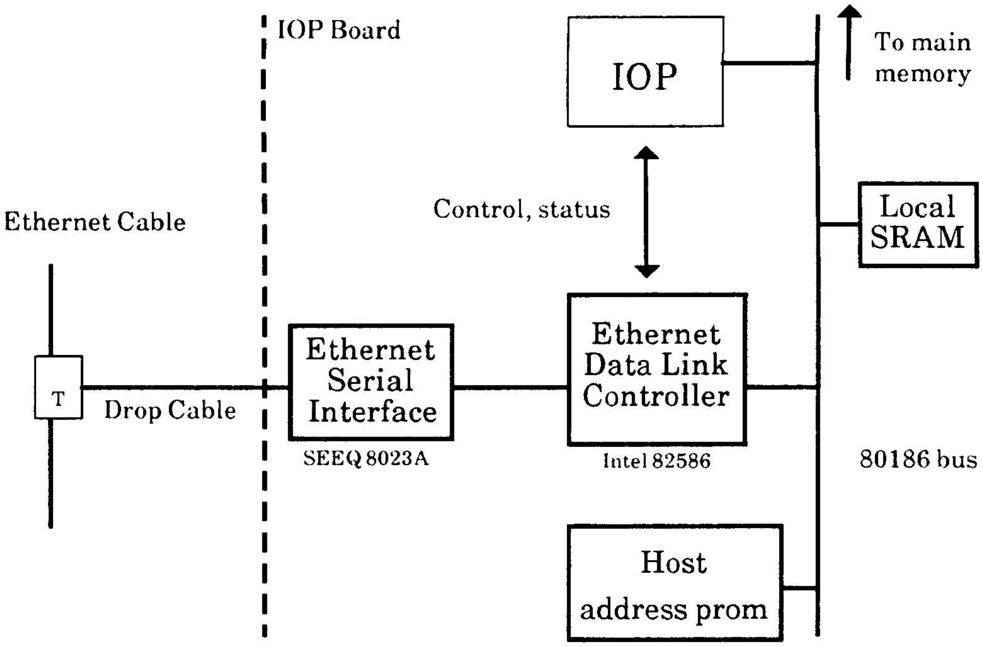

Figure 6.1. Ethernet controller block diagram

The Ethernet controller complies with the IEEE 802.3 10Base5 Standard.

Data Link Controller

The data link controller is a true coprocessor that completely manages the processes of transmitting and receiving frames from the Ethernet network. In addition, it manages the network link by monitoring the presence and quality of activity on the media and taking appropriate action. The data link controller parameters are software programmable.

The main functions of this controller are to execute commands from the IOP controller (80186) and to receive serial data from the serial interface. These two main functions are handled by two separate logical units within the data link controller: the Command Unit (CU), which is the logical unit that executes action commands from the IOP controller; and the Receive Unit (RU), which is the logical unit that receives and stores frames.

Serial Interface

The serial interface's prime functions are to perform Manchester-encoding/decoding of the outbound/inbound serial data, to provide the 10 MHz transmit and the recovered receive clock to the data link controller, and to provide the differential pair drive capability to the transceiver's drop cable.

6.1.2 Controller Relationship to Standards

International Standard
Organization (ISO)
Model

The Ethernet controller exists in both the data link layer and the physical layer with respect to the seven layer Open System Interconnect (OSI) Reference Model. The serial interface chip connects to the drop cable on the Ethernet port (differential pairs) side of the chip. The serial interface, transceiver cable, transceiver, and network cable exist in the physical layer.

The data link controller connects to the controller interface side (TTL compatible) of the serial interface chip. The boundary between the serial interface and the data link controller is the cross-over from the physical layer to the data link layer. The data link controller and its interface to the IOP bus is contained fully in the data link layer.

IEEE 802.3
LAN Standard

Note: The following description assumes that the reader has access to the IEEE 802.3 LAN Standard and relies on the standard for an understanding of the terminology and acronyms used within this section.

When considering the IEEE 802.3 LAN Standard relationship, the model to describe the network is different from the one in the ISO Reference model. In this case, the serial interface resides in the Physical Signaling (PLS) layer. It sends and receives bit-wise TTL-level[1] serial data and control signals to and from the data link controller in the Media Access Control (MAC) layer.

6.2 Hardware

The physical implementation of the Ethernet controller consists of an integrated data link controller (Intel 82586), an integrated serial interface (SEEQ 8023A), and a terminating resistor network.

The network of terminating resistors establishes a 78 ohm impedance at the serial interface differential inputs. The driving circuits of the transceiver's outputs expect to see a 78 ohm impedance at the serial interface end of the drop cable. The terminating resistors for the transmit pair are specified by the serial interface chip manufacturer for driving a 78 ohm load at the transceiver end of the drop cable.

Figure 6.2 illustrates the network. The voltage divider formed by R26 and R29, and similarly by R30 and R31, provides an "off" state voltage to the receive and collision input pairs when the transceiver is inactive on either of those pairs.

[1] The transmit and receive clocks are MOS level.

