intel®

THE COMPLETE VLSI LAN SOLUTION

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

The following are trademarks of Intel Corporation and may only be used to identify Intel Products:

BXP, CREDIT, i, ICE, I^2 ICE, iCS, i_m, iMMX, Insite, INTEL, int_el, Intelevision, Intellec, $int_eligent$ Identifier™, $int_eligent$ Programming™, Intellink, iOSP, iPDS, iRMS, iSBC, iSBX, iSXM, Library Manager, MCS, Megachassis, Micromainframe, MULTIBUS, Multichannel™ Plug-A-Bubble, MULTIMODULE, PROMPT, Promware, RMX/80, RUPI, System 2000, and UPI, and the combination of ICE, iCS, iRMX, iSBC, MCS, or UPI and a numerical suffix.

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

* MULTIBUS is a patented Intel bus.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Department
3065 Bowers Avenue
Santa Clara, CA 95051

Table of Contents

1.0 INTRODUCTION: THE COMPLETE SOLUTION FOR LOCAL AREA NETWORKS

The key to providing a complete solution for local area networks (LANs) is VLSI—integrating critical system level functions and providing the performance, reliability and flexibility required for cost-effective designs. By combining it's advanced process technology, experience in communications peripherals and growth in LAN technology, INTEL has developed the first complete VLSI solution for LANs, the 82586 Local Communications Controller (LCC). And, combined with the 82501 Ethernet* Serial Interface (ESI) and readily available transceivers, users now have a complete solution for the Ethernet physical and data link layers.

System level functions, previously supported by the host CPU, have been integrated on-chip. The 82586 implements the entire frame transmission and reception process, including managing transmit and receive buffers in memory, without CPU intervention. A powerful set of diagnostics and error reporting facilities is also available for effective fault detection and isolation. Applications flexibility is achieved through the controller's programmable network parameters. Designers are able to tailor the 82586 parameters for emerging LAN applications such as "serial backplanes" within a cabinet and local serial connections of workstation peripherals such as keyboards or printers.

Fully compatible with the Ethernet and proposed IEEE 802 specifications, the 82586 LCC and 82501 ESI represent the first complete VLSI solution for Ethernet and other emerging CSMA/CD LAN applications.

*Ethernet is a trademark of Xerox Corporation.

2.0 LOCAL AREA NETWORK SYSTEM OVERVIEW

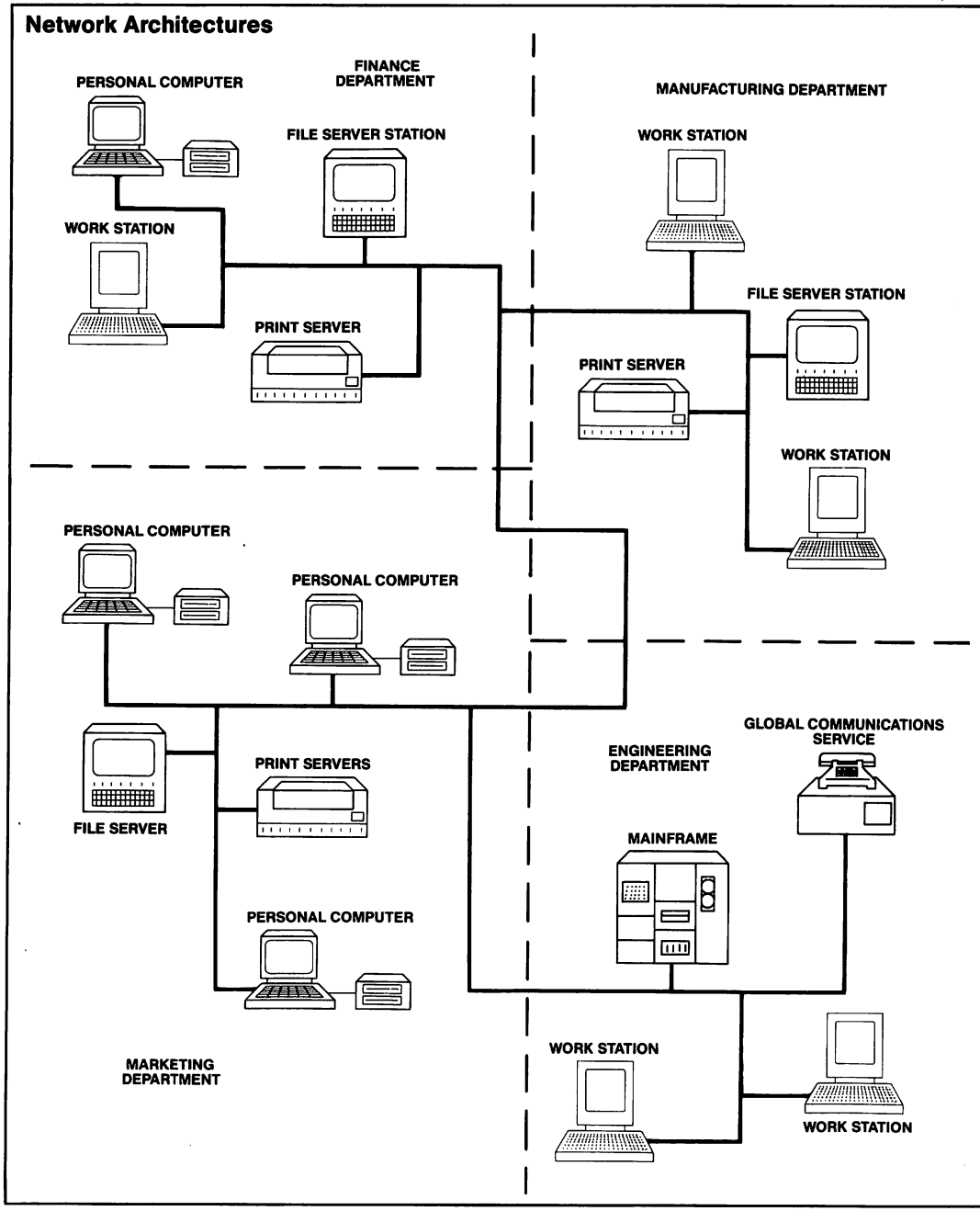


Figure 2.1 Local Area Network Connection

Before discussing the specifics of a local area network and its implementation, it is useful to first review the emerging "open systems" interconnection model for communications networks. Local area networks can then be described within the communications model and, finally, the solutions that VLSI and software offer.

As with a simple telephone conversation, communications between two stations (a word processor and the disk file in Figure 2.1) over a connecting wire involves the orderly transmitting and receiving of messages (or sentences in a conversation). In order to proceed with the exchange of messages, both parties must first agree to a set of rules for conducting the message exchange (or conversation).

The International Standard Organization (ISO), in an effort to encourage "open" networks, whereby equipment (and/or networks) from multiple vendors can be interconnected, developed the open systems interconnection (OSI) reference model to form the basis for all network development. In simple terms, it logically groups the functions and sets of rules necessary to establish and conduct communications between two or more parties into seven layers. Ethernet, for example, implements the first two layers, the physical and data link layers, of the reference model. The layers of the OSI model are summariazed in Figure 2.2.

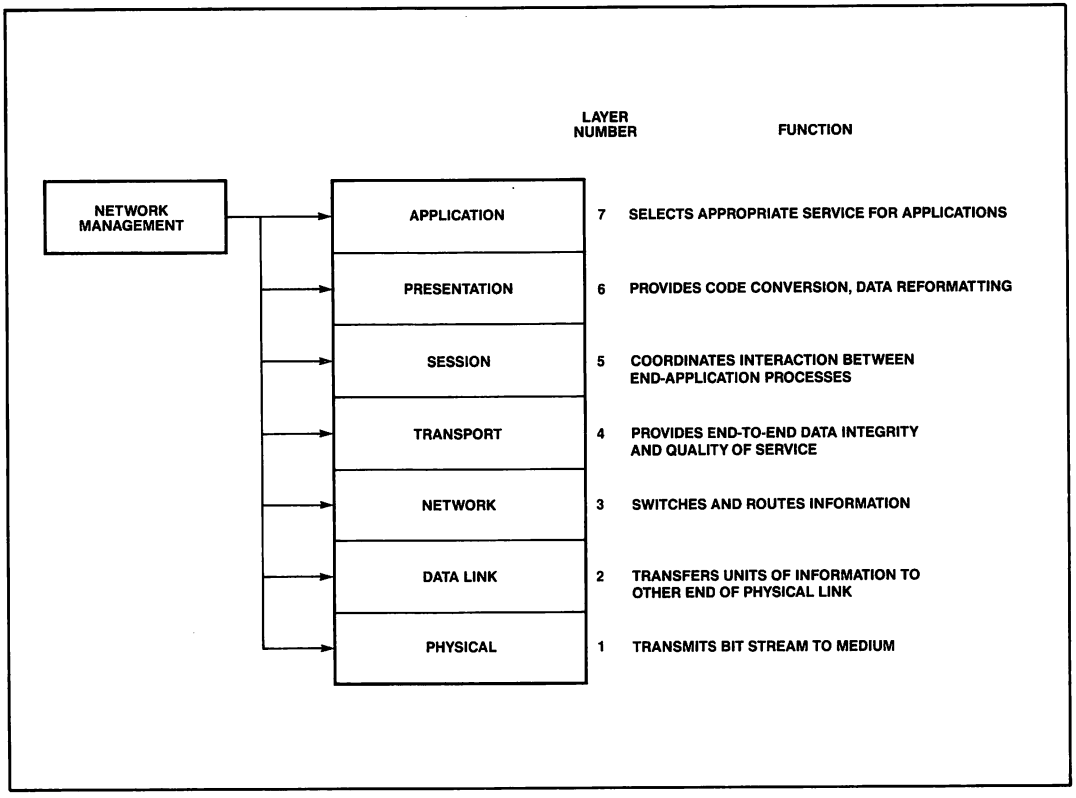


Figure 2.2 ISO Open Systems Model

PHYSICAL LAYER
The physical layer describes the physical media over which the bit stream is to be transmitted. Such specifics as type of cable (coax, twisted pair, etc.) signal levels, bit rate, etc. are described in the physical layer. In short, it describes the actual physical media over which the bit stream is transmitted and the method of transmission i.e., baseband or broadband.

DATA LINK LAYER
The data link layer describes the rules for transmitting on the physical media. Such items as the format of the information (or frame) and procedures for obtaining control of the physical media (referred to as the media access methods), transmitting the frame, and releasing the physical media are described in the data link layer.

NETWORK LAYER
The network layer governs the switching and routing of information between networks. Because all station-to-station communication within a single local area network is point-to-point, the Network layer is not necessary for a single LAN system.

TRANSPORT LAYER
The transport layer assures end-to-end integrity and provides for the required quality of service for exchanged information. End-to-end acknowledgements of successful message receptions are performed by the transport service, for example.

SESSION LAYER
The session layer manages the requesting and deleting of virtual circuit connection services provided by the Transport layer. The layer is also responsible for the mapping of logical names to network addresses.

PRESENTATION LAYER
The presentation layer provides for any necessary translation, format conversion, or code conversion to put the information into a recognizable form.

APPLICATION LAYER
The application layer directly serves the communicating end-user application process by providing the distributed information service appropriate to the application and its management. Network services such as file server, electronic mail or virtual terminal protocols are provided by the Application layer.

NETWORK MANAGEMENT
Network management is responsible for operation planning, which includes the gathering of operational statistics such as errors and traffic. It is also responsible for network initialization and maintenance (fault detection and isolation). Rather than being defined as a layer in the open systems model, network management interfaces with each layer in order to perform its tasks.

The physical and data link layers of the OSI model assure *interconnectability*. By implementing a particular physical and data link specification, equipment from multiple vendors can be physically and electrically connected i.e., *interconnection*. The remaining five layers of the OSI model assure *interoperation* among the interconnected stations in an open network.

For example, Intel's NDS-II Multi-User Networked Microcomputer Development System is a LAN based system utilizing Ethernet for layers 1 and 2 and Intel Network Architecture (iNA) for layers 3 through 7. Non-NDS-II equipment wishing to connect to the physical network need only adhere to the Ethernet specifications to assure proper *interconnection* and gain access to the "data highway". In order to communicate with the system's Network Resource Manager (for interoperation), the foreign station would have to conform to the remaining layers of iNA.

As a first step towards "open networks", the Ethernet and proposed IEEE 802 specifications describe the first two layers of the OSI model, the physical and data link layers. As readily available public specifications, they completely describe the type of cable, speeds, frame format, media access methods, etc. to allow equipment from multiple vendors to be physically interconnected. As can be seen from Figure 2.3, Ethernet covers the first two of the seven layer OSI model.

The 82586 Local Communications Controller, along with the 82501 Ethernet Serial Interface,

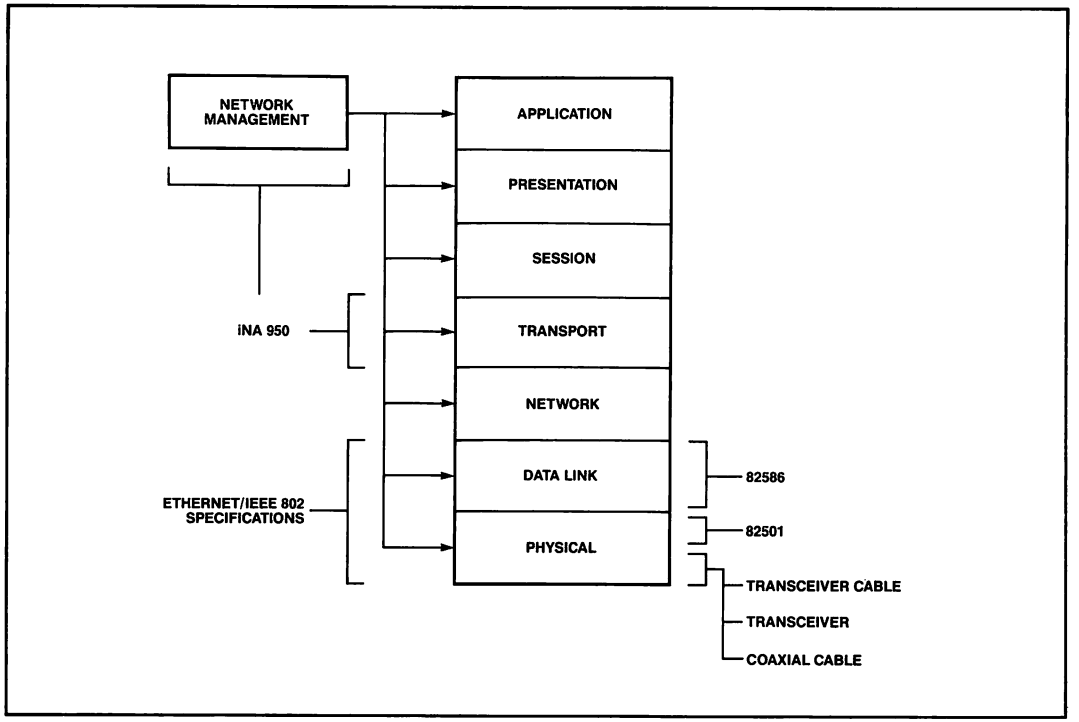


Figure 2.3 The OSI/Intel Implementation

and readily available transceiver cable, transceivers and coaxial cable provide the complete implementation for the Ethernet specification which covers the first two layers of the OSI model. iNA 950, which represents the transport and network management functions within Intel Network Architecture (iNA), provides the Transport layer and Network Management function of the OSI model.