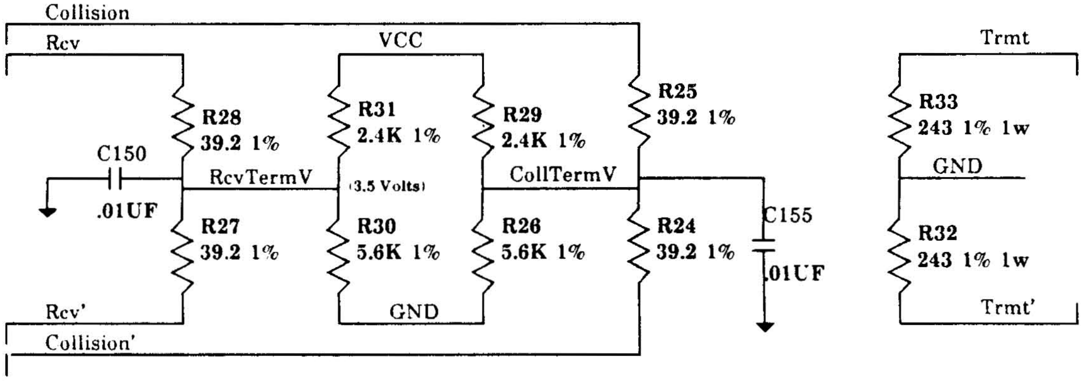

Figure 6.2. Terminating resistor network for transceiver cable

Figure 6.3 illustrates the data link controller and the Ethernet serial interface pins and signals. Tables 6.1 and 6.2 describe the signals. For a detailed description of these components, please refer to Intel's 1984 Microsystem Components Handbook and SEEQ's 8023A data sheet. For schematics refer to appendix D.

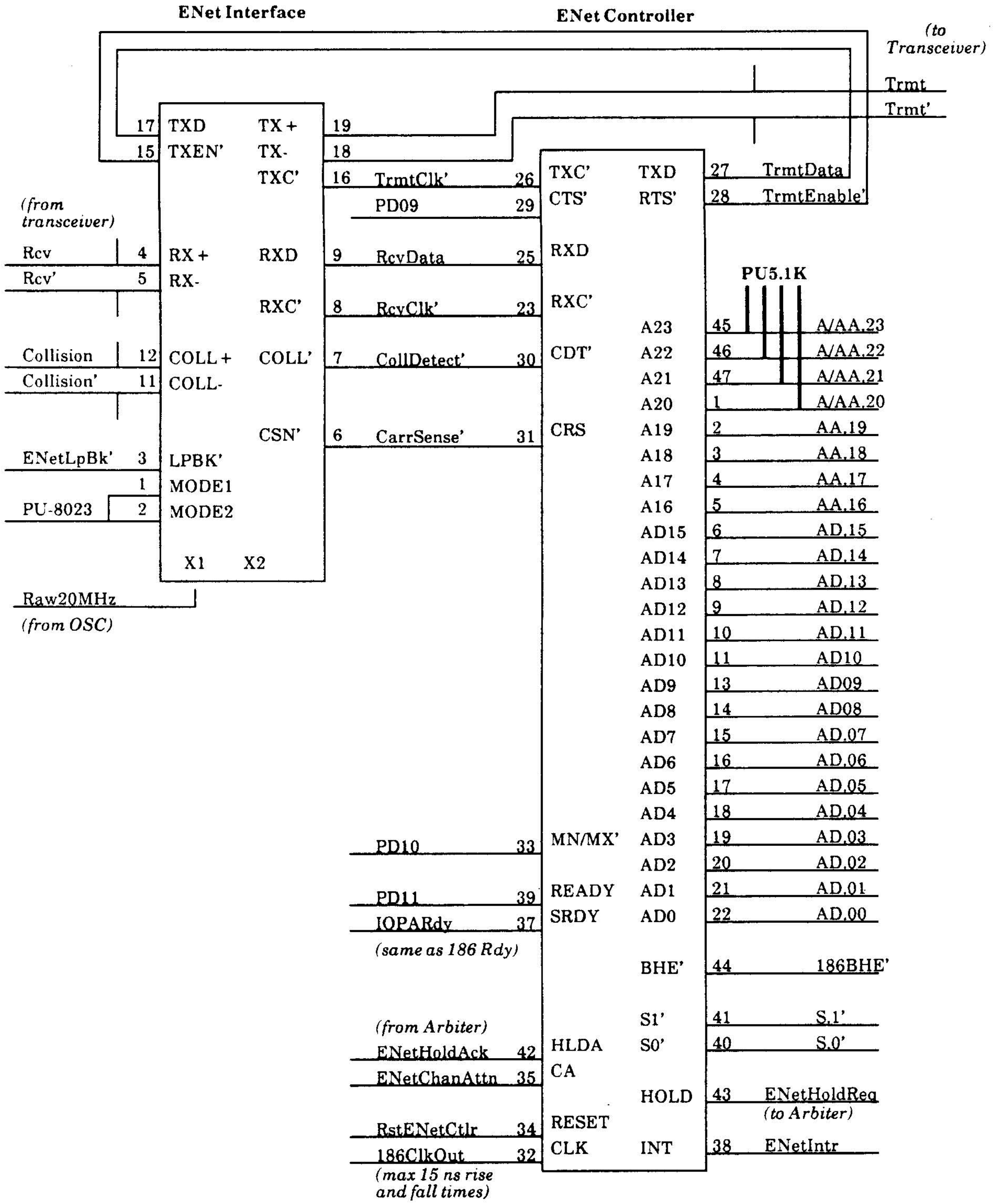

Figure 6.3. Ethernet controller and interface pins and signals

Table 6.1. Data Link Controller (82586) Pin Assignments
(Reprinted in part with permission of Intel Corporation)