Within a single local area network, all communications between any two stations is always point-to-point because of their direct connection. The network layer, which is responsible for internetwork message routing, is not needed for intranetwork communications. Thus, for terminals that are physically connected (*interconnection*) to communicate and operate with one another (*intercommunication* and *interoperation*), the remaining three layers, Session, Presentation and Application, must be implemented, typically in software.

NETWORK ELEMENTS

Local area networks are communications networks extending from several hundred to several thousand feet within a building or other facility. As seen from Figure 2.1, LANs are a means of connecting various types of equipment for the purposes of sharing resources and communicating in a distributed processing environment. Although LANs using a range of speeds (from 2400 bits/sec up to 2 Mbits/sec) exist today, the trend is clearly towards networks between 1 and 10 million bits per second.

The Ethernet, which has gained wide acceptance by both large and small corporations, is such a high speed (10 Mbps) LAN. In Figure 2.4, the main components of the Ethernet network are as follows:

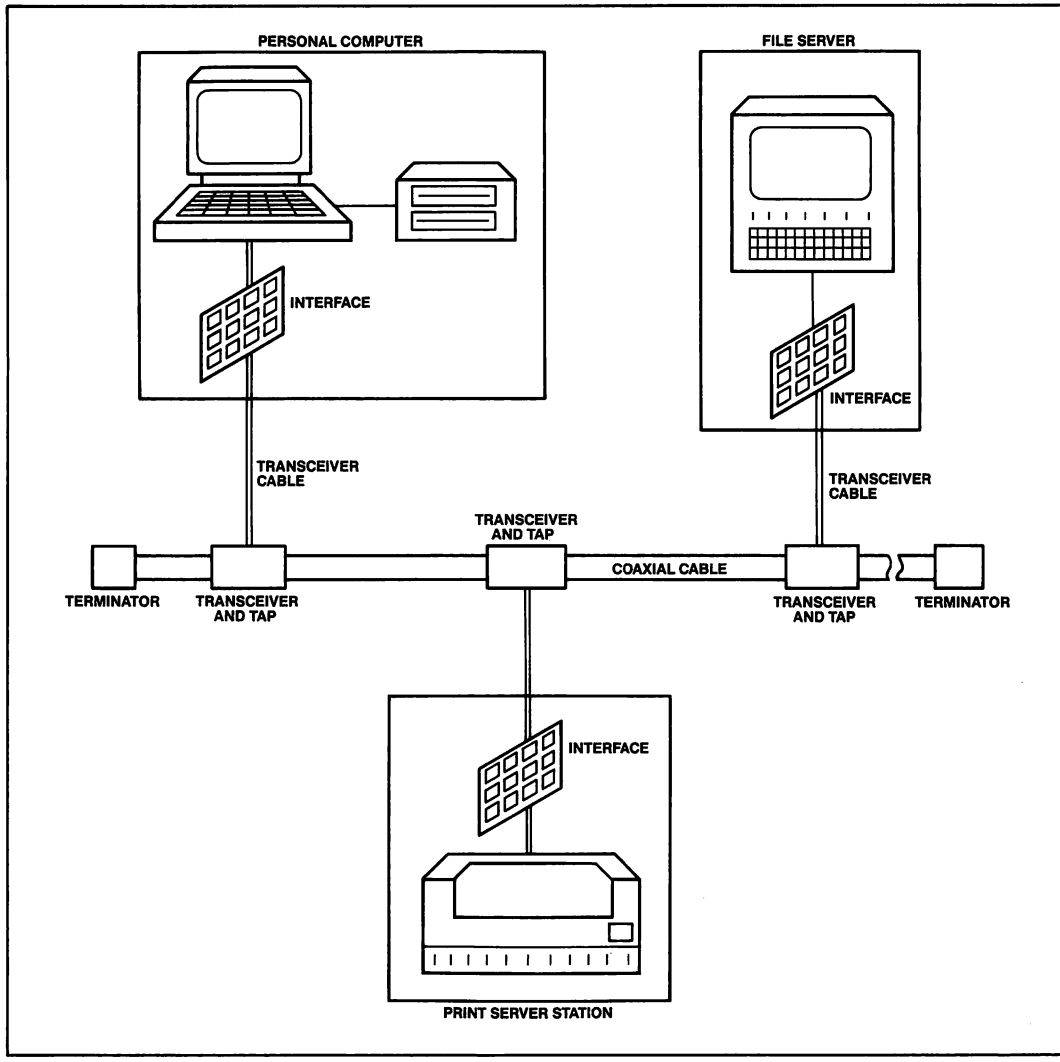


Figure 2.4 Network Elements

The *cable* is a low noise, shielded 50 ohm coaxial cable. Over this cable, information is transmitted at the rate of 10 million bits per second. Segments of the cable can be up to 500 meters (546.8 yards) in length and can be connected into longer network lengths using *repeaters*. A repeater provides the signal regeneration that is required to strengthen the data transmission signal along the extended length of the cable.

The *transceiver*, a small electronic device, transmits and receives signals on the coaxial cable, and generally protects against failure and detects electrical interference (referred to as collisions) on the cable. It is connected to the cable using a simple tap and to the interface by means of a *transceiver cable* which consists of four individual twisted pairs and may be up to 50 meters (54.68 yards) in length.

The *terminator* is a passive device which fits on both ends of each cable segment (or the ends of segments connected by repeaters), providing proper electrical termination.

Finally, the *interface* is the real work horse of the network. It provides the connection to the user or the server station and performs such functions as:

- Data encapsulation/decapsulation (frame assembly/disassembly)
  - —handling of source and destination addressing
  - —detection of physical channel transmission errors
  - —frame delimiting
- Network link management
  - —collision avoidance
  - —collision handling
- Encoding and decoding of the signal to and from the transceiver

The key element of the Ethernet and IEEE 802 specifications is the media access method (or rules for using the shared coaxial cable). Commonly referred to as *C*arrier *S*ense *M*ultiple *A*ccess with *C*ollision *D*etection (CSMA/CD), it is a simple and efficient means of determining how a station transmits information over common medium that is shared with other stations. In order to transmit information, a station takes the following steps:

CARRIER SENSE
Any station wishing to transmit "listens" first. If the cable is busy (i.e., some other station is transmitting) the station waits until the line is clear before transmitting.

MULTIPLE ACCESS
Any station wishing to transmit can do so. No central controller is needed to decide who is able to transmit and in what order. This is commonly referred to as distributed control, where all stations on the network are peers with equal access.

COLLISION DETECTION
When the cable is free (no other station is transmitting), a station can start transmitting. The transmitting station (or stations) always listens while transmitting in order to detect any other station transmitting on top of its own signal, causing a "collision". In the event of such collisions, where two or more stations are transmitting at the same time, the transmitting stations will continue transmitting for a fixed time to insure that all transmitting stations detect the collision. This is known as the "jam". After the jam, the stations stop transmitting and wait a random period of time before retrying. The range of random wait times increases (by the power of 2) with the number of successive collisions such that collisions can be resolved even if a large number of terminals are colliding (this is referred to as the exponential backoff algorithm).

The three most significant charcteristics of CSMA/CD based networks are the "passive" nature of the network, its reliability and expandability. The CSMA/CD media access method enables the network to operate without central control or switching logic. If a station on the network malfunctions, it does not affect the ability of other stations to communicate with each other, nor impact the operation of the network.

A direct result of such a passive network is increased reliability. Total network failure cannot be caused by a single station malfunctioning. A system wide failure can be caused by a cable malfunction such as an open or short circuit or a continuously transmitting station, and system hardware has built-in checks to detect and correct such situations.

The passive, distributed nature of a CSMA/CD based network also permits easy expansion. Stations can be added to (or deleted from) an existing network without reinitialization or reconfiguration of all other stations. Such a capability supports future growth requirements through simple expansion of the network.

In an Ethernet network, both the 82586 and the 82501 provide the interface functions as seen in Figure 2.5. Together, they provide the data link and part of the physical layer functions of the Ethernet specification. The balance of the

physical link is provided by readily available transceiver cables, transceivers and coaxial cables. And, through the OSI layered architecture model, other physical layer implementations are possible. The programmability of the 82586 will also allow designers to apply the CSMA/CD access method to other LAN alternatives including lower speed applications, serial backplanes and local peripheral connections.

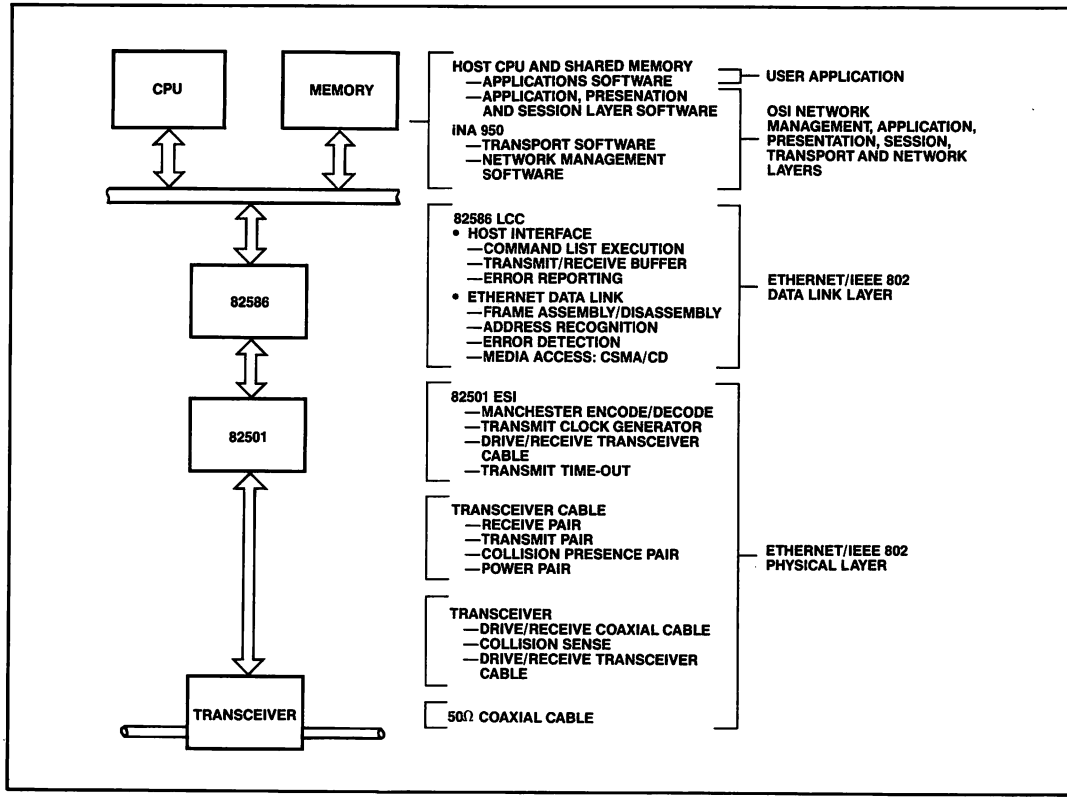


Figure 2.5 82586/82501 Implementation

3.0 EMERGING APPLICATIONS —SERIAL BACKPLANES

Perhaps the largest application of local area networks is not the connection of stations in a building but the connection of the functional sub-modules within each station. By using a high speed "serial backplane", a single cable can replace multiple cable bundles, multi-pin connectors and arrays of drivers and receivers now used to connect sub-modules within a cabinet. And, by extending the serial backplane concept outside the cabinet, a single serial cable can also connect a station's peripherals such as local printers and data storage units.

An example of a system with multiple sub-modules (or sub-assemblies) is a high speed document copier. Typical sub-modules would include a display and control panel, camera and developer and sorter. Using a single CPU to control the copier, separate cable bundles containing data and unique control signals would be required for each sub-module as shown in figure 3.1a.

As seen from the example, the major contributors to system cost are the cable bundles and associated mechanical and electrical assemblies needed to connect each sub-module. Such connection techniques also impose rigid design constraints because of the cost of running unique cable bundles for each sub-module. Standard options such as a high speed document feeder or collator for the copier require costly cable bundles that are unique for each option. And, multiple cable bundles and their associated connectors and drivers/receivers introduce additional failure potentials which reduce system reliability while compounding service and maintenance costs.

The rapid decline in microprocessor and memory costs now makes it possible for the designer to dedicate a single inexpensive microprocessor to each of the sub-modules in a system. With the added intelligence in each sub-module to perform control and communications functions, the bulky cable bundles carrying individual control and data signals can be replaced by a single, shared serial backplane. Thus in the high speed copier example, individual microprocessors are used to control each sub-module and manage the serial I/O interface, Figure 3.1b. The serial backplane carries both control and data signals for each sub-module as required to coordinate activities between the display and control panel and the sorter and camera assemblies. Copier options such as a high speed document feeder would connect to the same serial backplane and not require their own unique cable assemblies.

In short, replacing multiple cable bundles, connectors and drivers and receivers with a single shared serial backplane will result in a lower cost, more reliable and flexible (i.e., modular growth) system.

Up to now the missing "link" limiting the wide spread application of serial backplanes has been the lack of a cost-effective VLSI solution. The 82586 LCC fills that missing link. Intel has designed the 82586 to be flexible enough to apply to both the Ethernet/IEEE 802 open LAN interfaces, serial backplanes, as well as other emerging CSMA/CD applications. The cumulative volumes and resulting economy of scale will make the 82586 Local Communications Controller the most cost-effective VLSI solution for LANs.

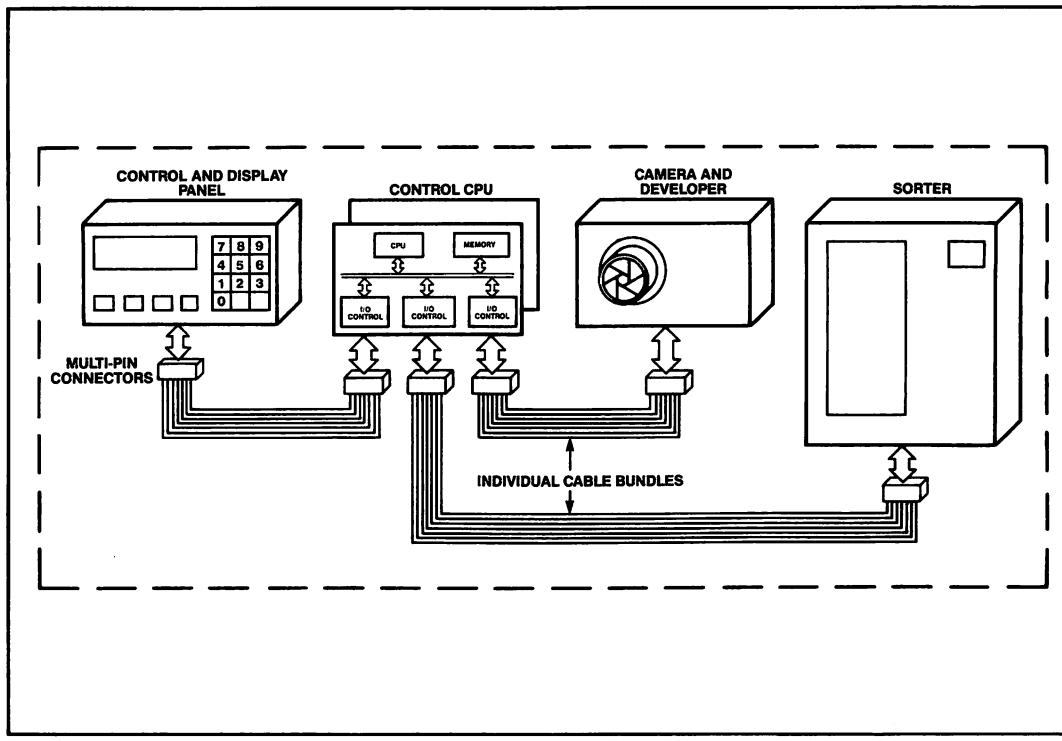


Figure 3.1a Multiple Cable Connections

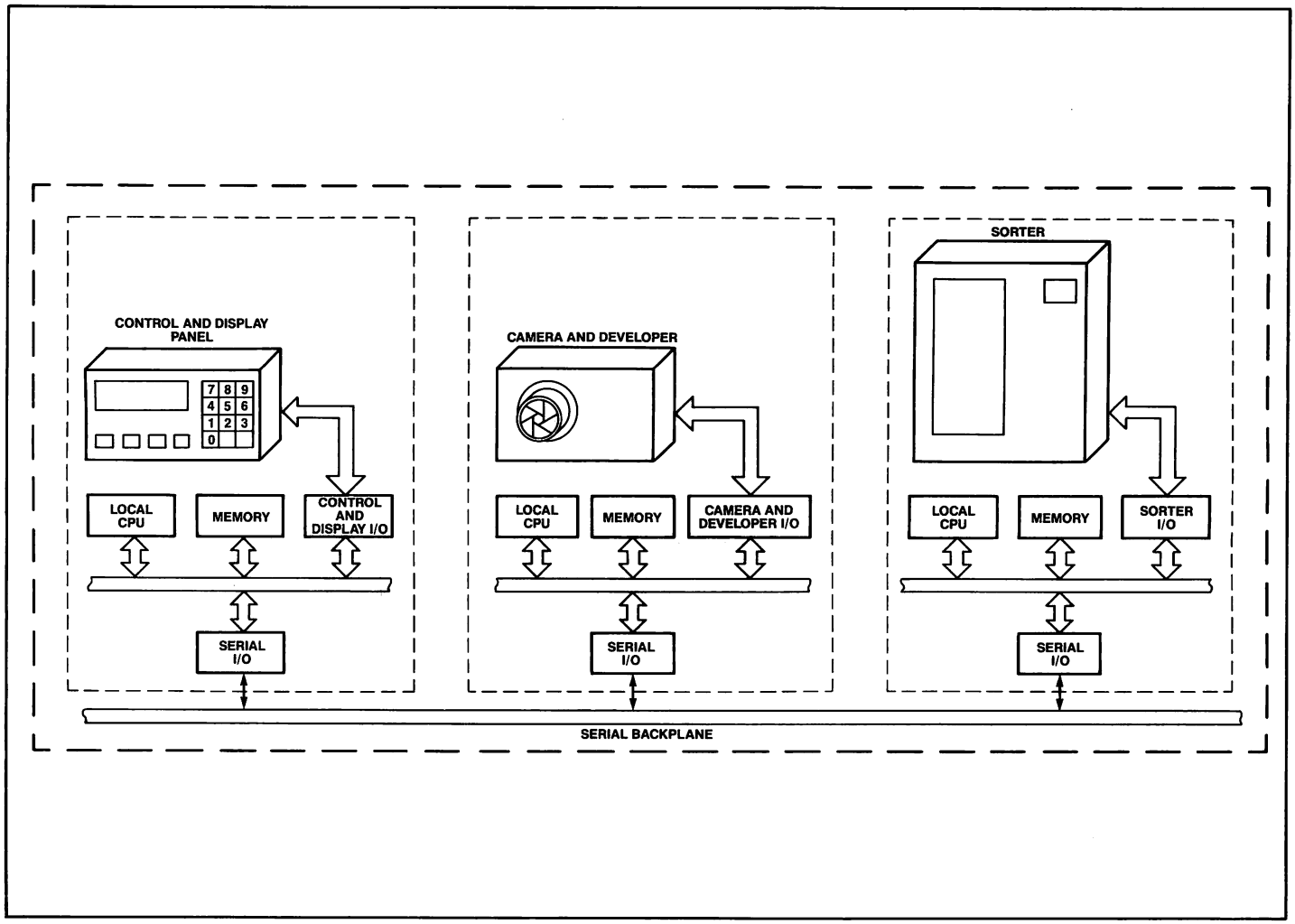


Figure 3.1b Serial Backplane Connection

82586
LOCAL COMMUNICATIONS CONTROLLER

- **Fully Implements the Ethernet* and Proposed IEEE 802 Specifications**
- **User Configurable for Non-Ethernet Applications**
  - **—From 0 to 6 Bytes of Address Generation/Checking**
  - **—16- or 32-Bit CRC Generation/Checking**
  - **—Optional Priority Function**
  - **—Variable Preamble Length**
  - **—Serial Transmission from 100 Kbps to 10 Mbps**
  - **—8- or 16-Bit Data Bus**
- **4 On-Chip DMA Channels for Efficient, High-Speed Transfer of Data, Status and Commands**
- **Complete Set of Diagnostics for Reliable Network Operation**
- **Fully Implements the CSMA/CD Access Method Including Retries and Random Backoff (Wait) Time**
- **Two Methods of Frame Delimiting: Ethernet and HDLC Flags/Bit Stuffing**
- **Independent 8-MHz System Clock Input**

Designed as an intelligent peripheral, the 82586 manages the entire process of transmitting and receiving frames, thereby relieving the host processor of the tasks of managing the communications peripheral. The major functions performed by the 82586 include:

—direct transfer of frames to and from external memory using four on-chip DMA channels.
—executes commands from lists residing in external shared memory.
—automatically reports the status of both transmitted and received frames, including error conditions.
—fully integrates the CSMA/CD access method including automatic retries after collisions and random backoff (wait-time) generation.
—diagnostic commands to identify and isolate faults.

In order to take full advantage of the LAN concept and CSMA/CD access method, the 82586 architecture is also configurable under program control. This allows the 82586 to be "customized" for other applications requiring high-speed serial transmission including serial backplanes (serial peripheral interconnection) and low-cost, short-distance LANs.

*Ethernet is a trademark of Xerox Corporation.

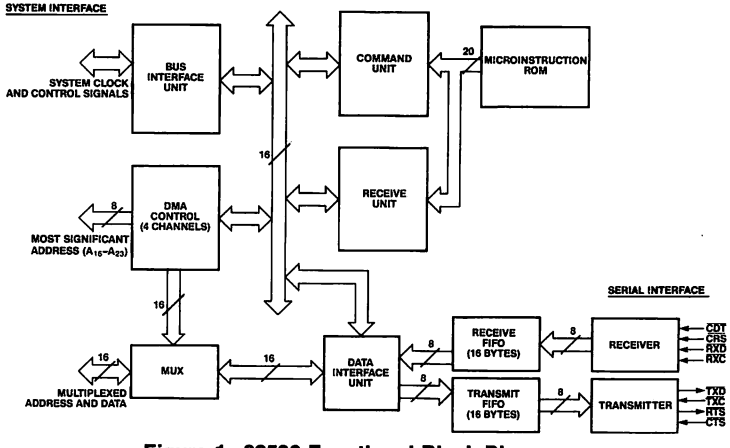
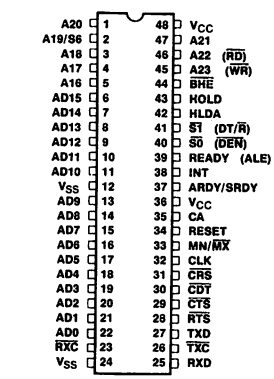


Figure 1. 82586 Functional Block Diagram



NOTE: THE SYMBOLS IN PARENTHESES CORRESPOND TO MINIMUM MODE.

Figure 2. 82586 Pinout

OCTOBER 1982
ORDER NUMBER: 210783-001

Controlling the 82586

From the user's point of view, the 82586 consists of two independent, though communicating units: the Command Unit (CU) and the Receive Unit (RU) as shown in Figure 3. The CU executes commands given by the host CPU and manages frame transmissions. The RU handles all activities related to frame reception such as buffer management, frame and address recognition, and CRC checking. The two units are controlled and monitored by the host CPU via a shared memory structure called the System Control Block (SCB). All logical communication between the CPU and 82586 takes place through the SCB. The two other memory structures used by the 82586 are the Command Block List (CBL) and Receive Frame Area (RFA). These are used to hold the list of commands to be executed by the 82586 and hold all received frames, respectively. Pointers to the CBL and RFA are in the SCB, along with status registers and counters for certain tallies maintained by the 82586, and control commands for the 82586. The only direct control lines between the CPU and the 82586 are the interrupt to the CPU and Channel Attention to the 82586.

82586 Memory Structures

The three primary memory structures used by the host CPU and the 82586 to pass control, status and data are the System Control Block (SCB), Command Block List (CBL) and the Receive Frame Area (RFA), Figure 4.

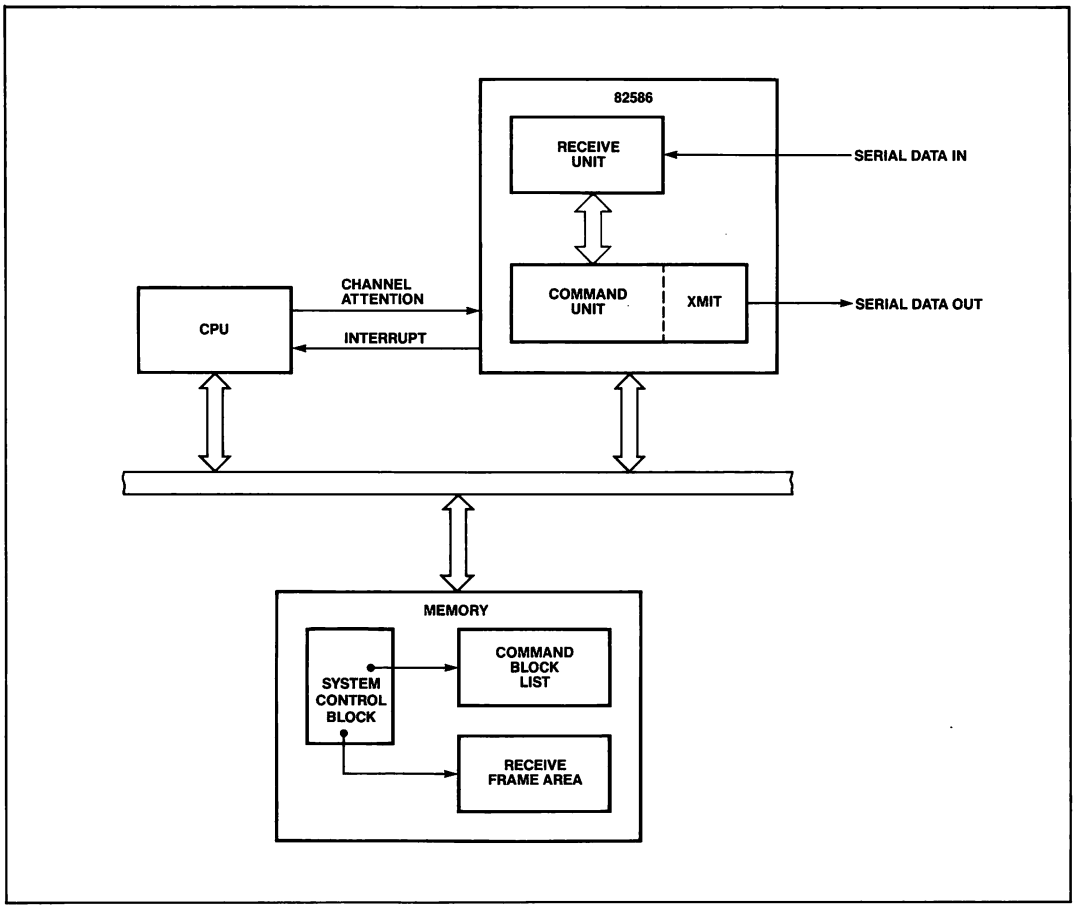


Figure 3. System Overview

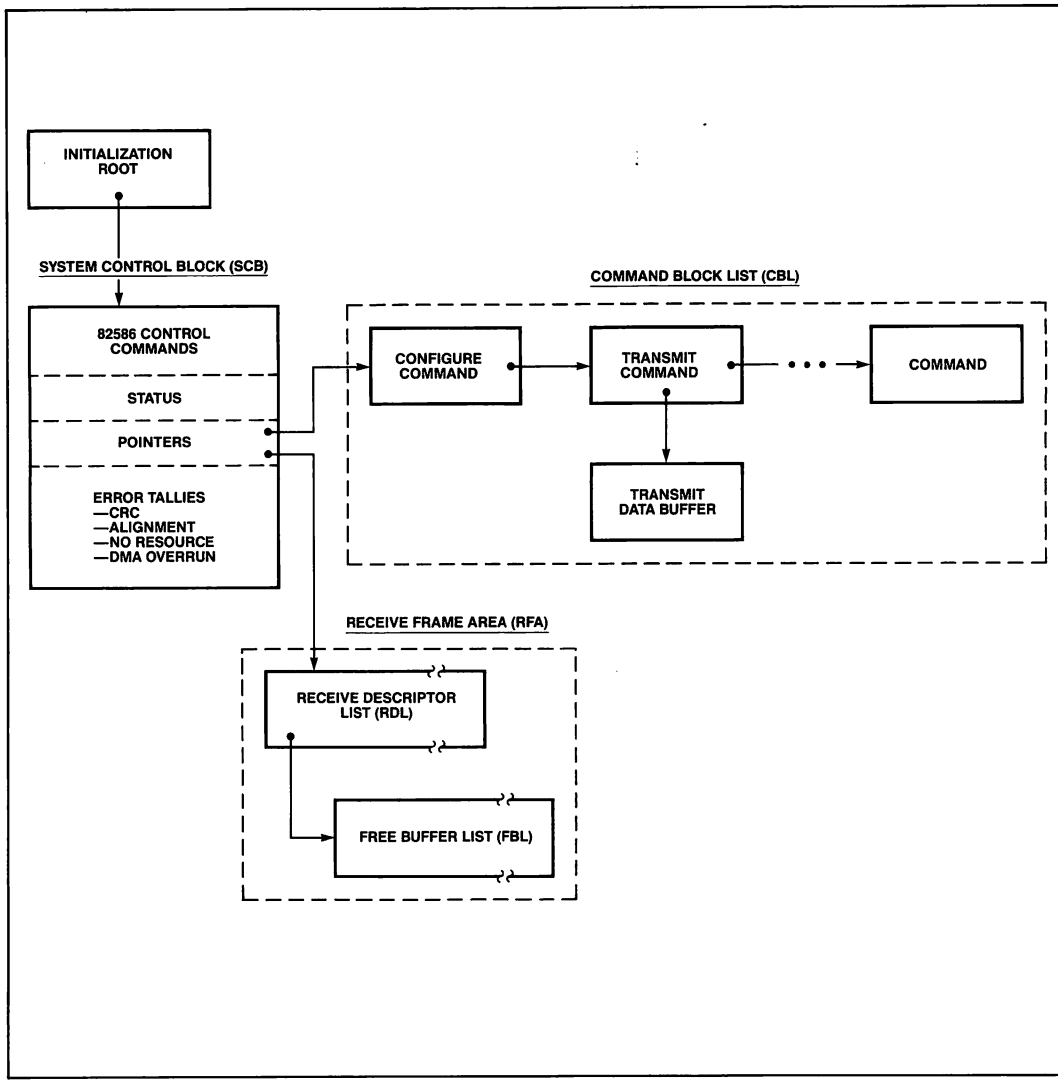


Figure 4. 82586 Memory Structures

Upon initialization, the 82586 obtains the address of its System Control Block through the Initialization Root which begins at location 0FFFFF6H. The SCB contains control commands, status register, pointers to the Command Block List (CBL) and Receive Frame Area (RFA), and tallies for CRC, Alignment, DMA Overrun and No Resource errors. Through the SCB, the 82586 is able to provide status and error counts for the host CPU, execute "programs" contained in the Command Block List (CBL) and receive incoming frames in the Receive Frame Area (RFA).