Schematic Symbol	Pin No.	Type	Function
VCC	48,36		System Power: +5 volt power supply
GND	12.24		System Ground
RstENetCtlr	34	I	Reset is an active high internally synchronized signal that causes the 82586 to terminate present activity immediately. The signal must be high for at least four clock cycles. The 82586 will execute Reset within ten system clock cycles starting from Reset high. When Reset returns low, the 82586 waits for the first CA to begin the initialization sequence.
TrmtData	27	O	Transmitted serial data output signal. This signal is high when not transmitting.
TrmtClk'	26	I	Transmit Data Clock. This MOS-level signal provides timing information to the internal serial logic, depending upon the mode of data transfer. For NRZ mode of operation, data is transferred to the TxD pin on the high-to-low clock transition.
RcvData	25	I	Received Data input signal.
RcvClk'	23	I	Received Data Clock. This MOS-level signal provides timing information to the internal shifting logic depending upon the mode of data transfer. For NRZ data, the state of the RxD pin is sampled on the high-to-low clock transition.
TrmtEnable'	28	O	Request To Send signal. When low, RTS notifies an external interface that the 82586 has data to transmit. It is forced high after a Reset and while the transmit Serial Unit is not sending data.
PD09	29	I	The Clear-To-Send input has been tied low, which enables 82586 to send data. No handshaking with serial interface is required.
CarrSense'	31	I	Active low Carrier Sense input used to notify the 82586 that there is traffic on the serial link. The 82586 is configured for external Carrier Sense. External circuitry is required for detecting serial link traffic. It is internally synchronized. To be accepted, the signal must stay active for at least two serial clock cycles.
CollDetect'	30	I	Active low Collision Detect input is used to notify the 82586 that a collision has occurred. The 82586 is configured for external Collision Detect. External circuitry is required for detecting the collision. It is internally synchronized. To be accepted, the signal must stay active for at least two serial clock cycles. During transmission, the 82586 is able to recognize a collision one bit time after preamble transmission has begun.
ENetIntr	38	O	Active high interrupt request signal.
186ClkOut	32	I	An 8 MHz system clock input from the 80186.
PD10	33	I	MN/MX' input is tied low (maximum mode) which selects A22, A23, Ready, S0', S1'.
AD.00-AD.15	6-11 13-22	I/O	These lines form the time multiplexed memory address (t1) and data (t2, t3, tw, t4) bus. When operating with an 8-bit bus, the high byte will output the address during the entire cycle. AD0-AD15 are floated after a Reset or when the bus is not acquired.
AA.16-AA.18 AA.20-AA.23	1, 3-5 45-47	O	Used maximum mode only. These lines constitute 7 out of 8 most significant address bits for memory operation. They switch during t1 and stay valid during the entire memory cycle. The lines are floated after Reset or when the bus is not acquired.
AA.19	2	O	During t1, the signal forms line 19 of memory address. During t2 through t4 it is used as a status indicating that this is a Master peripheral cycle and is high. Its timing is identical to that of AD0-AD15 during write operation.
ENetHoldReq	43	O	HOLD is an active high signal used by the 82586 to request local bus mastership at the end of the current CPU bus transfer cycle, or at the end of the current DMA burst transfer cycle. In normal operation, HOLD goes inactive before HLDA. The 82586 can be forced off the bus (by HLDA going inactive) within three bus cycles at most.

- more -

Table 6.1. Data Link Controller (82586) Pin Assignments (continued)
(Reprinted in part with permission of Intel Corporation)

Schematic Symbol	Pin No.	Type	Function
ENetHoldAck	42	I	HLDA is an active high Hold Acknowledge signal indicating that the CPU has received the HOLD request and that bus control has been relinquished to the 82586. It is internally synchronized. After HOLD is detected as low, the processor drives HLDA low. Note: Connecting VCC to HLDA IS NOT ALLOWED because it will cause a deadlock. Users wanting to give permanent bus access to the 82586 should connect HLDA with HOLD. If HLDA goes inactive before HOLD, then the 82586 will release the bus (by Hold going inactive) within three bus cycles at most.
ENetChanAttn	35	I	The CA pin is a Channel Attention input used by the CPU to initiate the 82586 execution of memory resident Command Blocks. The CA signal is synchronized internally. The signal must be high for at least one system clock period. It is latched internally on high-to-low edge and then detected by the 82586.
186BHE'	44	O	The Bus High Enable signal (BHE') is used to enable data onto the most significant half of the data bus. Its timing is identical to that of A16-A23. With a 16-bit bus it is low and with an 8-bit bus it is high. Note: After Reset, the 82586 is configured to 8-bit bus.
PD11	39	I	The READY input has been tied low. This allows the READY signal internal to the 82586 to be controlled by the SRDY/ARDY' pin.
IOPArdy	37	I	The IOPArdy function is programmed in the ARDY' mode and functions as described in the 82586 pin description.
S.0', S.1'	40,41 ,	O	Maximum mode only. These status pins define the type of DMA transfer during the current memory cycle. They are encoded as follows: S1' S0' 0 0 Not used 0 1 Read Memory 1 0 Write Memory 1 1 Passive Status is active from the middle of t4 to the end of t2. They return to the passive state during t3 or during tw when READY or ARDY is high. These signals can be used by the 8288 Bus Controller to generate all memory control and timing signals. Any change from the passive state signals the 8288 to start the next t1 to t4 bus cycle. These pins are pulled high and floated after a system Reset and when the bus is not acquired.
RD'	46	O	Not used
WR'	4	O	Not used
ALE	39	O	Not used
DEN'	40	O	Not used
DT/R'	41	O	Not used

Table 6.2. Serial Interface (SEEQ 8023A) Pin Assignments

Schematic Symbol	Pin No.	Type	Function
TrmtClk'	16	O	Transmit Clock: A MOS-level 10 MHz clock output with 5 ns rise and fall times. This clock is provided to the 82586 for serial transmission.
TrmtEnable'	15	I	Transmit Enable: Selected as active low when MODE 2 is high.
TrmtData	17	I	Transmit Data: A TTL-level input signal that is directly connected to the serial data output, TXD, of the 82586.
RcvClk'	8	O	Receive Clock: A MOS-level recovered clock output. This signal is inverted. It is switched to TxC when no incoming data is present. There is a 1.25 microsecond discontinuity at the beginning of frame reception (MODE 2 is high).
CarrSense'	6	O	Carrier Sense: A TTL-level, active low output to notify the 82586 that there is activity on the coaxial cable. This signal is selected as active low (Mode 2 is high)
RcvData	9	O	Receive Data: A TTL-level output directly tied to the RXD input of the 82586. This output is high during idle (MODE 2 is high).
CollDetect'	7	O	Collision Detect: A TTL-level, active low signal which drives the CDT' input of the 82586. This signal is selected as active low (MODE 2 is high).
ENetLpBk'	3	I	Loopback'/Watchdog Timer Disable: A TTL-level control signal to enable the loopback mode. In this mode, serial data on the TXD input is routed through the 8023A internal circuits and back to the RXD output without driving the TRMT/TRMT' output pair to the transceiver cable. When LPBK' is asserted, the collision circuit will also be turned on at the end of each transmission to simulate the collision test. The on-chip watchdog timer can be disabled by applying a 10V to 16V level to this pin. (not implemented)
Trmt/ Trmt'	19-18	O	Transmit Pair: An output drive pair which generates the differential signal for the transmit pair of the Ethernet transceiver cable. Following the last transition, which is always positive at TRMT, the differential voltage is slowly reduced to zero volts. The output stream is Manchester encoded.
Rcv/Rcv'	4-5	I	Receive Pair: A differentially driven input pair which is tied to the receive pair of the Ethernet transceiver cable. The first transition on RCV is expected to be negative-going to indicate the beginning of a frame. At the end of a frame, the last transition should be positive-going. The received bit stream is assumed to be Manchester encoded.
Collision/ Collision'	11-12	I	Collision Pair: A differentially driven input pair tied to the collision-presence pair of the Ethernet transceiver cable. The collision-presence signal is a 10 MHz+/- 15 % square wave. The first transition at CLSN is expected to be negative-going to indicate the beginning of the signal; the last transition is expected to be positive-going to indicate the end of the signal.
X1- X2	13 14	I I	Clock Crystal: Driven by a 20 MHz TTL-level clock oscillator No Connection
Vcc	20		Power: 5 + 10% volts
VSS	10		Ground: Reference
MODE1	1	I	PU-8023 (tied high) 8023A is a direct replacement for SEEQ's 8002
MODE2	2	I	PU-8023 (tied high) 8032A operates in a mode compatible with the Intel 82586.

6.3 Theory of Operations

The Ethernet controller performs two major tasks: transmitting data from the transmit buffers in main memory to the network, and receiving data from the network and placing it in the receive buffers in the I/O region in main memory. This section describes the interfaces that contribute to task performance, the processes leading to task performance, the actual tasks, and diagnosis of poor performance.

6.3.1 Interfaces

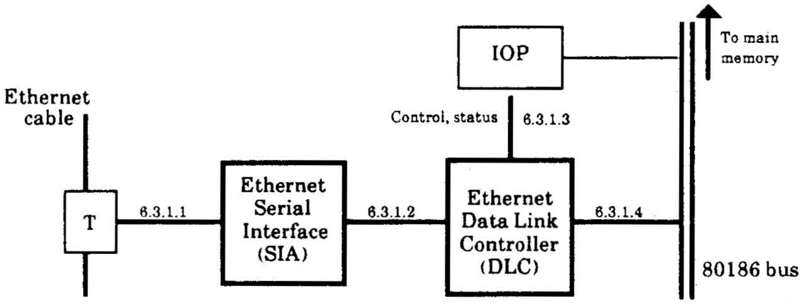

Figure 6.1 illustrated interfaces; how they function is described below.

6.3.1.1. Communicating with the Transceiver

The data link controller cannot communicate directly with the transceiver because its network communication interface requires TTL level signals with NRZ format. The transceiver is designed for Manchester-encoded differential signals from the Ethernet controller. Therefore, a serial interface between the data link controller and the transceiver is required.

The serial interface:

- converts data from Manchester-encoded differential signals to TTL compatible NRZ signals for incoming signals and reverses the process for outgoing signals.
- recovers the clock rate of the inbound data signal by use of phase locked loop circuitry. The recovered clock is sent to the data link controller to synchronize the reception of inbound data from the serial interface.
- obtains the transmit clock from a 20 MHz crystal oscillator and divides it by two before sending it out to the data link controller.
- provides the transmit clock to the data link controller for synchronized transmission of data from the data link controller to the serial interface.
- sends TTL level control signals to the data link controller to indicate Carrier Presence and Collision Detect, which the

serial interface receives from the transceiver via the differential pairs of the transceiver cable.

- accepts the Transmit Enable control signal from the data link controller. If Transmit Enable is inactive, then the transmit output pair is always inactive.
- has loopback capabilities, as discussed in the diagnostics section.

Differential Pairs

The serial interface has two pairs of differential inputs, the receive pair and the collision pair and one pair of diffential outputs, the transmit pair.

The receive pair routes activity received from the transceiver to three internal functional areas. One location is the data processing area that converts the received data from a Manchester-encoded signal to a TTL-level NRZ data signal. Another area recovers the receive clock from the incoming Manchester-encoded data and forwards the recovered clock to the data link controller. The final area senses activity on the receive pair and generates a carrier sense control signal to indicate that information. The carrier sense control signal is fed back to the data link controller even when the serial interface senses its own transmission. Therefore, the data link controller is capable of detecting its own activity on the network.

The collision pair receives a 10 MHz square wave differential signal from the transceiver when a collision level is sensed on the network. In turn, the serial interface outputs a collision detect control signal to the data link controller indicating that a collision has occurred. A carrier sense control signal is also generated when a collision is detected, even if the collision causes a full cancellation of bit transitions on the receive pairs.

The transmit pair sends Manchester-encoded transmit data to the transceiver.

6.3.1.2. Communicating with the SIA and DLC

The data link controller must work with the serial interface in order to have the proper data format when receiving from or transmitting to the network. The serial interface also generates control signals, depending on the activity on its differential inputs, to indicate to the data link controller the nature of the activity on the network. The interface between the data link controller and the serial interface consists of two data lines and five control lines.

Transmit Interface

The transmit data line is the serial data line from the data link controller to the serial interface for the transfer of the TTL-level NRZ data.