Both the Command Block List and the Receive Frame Area are first configured by the host CPU and then passed to the 82586 via the SCB. As commands are executed by the 82586, it returns status information back to the CPU, which may, in turn, update the CBL with additional commands. As frames are received by the 82586 and stored in the RFA, the 82586 will return the appropriate status to the CPU. The CPU retrieves the received frames from the RFA and returns free buffers to the Receive Descriptor and Free Buffer lists.

The 82586 has a 22-bit memory address range in minimum mode and 24-bit memory address range in maximum mode. All memory structures, the System Control Block, commands in the Command Block List, Receive Descriptor List, and all buffer descriptors (see Figure 6), must reside within one 64K-byte memory segment. The Data Buffers can be located anywhere in the memory space.

Transmitting Frames

The 82586 executes commands from the Command Block List in external memory. These commands are fetched and executed in parallel with the host CPU's operation, thereby significantly improving system performance potential. The general format for such commands is

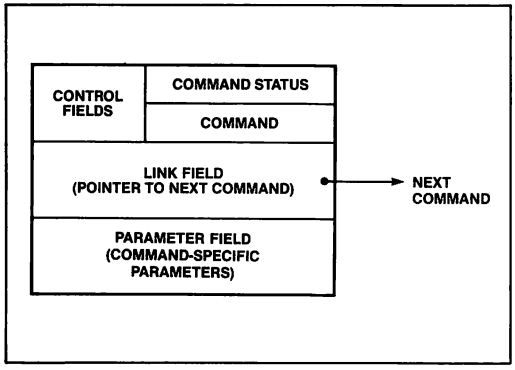


Figure 5. Action Command Format

Via the Link Field, commands can be linked together to form a list of commands for execution by the 82586.

One such command is the Transmit command. A single Transmit command contains, as part of the command-specific parameters, the destination address and type field for the transmitted frame along with a pointer to a buffer area in memory containing the data portion of the frame. The data field is contained in a memory data structure consisting of a Buffer Descriptor (BD) and Data Buffer (or a linked list of buffer descriptors and buffers) as seen in Figure 6. The BD contains a Link Field which points to the next BD on the list and a 24-bit address pointing to the Data Buffer itself. The length of the Data Buffer is specified by the Actual Count field of the BD.

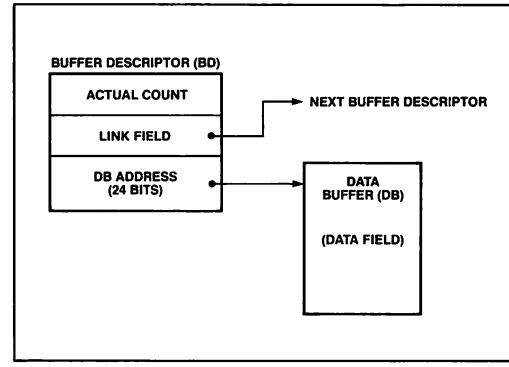


Figure 6. Transmit Data Buffer

Using the BDs and Data Buffers, multiple Data Buffers can be "chained" together. Thus, a frame with a long Data Field can be transmitted using multiple (shorter) Data Buffers chained together. This chaining technique allows the system designer to develop efficient buffer management policies.

When transmitting a frame as shown below:

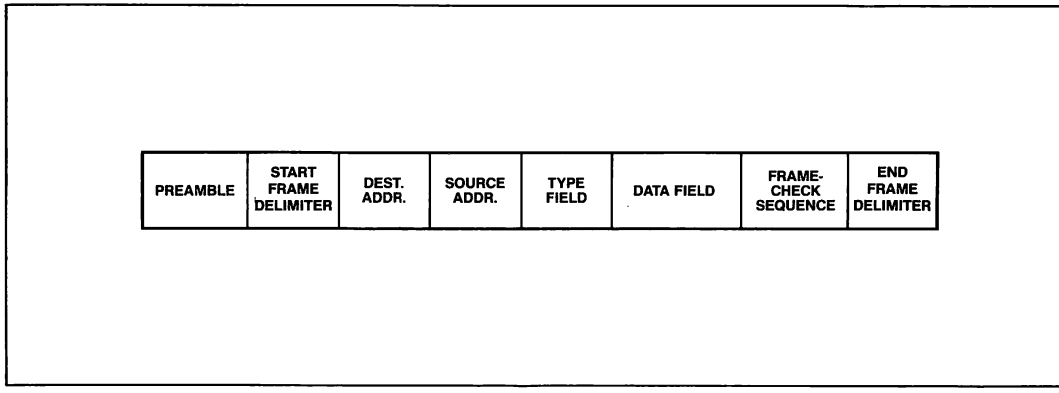


Figure 7. Frame Format

The 82586 automatically generates the preamble (alternating 1s and 0s) and start frame delimiter, fetches the destination address and type field from the Transmit command, inserts its unique address as the source address, fetches the data field from buffers pointed to by the Transmit command, and computes and appends the CRC at the end of the frame.

The 82586 can be configured to generate either the Ethernet or HDLC start and end frame delimiters. In the Ethernet mode, the start frame delimiter is two consecutive 1 bits and the end frame delimiter indicated by the lack of a signal after transmitting the last bit of the frame-check sequence field. When in the HDLC mode, the 82586 will generate the 01111110 "flag" for the start and end frame delimiters and perform the standard "bit stuffing/stripping." In addition, the 82586 will optionally pad frames that are shorter than the specified minimum frame length by appending the appropriate number of flags to the end of the frame.

In the event of a collision (or collisions), the 82586 manages the entire jam, random wait and retry process, reinitializing DMA pointers without CPU intervention. Multiple frames can be sent by linking the appropriate number of Transmit commands together. This is particularly useful when transmitting a message that is larger than the maximum frame size (1518 bytes for Ethernet).

Receiving Frames

In order to minimize CPU overhead, the 82586 is designed to receive frames without CPU supervision. The host CPU first sets aside an adequate amount of receive buffer space and then enables the 82586's Receive Unit. Once enabled, the 82586 "watches" for any of its frames which it automatically stores in the Receive Frame Area (RFA). The RFA consists of a Receive Descriptor List (RDL) and a list of free buffers called the Free Buffer List (FBL) as shown in Figure 8. The individual Receive Frame Descriptors that make up the RDL are used by the 82586 to store the destination and source address, type field and status of each frame that is received. (Figure 9.)

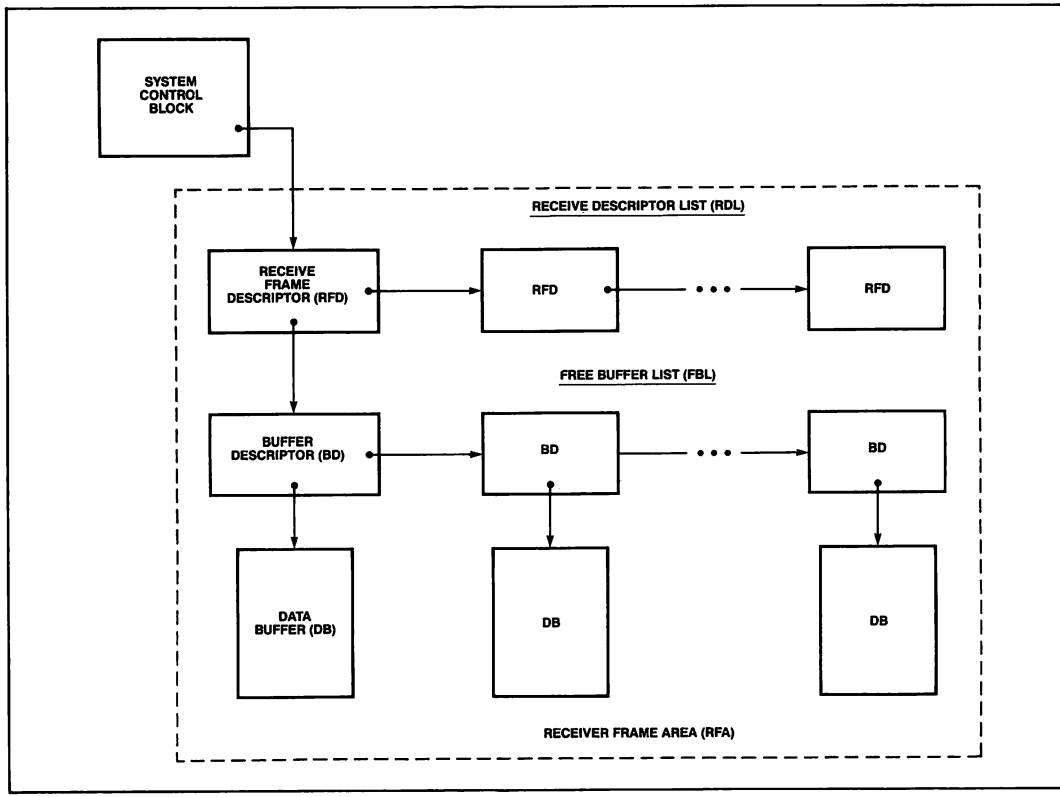


Figure 8. Receive Frame Area Diagram

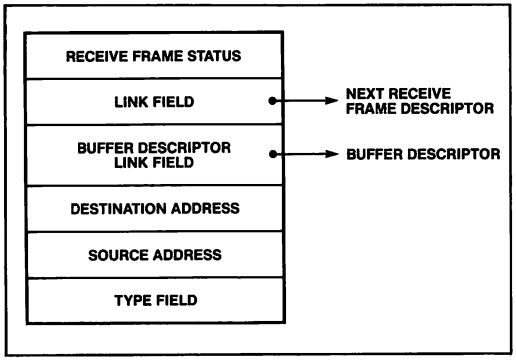


Figure 9. Receive Frame Descriptor

The 82586, once enabled, checks each passing frame for an address match. The 82586 will recognize its own unique address, one or more multicast addresses or the broadcast address.

If a match occurs, it stores the destination and source address and type field in the next available RFD. It then begins filling the next free Data Buffer on the FBL (which is pointed to by the current RFD) with the data portion of the incoming frame. As one DB is filled, the 82586 automatically fetches the next DB on the FBL until the entire frame is received. This buffer chaining technique is particularly memory efficient because it allows the system designer to set aside buffers that fit a frame size that may be much shorter than the maximum allowable frame. Without buffer chaining, all receive data buffers would have to be as long as the maximum allowable frame.

Once the entire frame is received without error, the 82586 performs the following housekeeping tasks:

—Updates the Actual Count field of the last Buffer Descriptor used to hold the frame just received with the number of bytes stored in its associated Data Buffer.
—Fetches the address of the next free Receive Frame Descriptor.
—Writes the address of the next free Buffer Descriptor into the next free Receive Frame Descriptor.
—Posts a "Frame Received" interrupt status bit in the SCB.
—Interrupts the CPU.

In the event of a frame error, such as a CRC error, the 82586 automatically reinitializes its DMA pointers and reclaims any data buffers containing the bad frame. As long as Receive Frame Descriptors and data buffers are available, the 82856 will continue to receive frames without further CPU help.

82586 Action Commands

The 82586 executes a "program" that is made up of action commands in the Command Block List. As shown in Figure 5, each command contains the command field, status and control fields, link to the next action command in the CBL, and any command-specific parameters. The 82586 has a repertoire of 8 commands.

NOP
Set Up Individual Address
Configure
Set Up Multicast Address
Transmit
TDR
Diagnose
Dump

NOP:
This command results in no action by the 82586 other than the normal command processing such as fetching the command and decoding the command field.

Individual Address Set Up:
This command is used to load the 82586's unique address. The unique address is contained in the parameter field of the command.

Configure:
The Configure command is used to load the 82586 with its operating parameters. Upon reset, the 82586 initializes to the Ethernet-based parameters. If the user wishes to use any other values, the Configure command is used.

Multicast Address Set Up:
This command allows the programmer to set up one or more multicast addresses into the 82586. The multicast addresses to be set up are located in the parameter field of the command.

Transmit:
One Transmit command is used to send a single frame. If more than one frame is to be sent, the host CPU can link multiple Transmit commands together. The destination address, type field and pointer to buffers containing the data field are contained in the parameter field of the Transmit command.

TDR:
This command performs the Time Domain Reflectometry test on the coaxial cable. The TDR command is used to detect and locate cable faults caused by either short or open circuits on the coaxial cable.

Diagnose:
The Diagnose command puts the 82586 through a self-test procedure and reports on the success or failure of the internal test.

Dump:
This command causes the 82586 to dump its internal registers into memory. The registers included are those loaded by the Configure and Address Set-Up commands, plus status and other internal working registers.

PROGRAMMABLE NETWORK PARAMETERS AND DIAGNOSTICS

Network Parameters

The Ethernet specification represents a complete description of the physical and data link layers of a local-area network. As such, items such as address and preamble length, maximum distance between two stations, and frame-check sequence length are fixed to assure that stations connecting to the network are compatible. Through the Configure command, the 82586 can also be tailored to achieve maximum efficiency in other network configurations. The following parameters can be specified in the Configure command:

PARAMETER	LENGTH
Source/Destination Address	0 to 6 bytes
CRC	16 or 32 bits
Preamble Length	16, 32, 64, or 128 bits
Frame Delimiter	Ethernet or HDLC (Flags and bit stuffing)
Slot Time	11 bits to specify number of transmit clock times

The Slot time is a period slightly longer than the maximum round-trip delay time through the network, i.e., the round-trip delay between the two most distant stations. The Slot time is used in the CSMA/CD Backoff calculation where the random time is defined in increments of the Slot Time. Shorter networks, such as a serial backplane within a cabinet would have a very short Slot Time compared to a 2500-meter Ethernet Network, for example.

Priority can also be assigned via a field in the Configure command. This field specifies the amount of time the particular 82586 must wait after the cable has been quiet before attempting to transmit its frame. By assigning lower-priority stations a longer wait time, the high-priority (shorter wait-time) stations will have better access to the cable during peak busy periods.

Diagnostics

In addition to specifying network parameters, the Configure command is also used to call up a powerful set of diagnostic functions through individual fields within the command.

Save Bad Frame:
Under normal operation, the 82586 automatically discards frames with errors, such as a CRC error. Frames can be saved for later examination by requesting it through this field.

Address/Type Field Location:
This field informs the 82586 that the destination and source addresses and type field are the first entries in the Transmit Data Buffer rather than in the parameter field of the Transmit command (destination address and type field) and Individual Address register of the 82586 (source address).

Loopback:
Two Loopback modes are available on the 82586. The Internal Loopback moves the transmitted frame from memory into the 82586 FIFO, through the bit transmitter, back into the bit receiver and back to memory without going through the external serial drivers and receivers of the 82586. Note that the data moves at one-fourth the normal bit rate when the 82586 is in Internal Loopback.

External Loopback is identical to the Internal Loopback except that the frame does move out through the serial drivers and back in through receivers of the 82586 at the normal bit rate. This allows external components, such as the 82501 ESI chip and Ethernet transceivers, to be tested independently of the coaxial cable or remote station.