The transmit control lines are the transmit clock and transmit enable lines. The transmit clock input on the data link controller is supplied by the serial interface at one-half the 20 MHz clock frequency that is supplied to the serial interface chip by a crystal oscillator. The data link controller outputs a transmit enable signal to the serial interface through its RTS' pin. This pin goes low when the data to be transmitted is ready to be sent to the serial interface.

The serial interface has a 25 ms watchdog timer that is reset when the transmit enable signal goes inactive. The watchdog timer disables the differential pair outputs if the timer is allowed to expire. When enabled by the transmit enable control line, the serial interface receives the TTL-level serial data from the data link controller and converts it to Manchester-encoded serial data and outputs it on the transmit differential pair to the transceiver.

Receive Interface

The receive data line is the serial data line from the serial interface to the data link controller for the reception of the TTL-level NRZ data.

The receive control lines are the receive clock and the carrier sense control lines. The receive clock input on the data link controller is driven by the serial interface. The serial interface outputs on the RxC' line either the recovered clock when receiving data or the transmit clock when not receiving data. During clock switching from the recovered clock to the transmit clock, RxC' may stay low for 200 ns maximum. At the beginning of a frame reception, the RxC' line will be held low for 1200 ns maximum while the phase lock loop acquires the data clock.

The carrier sense control signal is generated and outputed to the data link controller when the serial interface detects activity on the receive differential pair. This also includes activity due to sensing its own transmission. The carrier sense control signal is also enabled during collisions which is sensed by activity on the collision differential pair.

Collision Interface

The remaining control line is for signaling the data link controller that a collision has been detected on the network. The serial interface generates and outputs the collision detect control signal when it senses the 10 MHz square wave on its collision differential pair.

6.3.1.3. Communicating with the IOP

The data link controller and the IOP controller communicate via a "mailbox" in the I/O region of shared memory known as the System Control Block (SCB). When the IOP controller wants the data link controller to process a command, the following occurs:

1) The IOP controller puts the command information in the SCB and activates the Channel Attention (CA) control line, which is linked to the data link controller.

2) When the CA control line goes active and the data link controller has completed any higher priority tasks, the data controller will request the IOP bus by activating the Hold Request (HOLD) control line.

3) Once the data link controller has the bus (indicated by receiving the hold acknowledge signal from the bus arbiter), it will read the SCB that has the Command Unit command (CU START) for it to activate the Command unit and to fetch the Action Command information from the Command List (CBL).

Note: The SCB contains the pointers to the CBL.

4) The data link controller then reads the CBL and executes the appropriate Action Commands. The possible Action Commands are:
```
NOP, Setup Individual Address, Configure,
Setup Multicast Address, Transmit,
Time Domain Reflectometry, Diagnose,
Dump.
```
An explanation of these commands is given in the Intel LAN Components User's Manual.

5) The data link controller then reads in the SCB, updates the SCB status, and clears the SCB command word, so the IOP controller knows that the data has been accepted.

The data link controller also passes information to the IOP controller via the SCB. For example, after the data link controller receives and stores a data frame into the I/O region of shared memory, it posts a "Frame Received" interrupt status bit in the SCB. The data link controller then activates the INT control line, which is directly connected to the IOP controller, to indicate that it should read the SCB. The IOP controller then reads the SCB and will be informed of a received data frame and the status of the reception.

6.3.1.4. Communicating with the IOP Bus

Because the data link controller is a coprocessor, it requires independent use of the IOP bus. The data link controller's bus interface unit uses all the local resources for bus activity controlled by the status lines, S0 and S1.

The data path for the data link controller is 16 bits wide. The data link controller performs reading/writing from/to full word locations on even addresses only. The data link controller treats memory as a high (D15-D8) and low (D7-D0) bank of 512K 8-bit words addressed in parallel by A19-A1.

Of the 24 total address lines, the lower 16 address lines are multiplexed with the 16 data lines.

The bus interface unit control signals associated with bus arbitration are hold request (ENetHoldReq) and hold acknowledge (ENetHoldAck). A bus arbiter circuit is used in order to control the access to the IOP bus among the various processors that may need the use of it. (See section 4, Bus Arbiter and Mode Control, for detailed arbiter discussion.)

The data link controller has a built-in DMA controller which is capable of a maximum data transfer rate of 4 megabytes per second. Each memory transfer cycle requires four clock cycles.

System memory is accessed by several different processors (e.g., Mesa processor, graphics processor, IOP controller, data link controller) on three separate buses which are multiplexed onto one system memory bus by system memory arbitration logic. If other processors attempt access simultaneously with an attempt by the data link controller, then the arbitration logic generates an SRDY signal that lengthens access time to system memory for the data link controller. The SRDY signal causes a minimum of one wait state to be inserted into the data link controller's memory operation. When the processor that has the system memory bus completes its memory operation, the SRDY signal goes inactive and the data link controller's memory operation is completed.

For local IOP memory, no wait states are required during data link controller memory operations. The SRDY signal is kept inactive.

The data link controller requires a 20 ns minimum set up time before the hold acknowledge signal. It will not start any data transfers until two clock periods plus the set up time after the hold acknowledge signal arrives.

Bus Operation

The data link controller performs data transfers on one of four on-chip DMA channels: Receive DMA channel; Transmit DMA channel; Command Unit input/output channel; Receive Unit input/output channel.

The DMA unit is controlled by the Command Unit. The receive and transmit DMA channels operate in bursts (up to 16 bytes) while the other two channels operate in single word mode.

The DMA channels may work simultaneously. For example, the CU may do a prefetch of the next Transmit Buffer Descriptor while the transmit DMA channel is reading in data to the transmit FIFO.

The same is true for the prefetch operation in the receive mode. To prevent data under/overruns during simultaneous operation, at least two data reads/writes are performed between any two CU/RU operations.

Another situation arises when simultaneous receive and transmit DMA channel operations occur when a receive frame arrives while the Command Unit was setting up a transmission. The two channels operate simultaneously, with equal priority, by bus cycle interleaving until the transmit FIFO is filled. When the receive frame ends, the transmit resumes.

Normally the data link controller releases the bus after an operation but there are four exceptions. The exceptions are:

- RBD prefetch - The bus is not released until the entire prefetch operation is completed.
- TBD prefetch - The bus is not released until the entire prefetch operation is completed.
- Transmit or receive buffer switching.
- Receive end of frame processing.

When the last exception occurs, it may hold the bus with no data transfers for a maximum of 146 clock cycles. This hold allows the data link controller to process and post the receive status before the expiration of the Inter Frame Spacing (9.6 microseconds minimum) and possible reception of the next receive frame.

6.3.2 Processes Leading to Data Transmission and Reception

Initializing the Ethernet controller configures the data link controller and prepares the data link controller for frame reception. Booting the Ethernet controller activates it.

Before the initialization process occurs, a hardware reset must be issued to the data link controller. This reset clears all the major internal flags, sets the CU and RU to the idle state, and configures the data link controller parameters to the default values. A RESET is generated either by a master reset that clears the reset latch or by a selective reset, where the IOP controller loads a zero into the appropriate bit of the reset latch.

After the data link controller is reset, it remains idle until a Channel Attention signal from the IOP controller occurs, triggering the data link controller's initialization process. The initialization process is explained below.

6.3.2.1. Initialization

The data link controller reads the System Configuration Pointer (SCP) from memory to obtain the address of the Intermediate System Configuration Pointer (ISCP) and to determine if the data link controller is to interface an 8- or 16-bit data bus.

The ISCP contains the base address and address offset of the SCB. A bit in the ISCP also indicates that the data link controller is in the initialization process. This bit is cleared after the ISCP is read.

The data link controller expects to find the SCP in the fixed memory address locations of FFFFF6H to FFFFFFH. In the Dove system, these actual physical address locations are in main memory.

Note: To allow the IOP board to perform the boot process and the IOP diagnostics without the need of any other system boards installed, the SCP and the ISCP must reside in local IOP memory during these processes.

Local EPROM has the address range of 0FC000H to 0FFFFFH. Local SRAM has the address range of 000000H to 03FFFFH. The SCP is actually in the EPROM in the memory locations 0FFFF6H to 0FFFFFH and the ISCP has been placed in the local SRAM at the address pointed to by the SCP.

During the boot process, the entire I/O region resides in local SRAM; the ISCP is loaded with a SCB address that resides in that local SRAM.

Added logic circuitry uses the six high order address bits along with the ENetHoldAck signal to generate the chip select for the local EPROM. This circuitry allows the data link controller to access the local EPROM (0FFFF6H to 0FFFFFH) when it outputs the address for the high end of main memory (FFFFF6H to FFFFFFH). This EPROM indicates that the data link controller has control of the bus.

The initial reset, discussed above, of the data link controller is generated by a master reset. The master reset also clears another latch with a control bit that disables main memory. This reset prevents the overdrive of the IOP data bus by having two memory devices (local memory and main memory) responding to the data link controller's memory request.

When the data link controller has completed its initialization process, it signals the IOP controller with the interrupt control line.

6.3.2.2. Successful Boot

When the IOP controller successfully completes the boot routines, it generates another data link controller reset by loading a zero in the appropriate location in the reset control register.

The IOP controller also clears the inhibit bit in the latch that disables the main memory. The data link controller reset process is repeated but this time the SCB base and offset addresses in the ISCP are located in the shared main memory. Therefore, the I/O region is moved from local memory to main memory for normal system activities.

Notes:

1. The IOP controller is capable of simultaneously generating the chip select for the EPROMs and the inhibit bit for the main memory directly by the use of the programmable on-chip chip select pins (IOPUCS').

2. The data link controller has both an external hardware and an internal software reset. The software reset should not be used unless called by a software routine that properly sets up the ISCP and inhibits the system memory before issuing the RESET-CA sequence to the data link controller. Remember that a hardware master reset automatically generates both the data link controller reset and the main memory inhibit.

6.3.3 Data Transmission and Net Management

Transmitting data is the process wherein:

- Logical Link Control (LLC) data is fetched by the data link controller from shared memory in octets;
- The LLC data is encapsulated to form a packet with the other pertinent information; such as the preamble, including the Start Frame Delimiter and the CRC data;
- The packet is then transmitted out the data link controller through the serial interface to the transceiver in serial bit-wise format, provided that the network is available.

To transmit packets, the data link controller accepts LLC data from shared memory as instructed by the IOP controller (80186) via a direct control line. The data link controller is configured to buffer mode during initialization. In this mode, the data link controller expects to find the source address, destination address, and type field in the LLC data. The data link controller then adds the necessary information to convert the LLC data into a properly formatted packet for serial bit-wise transfer to the serial interface. The data link controller accepts transmit status information from the LLC layer.

6.3.3.1 Transmission

1) The LLC data is placed in the transmit buffer by Mesa processor. The transmit buffer can be anywhere in main memory and may not be contiguous.

2) The IOP controller writes to the System Control Block (SCB) the command (CU-START) for the Command Unit (CU) to start executing the Command List (CBL). It also places the pointer to the CBL into the SCB.

3) The IOP controller also sets up the next free command block of the CBL with the Transmit command, the pointer to the Transmit Buffer Descriptor (TBD), and the pointer to the following command block in the link.

4) The IOP controller then generates a Channel Attention (CA) signal to indicate to the data link controller that it needs to read the SCB for possible action.

5) The data link controller reads the SCB after completing any other higher priority activities. From the SCB, it determines by the CU-START command that it should activate the CU and obtains the pointer to the CBL. The CU then starts executing the Action Commands as they are linked in the CBL.

6) The CU eventually reads in the CBL command block which instructs the data link controller to begin the Transmit operation. Due to the transmit operation, the pointer to the first TBD and the pointer to the next command block are read into internal registers from command block with the Transmit command.