Promiscuous Receive:
The 82586 can be made to receive all good frames, regardless of address, using this field. This is useful as a monitor or diagnostic mode for a station.

Broadcast Disable:
This field is used to disable the reception of all broadcast messages by the 82586.

Minimum Frame Length:
The 82586 automatically rejects received frames which are shorter than the minimum frame length as specified in this field. This 9-bit field allows the minimum frame length to range from 1 to 511 bytes. (For Ethernet the minimum frame is 64 bytes.)

No CRC Insertion:
This field disables the automatic CRC insertion and terminates the frame after last byte of the data field is transmitted. Using this option, frames with the wrong CRC can be generated in order to test a receiving station's CRC checking circuitry.

As an aid to monitoring the operation of the network and tracking its "vital signs," the 82586 also reports the following conditions after each received and transmitted frame.

RECEIVED FRAME
—No errors
—Short Frame (less than minimum frame length)
—DMA Overrun; FIFO overflow before DMA service
—CRC error
—Alignment error
—No resources (buffers) to store frame

TRANSMITTED FRAME
—Frame transmitted
—Number of collisions encountered
—Transmission aborted, too many collisions
—DMA underrun; FIFO empty before DMA service
—Channel busy; 82586 deferred before frame transmission
—CTS (Clear To Send) lost
—CRS (Carrier Sense) lost
—Collision Test Status (Heartbeat Test)

System Interface

The 82586 operates as a bus master. Through its HOLD/HLDA signals, it is able to request bus cycles, transfer data and release the bus. Two internal 16-byte FIFOs are used to buffer data to and from the system bus through the four DMA channels on the 82586. Therefore, once the DMA request is granted, the 82586 is able to transfer multiple bytes of data (to fill or empty the FIFO) with each DMA request.

The 82586 system interface is a standard multiplexed bus that can be used with any of the popular 8- and 16-bit microprocessors. It is optimized for minimum-interface support logic when used with the iAPX 186 (Figure 10). When combined with the 82501 ESI chip, the Ethernet interface is complete from the CPU to the transceiver cable.

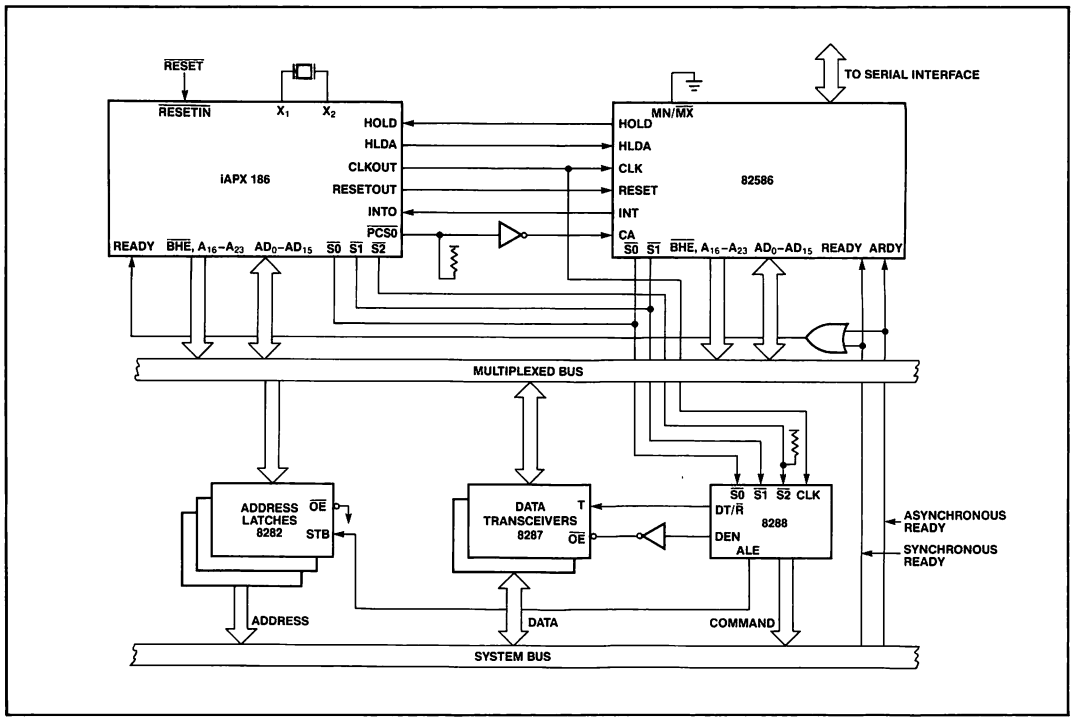


Figure 10. iAPX 186/82586 System

AFN-00864A

For 8086/8088-based systems, the 82285 is used as a clock generator for the 82586 system clock and the 8259A as an interrupt controller. The 8288 Bus Controller is common as both the CPU and the 82586 have the same data transfer timing. A bus arbiter is also needed to convert to/from the 82586 HOLD/HLDA from/to the 8086/8088 RQ/GT. This configuration is shown in Figure 11.

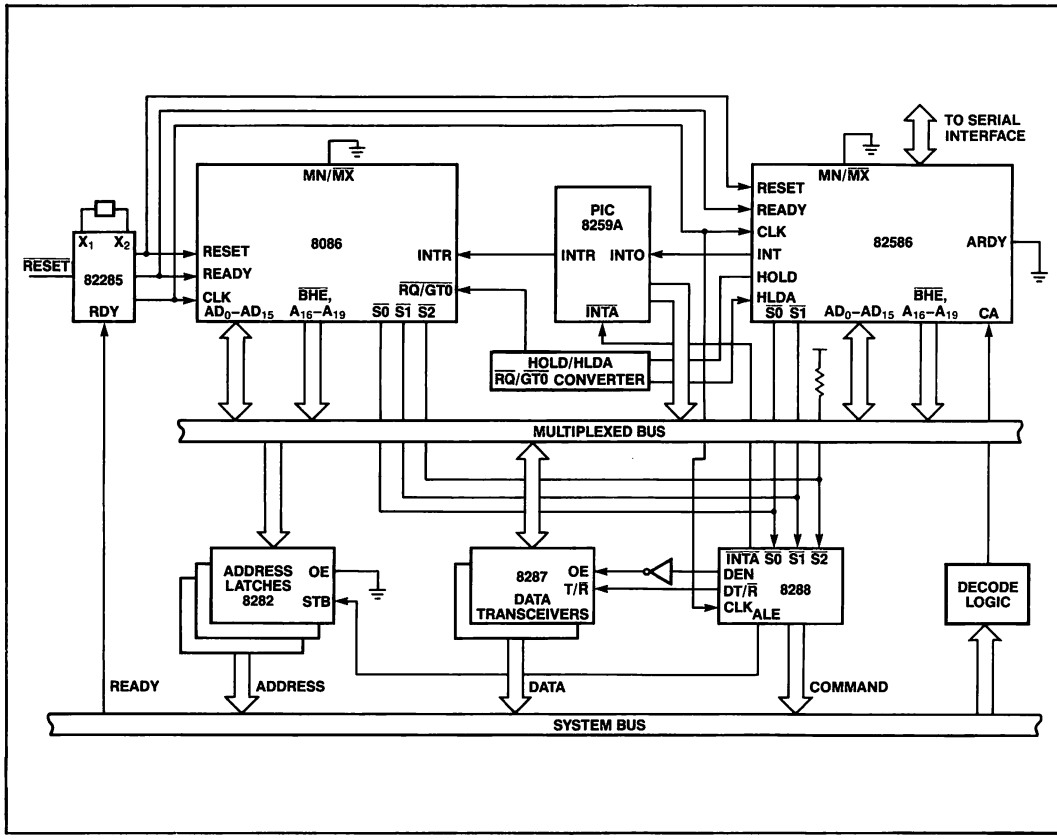


Figure 11. 8086/82586 System

A third system configuration shown in Figure 12 is a dual-port RAM-based system. In this configuration, the bus traffic between the 82586 and shared memory (via port B) is isolated from system bus traffic. Using such a configuration, high-bandwidth peripherals like the 82586 can operate with shared memory using its own local bus, leaving the system bus free for the CPU and other peripherals such as a display controller, with access to shared memory via Port A.

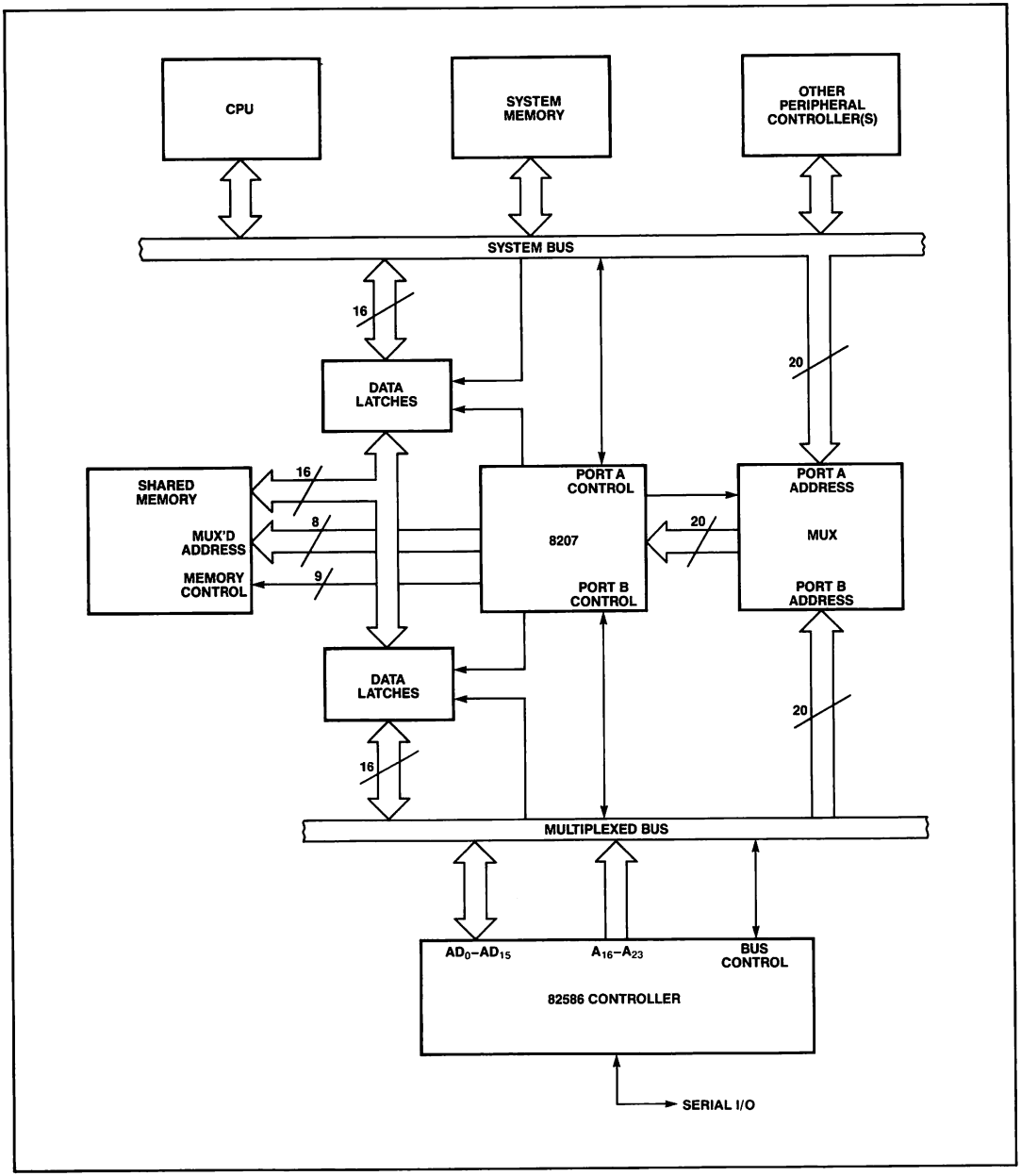


Figure 12. Dual-Port RAM-Based System

Emerging Applications

The availability of low-cost VLSI controllers to serve Ethernet local-area networks is also stimulating other high-speed serial applications. The serial backplane is a form of a local area network within a cabinet. A high-speed "Serial backplane" is used to connect modular subsystems within a box. For example, a high-feature copier will have modular designs for the display/control panel, feeder, sorter, camera and developing functions. By replacing multiconductor busses with a single serial backplane, costs are reduced while improving design flexibility, reliability and modularity for future growth (Figure 13).

A logical extension of the serial backplane within the cabinet is serial connection for local clusters of peripherals around a basic work station, for example (Figure 14). While maintaining a connection to the "public" local-area network, Ethernet, local peripherals such as printers or disk storage can be added to the work station via a separate serial connection. Such serial peripheral connections also result in more flexible system designs by allowing for modular growth. By using the 82586 programmable functions, the frame length, network size and transmission speed can be tailored to the specific serial backplane or peripheral connection environment.

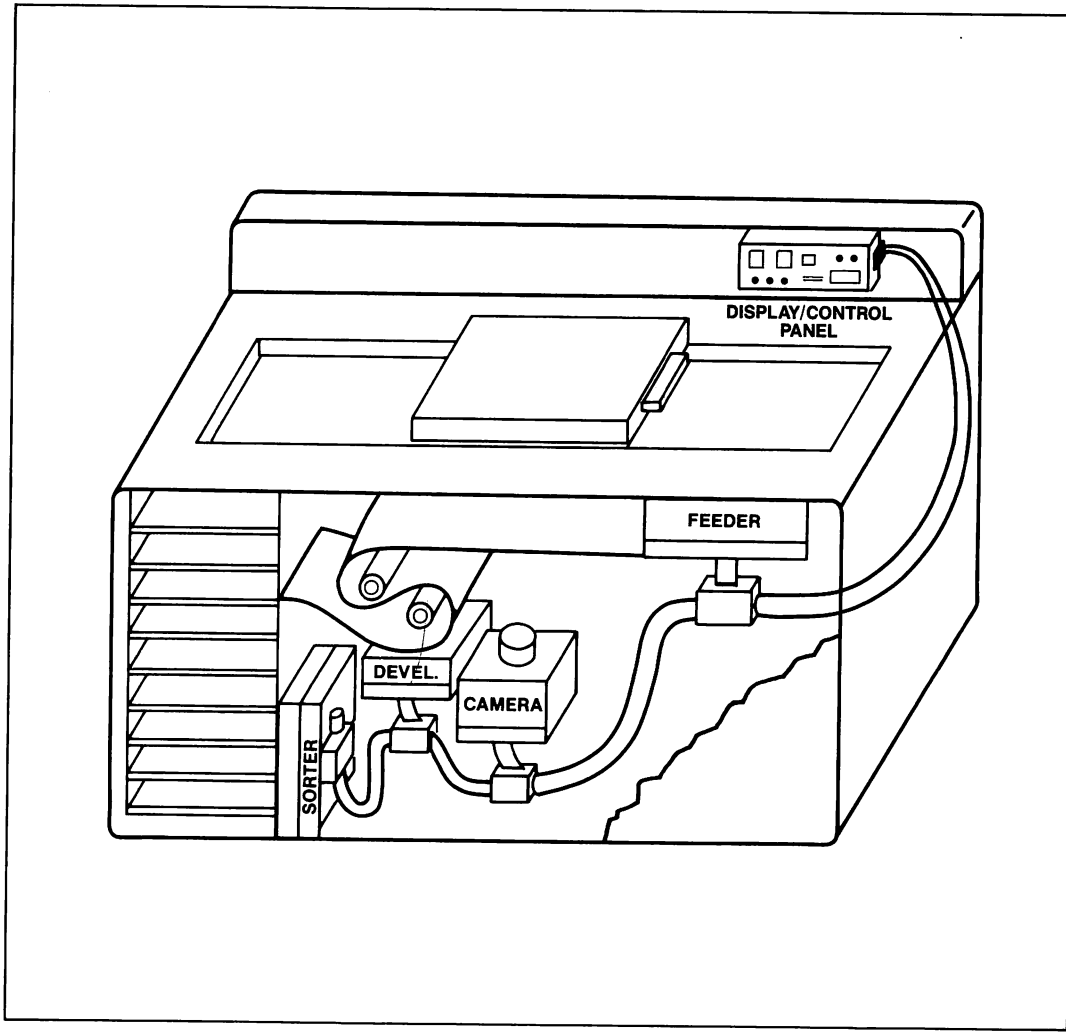


Figure 13. Serial Backplane

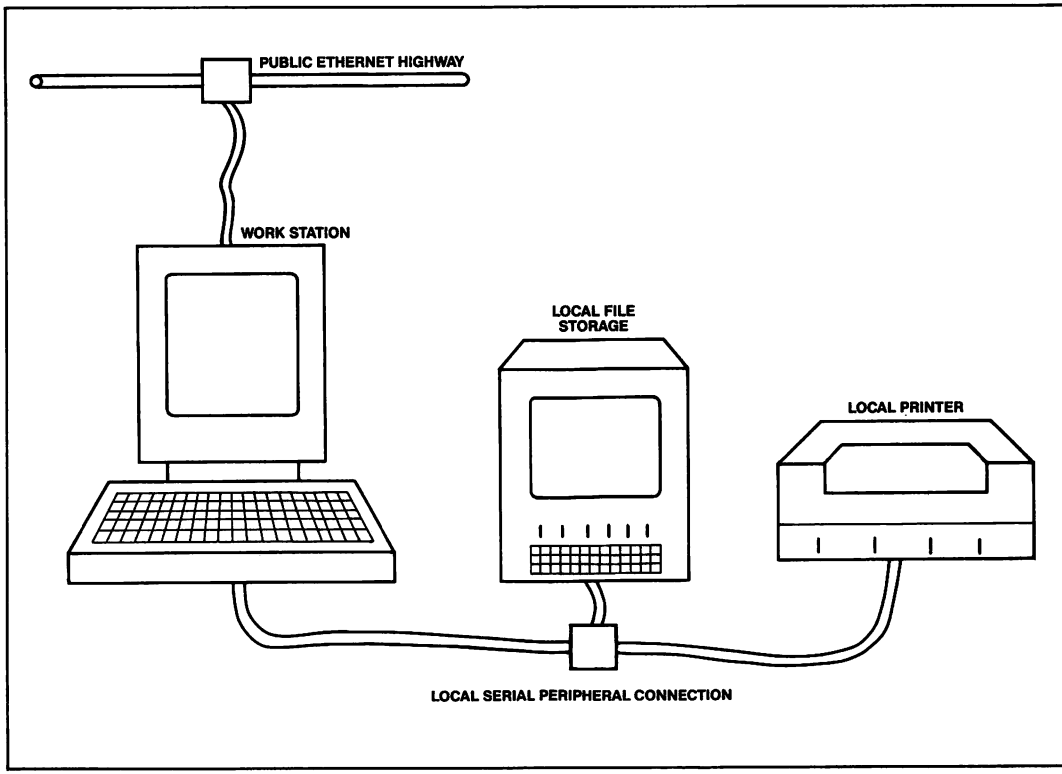


Figure 14. Local Peripheral Interconnection

82501 ETHERNET SERIAL INTERFACE

- **Compatible with the Ethernet* Specification**
- **10-Mbs Operation**
- **Replaces 8 to 12 MSI Components**
- **Manchester Encoding/Decoding and Receive Clock Recovery**
- **10-MHz Transmit Clock Generator**
- **Driving/Receiving Ethernet Transceiver Cable**
- **Fail-Safe Watchdog Timer Circuit to Prevent Continuous Transmissions**
- **Diagnostics Loopback for Fault Detection and Isolation**
- **Directly Interfaces to the 82586 Controller**

The 82501 Ethernet Serial Interface (ESI) chip is designed to work directly with the 82586 controller in Ethernet and non-Ethernet 10-Mbps local-area network applications. The major functions of the 82501 are to generate the 10 MHz transmit clock for the 82586, perform Manchester encoding/decoding of transmitted/received frames, and provide the electrical interface to the Ethernet transceiver cable. Diagnostic loopback control enables the 82501 to route the signal to be transmitted from the 82586 through its Manchester encoding and decoding circuitry and back to the 82586. The combined loopback capabilities of the 82586 and 82501 result in efficient fault detection and isolation by providing sequential testing of the communications interface. An on-chip fail-safe watchdog timer circuit prevents the station from locking up in a continuous transmit mode.

*Ethernet is a trademark of Xerox Corporation.

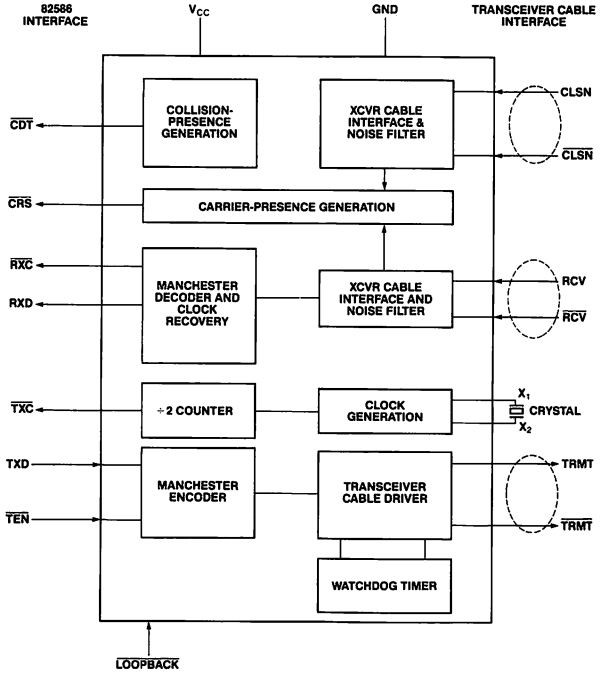


Figure 1. 82501 Functional Block Diagram

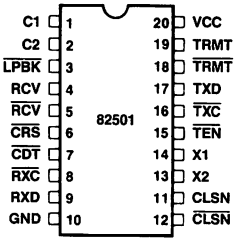


Figure 2. Pin Configuration

OCTOBER 1982
ORDER NUMBER: 210783-001

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
$\overline{TXC}$	16	O	**Transmit Clock:** A 10-MHz clock output with 5 nsec rise and fall times and MOS driving levels. This clock is provided to the 82586 for serial transmission.
$\overline{TEN}$	18	I	**Transmit Enable:** An active low, TTL-level signal synchronous to $\overline{TXC}$ that enables data transmission to the transceiver cable. $\overline{TEN}$ can be driven by $\overline{RTS}$ from the 82586.
TXD	17	I	**Transmit Data:** A TTL-level input signal that is directly connected to the serial data ouput, TXD, of the 82586.
$\overline{RXC}$	8	O	**Receive Clock:** An MOS-level clock output with 5 nsec rise and fall times and 50% duty cycle. This output is connected to the 82586 receive clock input $\overline{RXC}$. There is a maximum 1.2 μsec discontinuity at the beginning of a frame reception when the phase-locked loop switches from the on-chip oscillator to the incoming data. During idle (no incoming frames) the clock frequency will be half that of the 20 MHz crystal frequency.
$\overline{CRS}$	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 asserted when valid data or a collision signal from the transceiver is present. It is deasserted at the end of a frame synchronous with $\overline{RXC}$, or when the end of the collision-presence signal (CLSN and $\overline{CLSN}$) is detected, whichever occurs later. Once deasserted, $\overline{CRS}$ will not be reasserted again for a period of 5 μsec minimum, 7 μsec maximum, regardless of any activity on the receive or collision-presence pairs.
RXD	9	O	**Receive Data:** An MOS-level output tied directly to the RXD input of the 82586 controller and sampled by the 82586 at the negative edge of $\overline{RXC}$. The bit stream received from the transceiver cable is Manchester decoded prior to being transferred to the controller. This output remains high during idle.
$\overline{CDT}$	7	O	**Collision Detect:** A TTL, active low signal which drives the $\overline{CDT}$ input of the 82586 controller. It is asserted as long as there is activity on the collision-presence pair (CLSN and $\overline{CLSN}$).
$\overline{LPBK}$	3	I	**Loopback:** A TTL-level control signal to enable the loopback mode. In this mode, serial data on the TXD input is routed through the 82501 internal circuits and back to the RXD output without driving the TRMT/$\overline{TRMT}$ output pair to the transceiver cable. When $\overline{LPBK}$ is asserted, the collision circuit will also be turned on at the end of each transmission to simulate the collision test.
TRMT $\overline{TRMT}$	19 18	O O	**Transmit Pair:** An output driver 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 $\overline{RCV}$	4 5	I 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 will be negative-going to indicate the beginning of a frame. The last transition should be positive-going, indicating the end of a frame. The received bit stream is assumed to be Manchester encoded.
CLSN $\overline{CLSN}$	12 11	I 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 negative-going to indicate the beginning of the signal; the last transition is positive-going to indicate the end of the signal.
C1 C2	1 2	I I	**PLL Capacitor:** Phase-locked-loop capacitor inputs.
X_1 X_2	14 15	I I	**Clock Crystal:** 20-MHz crystal inputs.
V_{CC}	20		**Power:** 5 ±5% volts.
GND	10		**Ground:** Reference.

FUNCTIONAL DESCRIPTION

Clock Generation

A 20 MHz crystal-controlled oscillator is provided as the basic clock source. This 20 MHz signal is then

divided by 2 to generate a 10 MHz ± .01% clock as required in the Ethernet specification.

Manchester Encoder and Transceiver Cable Driver

The 20 MHz clock is used to Manchester encode data on the TXD input line. The clock is also divided by 2 to produce the 10 MHz clock required by the 82586 for synchronizing its $\overline{RTS}$ and TXD signals. See Figure 3. (Note that the 82586 $\overline{RTS}$ is tied to the 82501 $\overline{TEN}$ input as shown in Figure 4.)

Data encoding and transmission begins with $\overline{TEN}$ going low. Since the first bit is a '1', the first transition on the transmit output TRMT is always negative. Transmission ends with the $\overline{TEN}$ going high. The last transition is always positive at TRMT and may occur at the center of the bit cell (last bit = 1) or at the boundary of the bit cell (last bit = 0). A one-bit delay is introduced by the 82501 between its TXD input and TRMT/$\overline{TRMT}$ output as shown in Figure 3. Following the last transition, the output $\overline{TRMT}$ is slowly brought to its high state so that zero differential voltage exists between TRMT and $\overline{TRMT}$. This will eliminate DC currents in the primary of the transceiver's coupling transformer. See Figure 4.

An internal watchdog timer is started at the beginning of the frame. The duration of the watchdog timer is 25 msec ±15%. If the transmission terminates (by deasserting the $\overline{TEN}$) before the timer expires, the timer is reset (and ready for the next transmission). If the timer expires before the transmission ends, the frame is aborted. This is accomplished by disabling the output driver for the TRMT/$\overline{TRMT}$ pair and deasserting $\overline{CRS}$. RXD and $\overline{RXC}$ are not affected. The watchdog timer is reset only when the $\overline{TEN}$ is deasserted.

The cable driver is a differential gate requiring external resistors or a current sink of 20 mA (on both terminals). In addition, high-voltage protection of 15 volts maximum for 1 second maximum is provided.

Receive Section

CABLE INTERFACE AND NOISE FILTER

The 82501 input circuits can be driven directly from the Ethernet transceiver cable receive pair. In this case the cable is terminated with a pair of 39-ohm resistors in series for proper impedance matching. The center tap of the termination is tied to an external voltage reference (source impedance of 18.5 ohms min.) to establish the required common mode voltage bias for the 82501 receive circuitry. See Figure 4.

The input circuits can also be driven with ECL voltage levels. In either case, the input common mode voltage must be in the range of $V_{CC}-1.0$ to $V_{CC}-2.5$ volts to allow for a wide driver supply variation at the transceiver. The input terminals have a 15-volt maximum protection and additional clamping of low-energy, high-voltage noise signals.

A noise filter is provided at the RCV/$\overline{RCV}$ input pair to prevent spurious signals from improperly triggering the receiver circuitry. The noise filter has the following characteristics:

A negative pulse which is narrower than 30 ns or is less than −150 mV in amplitude is rejected during idle.

At the beginning of a reception, the filter is activated by the first negative pulse which is more negative than −250 mV and is wider than 50 ns.

As soon as the first valid negative pulse is recognized by the noise filter, the $\overline{CRS}$ signal is asserted to inform the 82586 controller of the beginning of a transmission, and the $\overline{RXC}$ will be held low for 1.2 μsec maximum while the internal phase-locked-loop is acquiring lock.

The filter is deactivated if no negative transition occurs within 160 ns from the last positive transition.

Immediately after the end of a reception, the filter blocks all the signals for 5 μsec minimum, 7 μsec maximum. This dead time is required to block-off spurious transitions which may occur on the coaxial cable at the end of a transmission but are not filtered out by the transceiver.

MANCHESTER DECODER AND CLOCK RECOVERY

The filtered data enters the clock recovery and decoder circuits. An analog phase-locked-loop (PLL) technique is used to extract the received clock from the data, beginning from the third negative transition of the incoming data. The PLL will acquire lock within the first 12 bit times, as seen from the RCV/$\overline{RCV}$ inputs. During that period of time, the $\overline{RXC}$ is held low. Bit cell timing distortion which can be tolerated in the incoming signal is ±15 nsec for the preamble and ±20 nsec for data. The voltage-controlled oscillator (VCO) of the PLL corrects its frequency to match the incoming signal transitions.

Its VCO cycle time stays within 5% of the RXD bit cell time regardless of the time distortion allowed at the RCV/$\overline{\text{RCV}}$ input. The RCV/$\overline{\text{RCV}}$ input is decoded from Manchester to NRZ and transferred synchronously with the receive clock to the 82586 controller.

At the end of a frame, the receive clock is used to detect the absence of RCV/$\overline{\text{RCV}}$ transitions and report it to the 82586 by deasserting $\overline{\text{CRS}}$ while $\overline{\text{RXD}}$ is held high.

Collision-Presence Section

The CLSN/$\overline{\text{CLSN}}$ input signal is a 10 MHz ±15% square wave generated by the transceiver whenever two or more data frames are superimposed on the coaxial cable. The maximum asymmetry in the CLSN/$\overline{\text{CLSN}}$ signal is 60/40% for low-to-high or high-to-low levels. This signal is filtered for noise rejection in the same manner as RCV/$\overline{\text{RCV}}$. The noise filter rejects signals which are less negative than −150 mV and narrower than 15 ns during idle. It turns on at the first negative pulse which is more negative than −250 mV and wider than 30 ns. After the initial turn-on, the filter remains active indicating that a valid collision signal is present, as long as the negative CLSN/$\overline{\text{CLSN}}$ signal pulses are more negative than −250 mV. The filter returns to the "off" state if the signal becomes less negative than −150 mV, or if no negative transition occurs within 160 ns from the last positive transition. Immediately after turn-off, the collision filter is ready to be reactivated.