7) The data link controller then fetches the first TBD which contains the pointer to the transmit buffer and the pointer to the next TBD and begins a DMA burst of data from the transmit buffer to the transmit FIFO.

8) The data link controller monitors link activity. If the network is idle when the data link controller is ready to begin transmitting, then it defers for the interframe spacing time of 96 bit times. It then transmits 64 bits of preamble including the two Start Frame Delimiter (SFD) bits (the last two bits of the preamble are ones) and the stored LLC data that is waiting in the transmit FIFO.

   Note: The source address, destination address and the type fields are included in the LLC data being loaded from the transmit buffer to the transmit FIFO.

9) When the transmit FIFO empties to a preset threshold, the data link controller acquires the bus and fetches more data from the transmit buffer pointed to by the initial TBD or the remainder of the transmit buffer pointed to by the subsequent TBDs until the FIFO is full again or no more data for that frame is available (e.g., data from last TBD of the present frame is fetched).

10) After all the LLC data is transmitted out of the FIFO, a 32-bit CRC code, which is generated during transmission, is appended to the end of the frame and is transmitted. At that point, the transmission of the packet ceases.

11) When the serial interface no longer receives carrier from its own transmission, it deasserts the carrier sense signal and will not reassert the carrier sense signal for a 4.5 microsecond inhibit period, regardless of the state of the receive or collision pair during that period.

    Note: If a collision signal occurs before the carrier sense signal is deasserted, then the serial interface will maintain the carrier sense signal until the collision is no longer present.

12) When the transmission is complete, the data link controller writes the status of the transmission to the SCB.

6.3.3.2. Network Management

The data link controller is responsible for maintaining proper CSMA/CD activity for the DTE according to the IEEE 802.3 standard. To accomplish this, the data link controller must be able to monitor activity and/or collisions on the network.

Steps 8 through 12 (above) are performed assuming that the data link controller has successfully obtained the network and has not collided during that process. If a collision occurs during transmission, then the data link controller aborts the transmission and takes action according to the link management protocol described in section 6.3.3.2.

If the network is active when transmission is ready, then the data link controller defers to the passing frame by delaying the transmission process until the network becomes idle (no carrier) and then proceeds with steps 8 through 12. Note that the controller will wait the appropriate interframe spacing time shown in step 8 and then will begin the transmit process of the remaining steps regardless of sensing carrier at that time.

Note: The present Intel data link controller chip used on the IOP board does not provide the optional redefer as specifed in the note in section 4.2.3.2.1 of the IEEE 802.3 LAN standard.

Transmission Completion

After transmission is started, the data link controller attempts to completely transmit the entire frame. When the transmission is complete, the data link controller notifies the IOP controller by writing the transmission status to the SCB and activating the INT control line.

Transmission Abortion

The transmission process is aborted if the transmit FIFO becomes empty before the DMA can fetch more data (DMA underrun), loss of the carrier sense signal occurs, the presence of the collision detect signal is detected, the maximum number (15) of collision retries is exceeded, or the IOP controller issues a CU-ABORT command.

The status, which includes the cause of the abortion, is then reported to the IOP controller. If the cause for aborting transmission was due to a collision, then the data link controller is responsible for enforcing the collision by transmitting 32 bits of jam pattern after the preamble.

After the jam pattern is transmitted, the data link controller is issued a retransmission priority, according to a backoff algorithm specified by the IEEE 802.3 standard.

The backoff algorithm calculates a random number between 0 and some maximum number determined by an internal formula which incorporates the number of transmission retries already made. That number is the number of slot times (512 bit times) that the data link controller must defer until it can begin the retransmission process.

If the random number is zero, then the data link controller has the highest priority and begins transmitting after the carrier sense goes inactive and the interframe spacing has elapsed. If the random number is one or more, then the data link controller waits one or more slot times and then waits the interframe spacing period and begins transmitting. If the cause for aborting was the IOP controller issuing an CU-ABORT command, then the data link controller transmits a jam pattern to cause a CRC mismatch.

Collisions

The transmission of a packet is aborted if a collision or a loss of carrier signal is detected by the data link controller. If the data link controller aborts the normal transmission activity after the preamble has been transmitted, then it continues to transmit a 32 bit jam pattern, reschedules transmission according to the back off algorithm, and tries to retransmit the packet to the maximum number of retries which is set into the data link controller during the initialization.

If a collision occurs during the preamble, then the transmission of the preamble is completed and then the 32 bit jam pattern is added. Therefore, a minimum transmission is 96 bits. If the collision occurs in the last 11 bits, then the jam pattern is not transmitted. If the collision occurs in the last bit, then it is not even reported in the end of transmission status that the data link controller outputs to the SCB.

After transmission begins, there is a period of time, called the slot time, when the transmitting station can experience a collision. After the slot time, no collisions should occur in a properly operating network.

The slot time is primarily determined by the round-trip propagation delay of a maximum length network. Also added to the slot time are other factors, such as repeater regeneration of preamble (i.e., repeater transmits more bits than are received) and worst case start-up, collision and jam of a second station at the maximum allowable distance from the original transmitting station.

In the IEEE 802.3 standard the slot time is defined to be 512 bit times (not including preamble). In a properly operating network, any transmitting station can assume that if it transmits 512 bits of data (again, excluding preamble), then it has acquired the network and will not experience a collision. Conversely, any receiving station that has received 512 bits of data (excluding preamble) will assume that it is receiving a valid packet and will continue to process it. Collisions occurring before that time generate a collision fragment of less than 512 bits on the network, and all receiving stations disregard the fragment. A collision occurring after that time is a late collision and indicates that the network is not operating properly.

Note: For a thorough understanding of time vs. distance considerations of the network, an exhaustive study of worst-case scenarios, using space/time diagrams, is strongly recommended.