The common mode voltage and external termination are identical to the RCV/$\overline{\text{RCV}}$ input. (See Figure 4.) The CLSN/$\overline{\text{CLSN}}$ input also has a 15-volt maximum protection and additional clamping against low-energy, high-voltage noise signals.

A valid collision-presence signal will assert the 82501 $\overline{\text{CDT}}$ output which can be directly tied to the $\overline{\text{CDT}}$ input of the 82586 controller.

During the time that valid collision-presence transitions are present on the CLSN/$\overline{\text{CLSN}}$ input, invalid data transitions will be present on the receive data pair due to the superposition of signals from two or more stations transmitting simultaneously. It is possible for RCV/$\overline{\text{RCV}}$ to lose transitions for a few bit times due to perfect cancellation of the signals. In any case, the invalid data will not cause any discontinuity of RXC.

When a valid collision-presence signal is present the $\overline{\text{CRS}}$ signal is asserted (along with $\overline{\text{CDT}}$). However, if this collision-presence signal arrives within 6.0 ±1.0 μs from the time $\overline{\text{CRS}}$ was deasserted, only $\overline{\text{CDT}}$ is generated.

Internal Loopback

When asserted, $\overline{\text{LPBK}}$ causes the 82501 to route serial data from its TXD input, through its transmit logic (retiming and Manchester encoding), returning it through the receive logic (Manchester decoding and receive clock generation) to RXD output. The internal routing prevents the data from passing through the output drivers and onto the transmit output pair, TRMT/$\overline{\text{TRMT}}$. When in loopback mode, all of the transmit and receive circuits, including the noise filter, are tested except for the transceiver cable output driver and input receivers. Also, at the end of each frame transmitted in loopback mode, the 82501 generates a 1-μsec $\overline{\text{CDT}}$ signal within 1 μsec after the end of the frame. Thus, the collision circuits, including the noise filter, are also tested in loopback mode. The watchdog timer remains enabled in loopback mode, terminating test frames that exceed its time-out period.

In the normal mode ($\overline{\text{LPBK}}$ not asserted), the 82501 operates as a full duplex device, being able to transmit and receive simultaneously. This is similar to the external loopback mode of the 82586. Combining the internal and external loopback modes of the 82586 and the internal loopback and normal modes of the 82501, incremental testing of an 82586/82501-based interface can be performed under program control for systematic fault detection and fault isolation.

Interface Example

The 82501 is designed to work directly with the 82586 controller in Ethernet as well as non-Ethernet 10 Mbps LAN applications. The control and data signals connect directly between the two devices without the need for additional external logic. The complete 82586/82501/Ethernet Transceiver cable interface is shown in FIGURE 4. The 82501 provides the driver and receivers needed to directly connect to the transceiver cable, requiring only terminating resistors on each input signal pair.

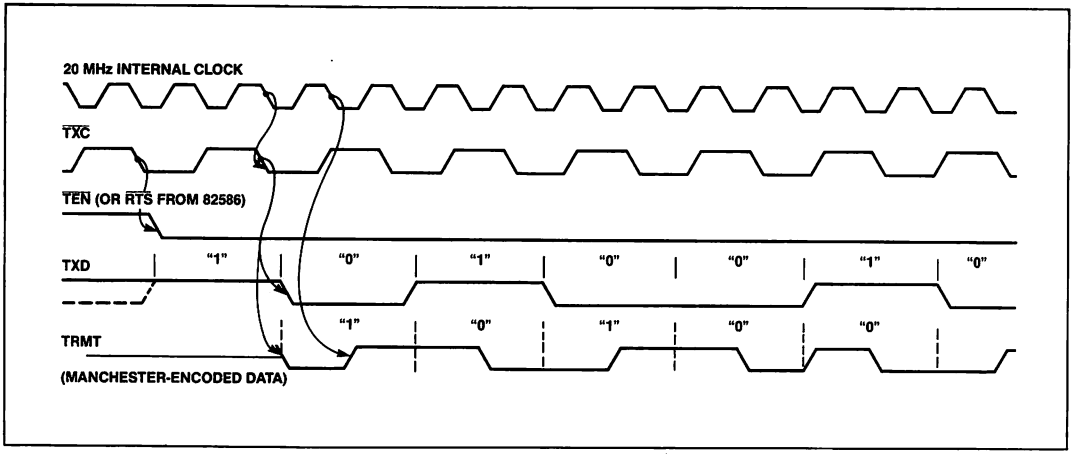


Figure 3. Start of Transmission and Manchester Encoding

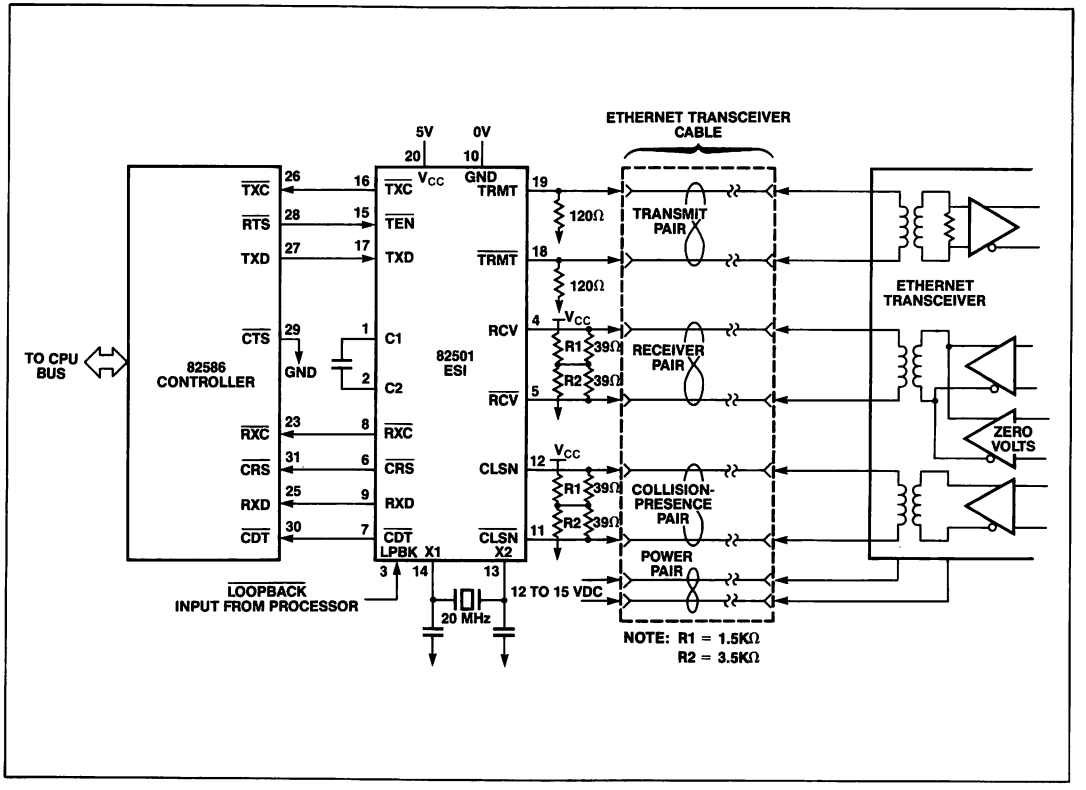


Figure 4. 82586/82501/Transceiver Cable Interface

D.C. CHARACTERISTICS (TA = 0–70°C, V_{CC} = 5V ±10%)

Symbol	Parameter	Min.	Max.	Units	Conditions
V_{IL}	Input Low Voltage (TTL)	−0.5	+0.8	V	
V_{IH}	Input High Voltage (TTL)	2.0	V_{CC} + 0.5	V	
V_{IDF}	Input Differential Voltage (Differential)	±250	±1000	mV	
V_{CM}	Input Common Mode Voltage (Differential)	V_{CC} − 2.5	V_{CC} − 1.0	V	
V_{OL}	Output Low Voltage TTL or MOS		0.45	V	I_{OL} = 8 mA
V_{OCM}	Common Mode Output	1.0	4.5	V	R_L = 78 Ohms Differential Termination and 120Ω pulldown
V_{OH}	Output High Voltage				
	TTL	2.4		V	I_{OH} = −1.0 mA
	MOS	3.9		V	I_{OH} = −400 μA
V_{ODF}	Differential Output Swing	0.55	1.2	V	R_L = 78 Ohms Differential Termination and 120Ω pulldown
I_{LI}	Input Leakage Current		±200	μA	$0 < V_{IN} < V_{CC}$
C_{IN}	Input Capacitance		10	pF	f = 1 MHz
C_{OUT}	Output Capacitance		20	pF	f = 1 MHz
I_{CC}			200	mA	

A.C. CHARACTERISTICS

A.C. Measurement Conditions

I) T_A = 0° to 70°C, V_{CC} = 5V ± 10%

II) The AC measurements are done at the following voltage levels for the various kinds of inputs and outputs

a) TTL inputs and oututs: 0.8V and 2.0V
The input voltage swing is 0.4 to 2.4V at least with 3–10 ns rise and fall times.

b) MOS outputs: The rise and fall times are measured between 0.6V and 3.6V points. The high time is measured between 3.6V points and the low time is measured between 0.6V points.

c) Differential inputs and outputs:
The 50% points of the total swing are used for delay measurements. The rise and fall times of outputs are measured at the 20 to 80% points. The differential voltage swing at the inputs is at least ±250 mV with rise and fall times of 3–15 ns measured at ±.2 volts.

III) The AC loads for the various kind of outputs are as follows:

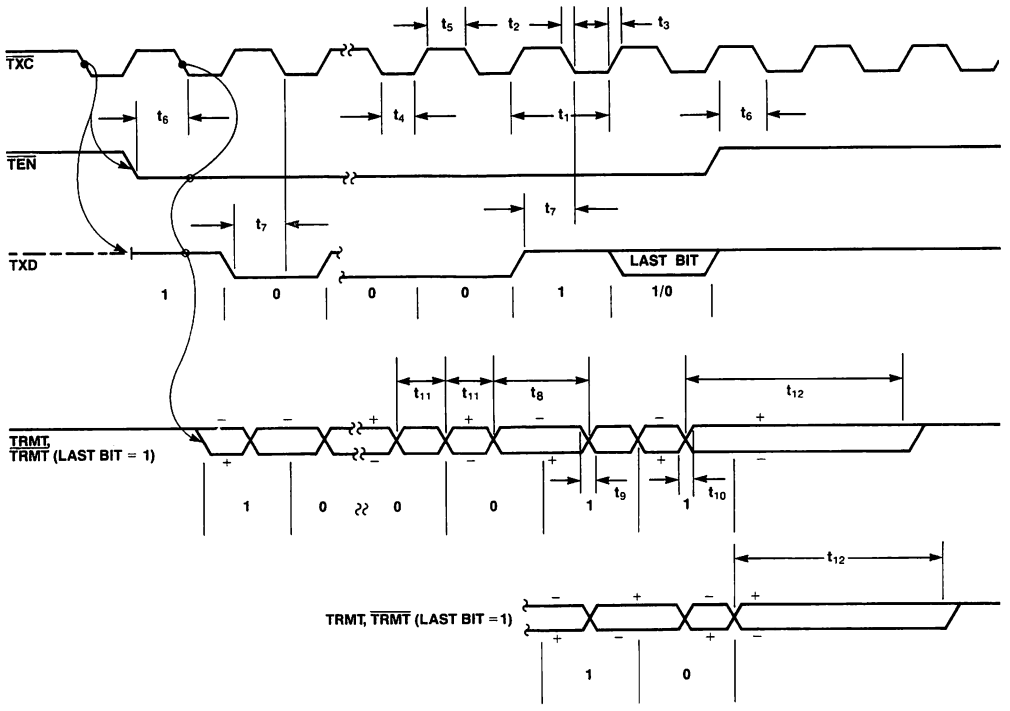
a) TTL and MOS: A 20-pF Capacitor to GND

b) Differential: A 10-pF Capacitor from each terminal to GND and a termination load resistor of 78 ohms in parallel with a 27 microhenries inductor between the two terminals.

TRANSMIT TIMING

Symbol	Parameter	Min.	Max.	Unit
t_1	$\overline{TXC}$ Cycle Time	99.99	100.01	ns
t_2	$\overline{TXC}$ Fall Time		5	ns
t_3	$\overline{TXC}$ Rise Time		5	ns
t_4	$\overline{TXC}$ Low Time	40		ns
t_5	$\overline{TXC}$ High Time	40		ns
t_6	Transmit Enable/Disable to $\overline{TXC}$ Low	50		ns
t_7	TXD Stable to $\overline{TXC}$ Low	50		ns
t_8	Bit Cell Center to Bit Cell Center of Transmit Pair Data	99.5	100.5	ns
t_9	Transmit Pair Data Fall Time	1.0	5.0	ns
t_{10}	Transmit Pair Data Rise Time	1.0	5.0	ns
t_{11}	Bit Cell Center to Bit Cell Boundary of Transmit Pair Data	49.5	50.5	ns
t_{12}	$\overline{TRMT}$ starts approaching its high level from Last Positive Transition of Transmit Pair Data during idle.	160		ns

TRANSMIT TIMING



AFN-00884A

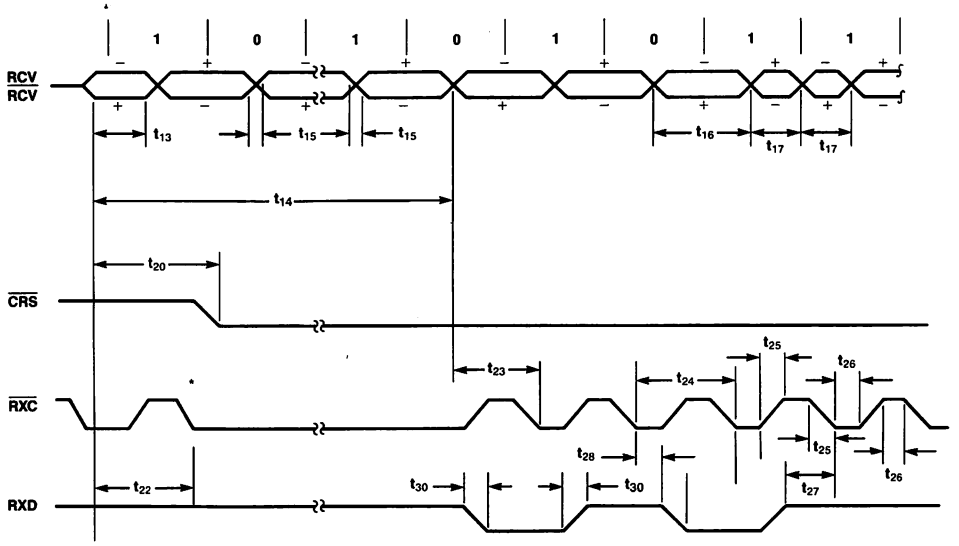
RECEIVE TIMING

Symbol	Parameter	Min.	Max.	Unit
t_{13}	Receive Pair Signal Pulse Width (at −.25V differential signal) of First Negative Pulse for a) Signal Rejection by Noise Filter, b) Noise Filter Turn-on in order to Begin Reception	 50	30	ns ns
t_{14}	Duration which the $\overline{RXC}$ is held at low state		1200	ns
t_{15}	Receive Pair Signal Rise/Fall Time at ±.2 volt		15	ns
t_{16}	Receive Pair Signal Bit Cell Center to Bit Cell Center allowing for timing distortion In preamble In data	 70 60	 130 140	 ns ns
t_{17}	Receive Pair Signal Bit Cell Center to Bit Cell Boundary allowing for timing distortion In preamble In data	 20 10	 80 90	 ns ns
t_{18}	Receive Idle Time Before the Next Reception can Begin (as measured from the deassertion of $\overline{CRS}$)		8	μs
t_{19}	Receive Pair Signal Return to Zero Level from Last valid Positive Transition	0.20	5	μs
t_{20}	$\overline{CRS}$ Assertion delay from the First received valid Negative Transition of Receive Pair Signal		100	ns

Symbol	Parameter	Min.	Max.	Unit
t_{21}	$\overline{CRS}$ Deassertion delay from the Last valid positive transition received (when no Collision-Presence signal exists on the transceiver cable)		300*	ns
t_{22}	$\overline{RXC}$ stopped from the first valid negative transition of RCV/$\overline{RCV}$		150	ns
t_{23}	$\overline{RXC}$ delay from RCV/$\overline{RCV}$ Bit Cell Center		100	ns
t_{24}	$\overline{RXC}$ Jitter		±5.0	ns
t_{25}	$\overline{RXC}$ Rise/Fall time		5	ns
t_{26}	$\overline{RXC}$ High/Low time	40		ns
t_{27}	Receive Data Stable before the Negative Edge of $\overline{RXC}$	30		ns
t_{28}	Receive Data Held valid past the Negative Edge of $\overline{RXC}$	30		ns
t_{29}	Carrier Sense deasserted before the Negative Edge of $\overline{RXC}$	10	30	ns
t_{30}	Receive data Rise/Fall time		10	ns
t_{31}	From the time $\overline{CRS}$ is deasserted until the time it can be asserted again	5	7	μs

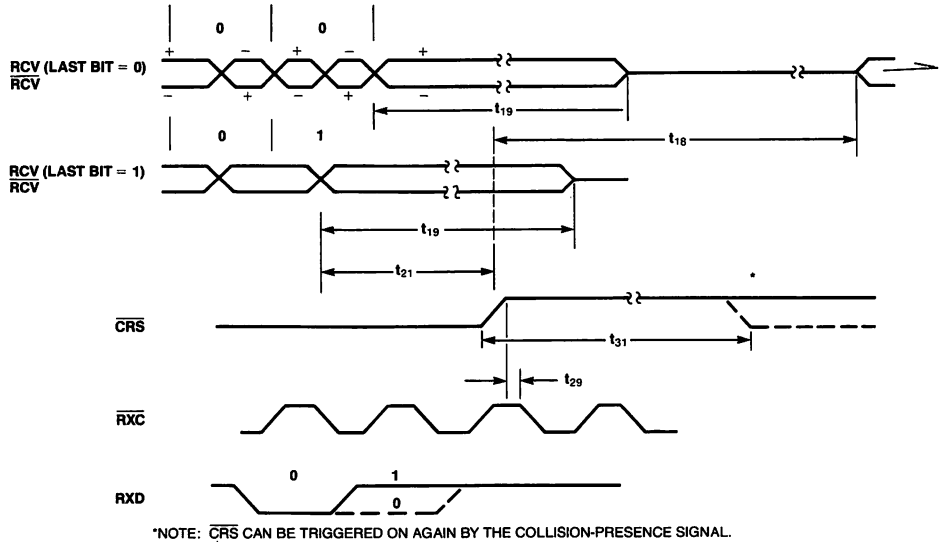
***NOTE:**
$\overline{CRS}$ is deasserted synchronously with the $\overline{RXC}$. This condition is not specified in the Ethernet specification.

RECEIVE TIMING: START OF FRAME



*THIS CLOCK PULSE MAY NOT BE A VALID CLOCK PULSE.

RECEIVE TIMING: END OF FRAME

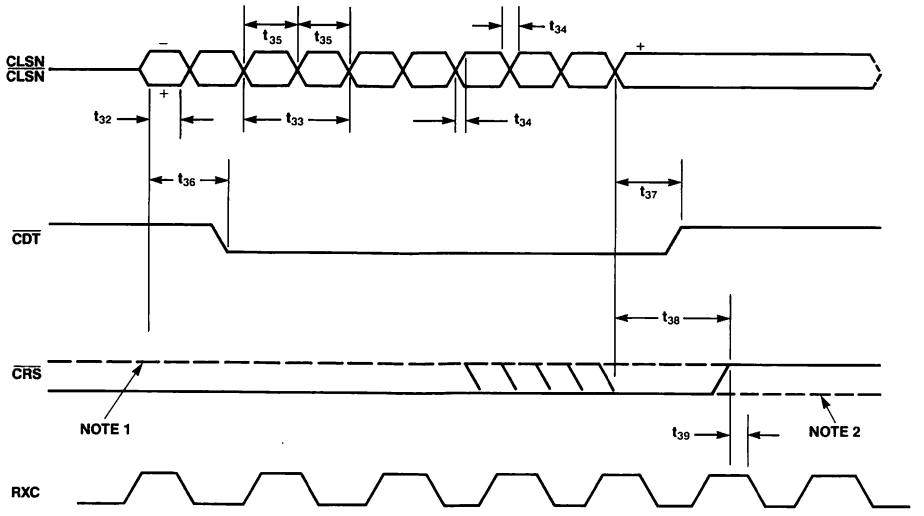


*NOTE: CRS CAN BE TRIGGERED ON AGAIN BY THE COLLISION-PRESENCE SIGNAL.

COLLISION TIMING

Symbol	Parameter	Min.	Max.	Unit
t_{32}	CLSN/$\overline{\text{CLSN}}$ Signal Pulse Width (at −.25V differential signal) of First Negative Pulse for Noise Filter Turn-on	30		ns
t_{33}	CLSN/$\overline{\text{CLSN}}$ Cycle Time	86	118	ns
t_{34}	CLSN/$\overline{\text{CLSN}}$ Rise/Fall Time at ±.2 volts		15	ns
t_{35}	CLSN/$\overline{\text{CLSN}}$ Transition Time	35	70	ns
t_{36}	$\overline{\text{CDT}}$ Assertion from the First Valid Negative Edge of Collision Pair Signal		75	ns
t_{37}	$\overline{\text{CDT}}$ Deassertion from the Last Positive Edge of CLSN/$\overline{\text{CLSN}}$ Signal		200	ns
t_{38}	$\overline{\text{CRS}}$ Deassertion from the Last Positive Edge of CLSN/$\overline{\text{CLSN}}$ signal (If no post-collision signal remains on the receive pair.)		350	ns
t_{39}	$\overline{\text{CRS}}$ stable before the negative edge of $\overline{\text{RXC}}$ at deassertation	10	30	ns

COLLISION TIMING



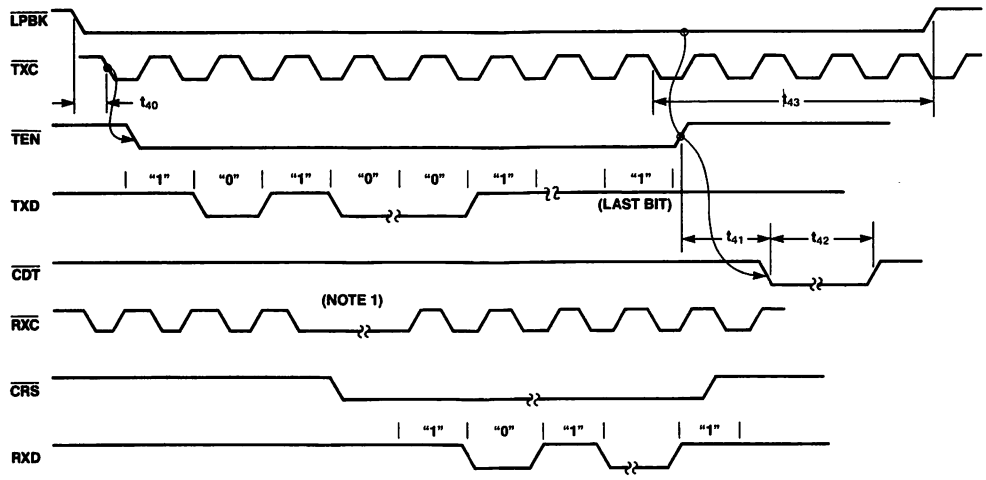
NOTES:
1. $\overline{\text{CRS}}$ WILL BE DEASSERTED FOR A PERIOD UP TO 7 μSEC MAXIMUM WHEN RCV/$\overline{\text{RCV}}$ OR CLSN/$\overline{\text{CLSN}}$ TERMINATES, WHICHEVER OCCURS LATER.
2. $\overline{\text{CRS}}$ WILL REMAIN ASSERTED AFTER THE CLSN/$\overline{\text{CLSN}}$ SIGNAL TERMINATES IF RCV/$\overline{\text{RCV}}$ SIGNALS CONTINUE.

LOOPBACK TIMING

Symbol	Parameter	Min.	Max.	Unit
t_{40}	$\overline{\text{LPBK}}$ asserted before the first attempted transmission	500		ns
t_{41}	Simulated collision test delay from the end of each attempted transmisssion	.5	1.0	μs
t_{42}	Simulated collision test duration	.5	1.0	μs
t_{43}	$\overline{\text{LPBK}}$ deasserted after the last attempted transmission	5		μs

NOTE:
In Loopback mode, $\overline{\text{RXC}}$, $\overline{\text{RXD}}$ and $\overline{\text{CRS}}$ function in the same manner as a normal Receive.

LOOPBACK TIMING



NOTE:
1. DURING LOOPBACK, THE 82501 RECEIVE CIRCUITRY USES 12 BIT TIMES WHILE THE PLL LOCKS ON THE DATA. AS A RESULT, THE FIRST 12 BITS ARE LOST.

TECHNICAL REFERENCES

6.1 The Ethernet Specification Version 1.0, August, 1980 Digital Equipment Corporation, Intel, Xerox
6.2 ISO "Reference Model for Open Systems Interconnection Architecture", ISO/TC97/SC16 N309
6.3 Intel "Local Network Architecture Proposed For Work Stations" August, 1981 Article Reprint 186
6.4 Intel iNA 950-1 Local Area Network Software Data Sheet
6.5 Intel iSBC 550 Ethernet MultiBus Controller Board Set Data Sheet
6.6 Intel "System Level Functions Ehance Controller IC", October 6, 1982 Electronics

U.S. SALES OFFICES

October 1982

ALABAMA*

Intel Corp.
303 Williams Avenue, S.W.
Suite 1422
Huntsville 35801
Tel: (205) 533-9353

ARIZONA

Intel Corp.
10210 N. 25th Avenue,
Suite 11
Phoenix 85021
Tel: (602) 869-4980

CALIFORNIA

Intel Corp.
1010 Hurley Way
Suite 300
Sacramento 95825
Tel: (916) 929-4078

Intel Corp.
7670 Opportunity Road
Suite 135
San Diego 92111
Tel: (714) 268-3563

Intel Corp.*
2000 East 4th Street
Suite 100
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*
3375 Scott Boulevard
Santa Clara 95051
Tel: (408) 987-8086
TWX: 910-339-9279
910-338-0255

Intel Corp.*
5530 Corbin Avenue
Suite 120
Tarzana 91356
Tel: (213) 708-0333
TWX: 910-495-2045

COLORADO

Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (303) 594-6622

Intel Corp.*
650 S. Cherry Street
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

Intel Corp.
36 Padanaram Road
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

EMC Corp.
48 Purnell Place
Manchester 06040
Tel: (203) 646-8085

EMC Corp.
393 Center Street
Wallingford 06492
Tel: (203) 265-6991

FLORIDA

Intel Corp.
1500 N.W. 62nd Street
Suite 104
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

Intel Corp.
500 N. Maitland
Suite 205
Maitland 32751
Tel: (305) 628-2393
TWX: 810-853-9219

GEORGIA

Intel Corp.
3300 Holcomb Bridge Road
Suite 225
Norcross 30092
Tel: (404) 449-0541

ILLINOIS

Intel Corp.*
2550 Golf Road
Suite 815
Rolling Meadows 60008
Tel: (312) 981-7200
TWX: 910-651-5881

INDIANA

Intel Corp.
9100 Purdue Road
Suite 400
Indianapolis 46268
Tel: (317) 875-0623

IOWA

Intel Corp.
St. Andrews Building
1930 St. Andrews Drive N.E.
Cedar Rapids 52402
Tel: (319) 393-5510

KANSAS

Intel Corp.
8400 W. 110th Street
Suite 170
Overland Park 66210
Tel: (913) 642-8080

LOUISIANA

Industrial Digital Systems Corp.
2332 Severn Avenue
Suite 202
Metairie, LA 70001
Tel: (504) 831-8492

MARYLAND

Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Intel Corp.
1620 Elton Road
Silver Spring 20903
Tel: (301) 431-1200

MASSACHUSETTS

Intel Corp.*
27 Industrial Avenue
Chelmsford 01824
Tel: (617) 256-1800
TWX: 710-343-6333

EMC Corp.
381 Elliot Street
Newton 02164
Tel: (617) 244-4740
TWX: 922531

MICHIGAN

Intel Corp.*
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 810-244-4915

MINNESOTA

Intel Corp.
3500 W. 80th Street
Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

Intel Corp.*
Raritan Plaza III
Raritan Center
Edison 08837
Tel: (201) 225-3000
TWX: 710-480-6238

NEW MEXICO

BFA Corp.
P.O. Box 1237
Las Cruces 88001
Tel: (505) 523-0601
TWX: 910-983-0543

BFA Corp.
3705 Westerfield, N.E.
Albuquerque 87111
Tel: (505) 292-1212
TWX: 910-989-1157

NEW YORK

Intel Corp.*
300 Motor Parkway
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.
80 Washington Street
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.*
211 White Spruce Boulevard
Rochester 14623
Tel: (716) 424-1050
TWX: 510-253-7391

T-Squared
4054 Newcourt Avenue
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared
7353 Pittsford
Victor Road
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA

Intel Corp.
2306 W. Meadowview Road
Suite 206
Greensboro 27407
Tel: (919) 294-1541

OHIO

Intel Corp.*
6500 Poe Avenue
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
Chagrin-Brainard Bldg., No. 300
28001 Chagrin Boulevard
Cleveland 44122
Tel: (216) 464-2736
TWX: 810-427-9298

OKLAHOMA

Intel Corp.
4157 S. Harvard Avenue
Suite 123
Tulsa 74101
Tel: (918) 749-8688

OREGON

Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 324
Beaverton 97005
Tel: (503) 641-8086
TWX: 910-467-8741

PENNSYLVANIA

Intel Corp.*
510 Pennsylvania Avenue
Fort Washington 19034
Tel: (215) 641-1000
TWX: 510-661-2077

Intel Corp.*
201 Penn Center Boulevard
Suite 301W
Pittsburgh 15235
Tel: (412) 823-4970

Q.E.D. Electronics
300 N. York Road
Hatboro 19040
Tel: (215) 674-9600

TEXAS

Intel Corp.*
12300 Ford Road
Suite 380
Dallas 75234
Tel: (214) 241-8087
TWX: 910-860-5617

Intel Corp.*
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490

Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713) 988-9421

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

UTAH

Intel Corp.
268 West 400 South
Salt Lake City 84101
Tel: (801) 533-8086

VIRGINIA

Intel Corp.
1501 Santa Rosa Road
Suite C-7
Richmond 23288
Tel: (804) 282-5668

WASHINGTON

Intel Corp.
110 110th Avenue N.E.
Suite 510
Bellevue 98005
Tel: (206) 453-8086
TWX: 910-443-3002

WISCONSIN

Intel Corp.
150 S. Sunnyslope Road
Brookfield 53005
Tel: (414) 784-9060

*Field Application Location