6.3.4 Data Reception

Receiving data is the process wherein:

- the signal is received by the serial interface from the transceiver in bit-wise serial format;
- converted and forwarded to the data link controller;
- stripped of non-data bits (preamble including the Start Frame Delimiter);
- checked for an appropriate destination address, and if appropriately addressed, the data is stored temporarily in the receive FIFO;
- moved in octets to the receive buffer in the I/O region;
- and finally the IOP controller is interrupted to indicate that a frame was received.

To receive packets, the data link controller assembles the data received from the PLS layer into octets which when concatenated make up the MAC frame. The data link controller then passes these frames to the next higher layer, the LLC layer. It performs this task by storing the octets of the inbound frame in shared memory and then alerting the IOP controller of its availability. The data link controller also passes receive status information to the LLC layer.

6.3.4.1. Initializing

When the IOP controller is initialized, it establishes a receive data buffer area in the I/O region and writes initialization commands to the SCB.

The IOP controller then issues a Channel Attention signal to indicate to the data link controller that it should read the SCB.

Within the SCB, the data link controller is informed, via the command list, to configure itself and to ready the Receive Unit. From this point the data link controller constantly monitors the network for activity.

6.3.4.2. Receiving a Frame

The carrier sense circuit within the serial interface detects activity on its differential receive pair and triggers a TTL-level output signal. The signal is called the Carrier Sense signal. This signal indicates to the data link controller that the serial interface has detected activity on the network.

The serial interface takes in the data signal received on its differential receive pair inputs. This signal is then passed to a decoder circuit. The decoder has a phase locked loop circuit that recovers the synchronization clock from the incoming Manchester-encoded data stream for the data link controller's receive clock input.

This synchronization requires a number of bit times until the phase locked loop circuit locks onto the frequency. The synchronization loss is absorbed during the preamble portion of the frame being received.

The decoder also converts the data from Manchester-encoded data to NRZ data; the data is then forwarded on to the data link controller via the receive data output pin.

Notes:
1) The carrier sense signal is also activated when a collision signal is received from the transceiver.
2) The receive clock is inverted and there is a 1.25 microsecond discontinuity at the beginning of frame reception.
3) The receive data output is high when idle.

Data Link Controller

After the data link controller is initialized, the Receive Unit constantly monitors the receive data input. The serial input converts data and recovers the synchronization clock, which it then forwards to the data link controller.

When a new packet is received, a small number of bits are lost in the preamble portion of the packet. The remainder of the preamble (up to and including the beginning of frame, BOF, flag) is discarded.

The destination address is then accepted and scrutinized for an address match. The DTE will only accept packets with the proper individual address, a proper multicast address, or a broadcast address, unless configured in Promiscuous mode (i.e., accepts any packets regardless of type of address).

If there is a match, then the data link controller will store all the received data to the appropriate locations in the Receive Frame Area of the I/O region. If no address match is made, then the data link controller becomes ready to receive the next frame.

The process begins when the data link controller's Receive Unit places the incoming serial data into the receive FIFO. When the FIFO fills to a programmable threshold level of bytes, the data link controller requests use of the system bus by sending a hold request to the bus arbiter.

When the bus is released to the data link controller, the data link controller does a DMA transfer of the data, in octets, from the FIFO to the Receive Frame Area. When the FIFO is emptied, the data link controller releases the system bus, but will request it again when the FIFO reaches the receive threshold or when the end of the incoming data is received.

If the FIFO fills to limit before the DMA can transfer out the data, then a receive FIFO overrun occurs and the data is lost. The data link controller continues this cycle until there is no more data to transfer from the FIFO to the Receive Frame Area or until an error is encountered; for example, CRC violation, alignment, no resources, DMA overrun.

At the end of the frame reception and FIFO data transfer process, the data link controller posts a "frame received" interrupt bit in the SCB and interrupts the IOP controller, indicating that a received frame is in memory.

Fragmented Packet

If a frame containing less than the minimum number of bytes (64) is received, then the data link controller assumes the frame to be a fragment resulting from a collision. When a collision fragment is received, the data link controller disregards the fragment and reclaims the area in the I/O region associated with the fragment.

6.3.5 Diagnostics

The data link controller has the diagnostic capabilities that are summarized in Table 6.3. For detailed information on diagnostics, please refer to Intel's 1984 LAN Components User's Manual.

Table 6.3. DLC Error Reporting Capabilities

Transmission	Reception (after address filtering)
1. Transmission successful	1. Reception successful
2. Transmission unsuccessful a. Lost Carrier b. DMA underrun c. Excessive number of collisions*	2. Reception unsuccessful a. CRC error: well-aligned frame b. Alignment error: CRC error with misaligned frame c. Frame too short (less than 512 bits) d. DMA overrun e. Out of buffers**
3. Statistics: a. Number of collisions b. Number of deferred transmissions	3. Statistics: (if frame exceeds minimum frame length) a. CRC errors: the number of frames with CRC error and were aligned. b. Alignment errors: the number of frames with CRC error and were misaligned c. No resources: number of correct frames lost due to lack of memory resources. d. Overrun errors: number of frames lost due to DMA overruns.

Note: *The data link controller will also report loss of Clear-to-Send, but that pin is tied high on the IOP board.

**The data link controller will only report these events if in 'Save Bad Frame' mode. Otherwise the data link controller will only update the statistics counters. The data link controller will not report a misaligned frame with a correct CRC.

6.3.5.1. Hardware Diagnostics

The data link controller chip is capable of the following hardware related diagnostics:

- Time Domain Reflectometry
- Internal and External loopback
- SQE test

6.3.5.2. Software Diagnostics

The data link controller also has the following software commands that may be used for diagnostic purposes:

- Dump command: causes the data link controller to write over 100 bytes of its internal registers to memory
- Diagnose command: triggers an internal self-test procedure of backoff related registers and counters.