AN ABSTRACT OF THE THESIS OF

Tae-On Yoo for the degree of Master of Science in Electrical and Computer Engineering presented on May 13, 1988.

Title: Implementation of an Ethernet Local Area Network Board for the MacIntosh II System

Abstract approved: Redacted for Privacy
Roy C. Rathja

The purpose of this study was to design and to implement a practical Ethernet Local Area Network (LAN) board for the Macintosh II computer system. A survey of the literature provided a general definition of LANs, a reference model for network layers, specifications for an Ethernet LAN, and information on the Macintosh II system architecture.

Detailed information was provided for the design of the hardware and problems of programming the board. The design of the board conforms to the NuBus specification used in the MacIntosh II. The board consists of three general sections: (1) the NuBus interface block, containing a configuration Read Only Memory (ROM) for the Macintosh II; (2) the LAN coprocessor and its chip set; and (3) shared memory accessible by both the Macintosh II and the coprocessor.

The major design concern was the logic block for the arbitration of shared memory access between the Macintosh II and the coprocessor. Most of the control logic block consists of Programmable Array Logic (PAL) chips, giving the system flexibility and allowing for a reduction in the number of components used on the board.

Implementation of an Ethernet Local Area Network Board for the MacIntosh II System

by

Tae-On Yoo

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed May 13, 1988

Commencement June 1989

APPROVED:

Redacted for Privacy

Associate Professor of Electrical and Computer Engineering in charge of major

Redacted for Privacy

Head of Department of Electrical and Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented May 13, 1988

Typed by B. McMechan for Tae-On Yoo

Acknowledgements

I wish to thank my major advisor, Professor Roy C. Rathja, for his valuable advice. I would also like to thank Professors Rudolf S. Engelbrecht and James H. Herzog for their fruitful suggestions and to express my gratitude to my company for providing the research environment for this thesis.

I would like to acknowledge my friend, Hongsin Lee, for reading portions of this thesis. Lastly, I wish to thank my mother and my wife, Hyunsuk, for their support and patience. Without their support, this thesis would not have been completed.

Table of Contents

Table of Contents
(continued)

Page

List of Figures

List of Figures
(continued)

List of Tables

IMPLEMENTATION OF AN ETHERNET LOCAL AREA NETWORK BOARD FOR THE MACINTOSH II SYSTEM

CHAPTER 1: INTRODUCTION

A local area network (LAN) is a communication network intended to link processing elements within a restricted geographical area. Over the past decade LANs have become increasingly popular and the Ethernet LAN has been widely adopted. Ethernet is used to distribute data packets through a common medium with no dedicated central control. Each interconnected workstation on an Ethernet LAN has the intelligence to contend with other workstations on the network for the transmission of its data packets directed to destination addresses. The Macintosh II computer system, produced by Apple Computer, Inc., is the most powerful member of the Macintosh family. The Macintosh II system is composed of a full 32-bit microprocessor, six NuBus expansion slots for various adapters, and a number of other characteristics useful for adaptation to LANs.

This project analyzes the development of a practical implementation of an Ethernet interface board for

the Macintosh II. The board under discussion is provided with IEEE 802.3 type 10BASE5 (Ethernet) and type 10BASE2 (Thin-net or Cheapernet) connections for Macintosh II systems.

Chapter 2 includes descriptions of the basic terms used in subsequent chapters, e.g., local area networks or LANs, the ISO reference model, Ethernet and IEEE 802.3 standards, the Macintosh II system, and Intel's local area network coprocessor. Chapter 3 offers consideration of a practical hardware design for the Macintosh II Ethernet interface board, including analysis of some of its logic blocks and features. Chapter 4 is a discussion of the implementation procedures used to program the board, including initialization and specific programming procedures. Conclusions drawn from preparation of this project are in Chapter 5.

CHAPTER 2: REVIEW OF LITERATURE

2.1 Introduction

This chapter provides general information on the structure and usage of local area networks (LANs), the OSI reference model, the Ethernet and IEEE 802.3 specifications, the Apple Macintosh II system, and the Intel 82586 LAN coprocessor.

2.2 Local Area Networks

The actual research for the structuring of computer networks was begun in the late 1960s with the ARPANET project, aided by the Advanced Research Projects Agency of the U.S. Department of Defense. The result of this research has affected modern networking technologies and the appearance of low-cost microprocessors has led to the development of LAN applications in various industries. As the need for interactive data processing has grown within industry, the attractiveness of computer networks within a restricted geographical area has become increasingly obvious.

2.2.1 Overview

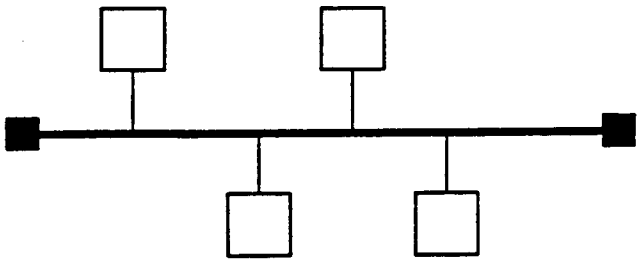
The functional definition of a LAN falls into the existing region between multiprocessing systems and long-haul networks [22]. The LAN may have 100 to 10,000 meters distance between interprocessors, with 0.1 to 10 megabits per second as a bit rate (see Table 2.1). LANs are designed to fulfil as many of the following goals as possible [15]: (1) high data transmission rates; (2) low error rates; (3) inexpensive transmission media; (4) simple interface connections; and (5) interconnection with other networks.

Table 2.1 Classification of Networks [10,19].

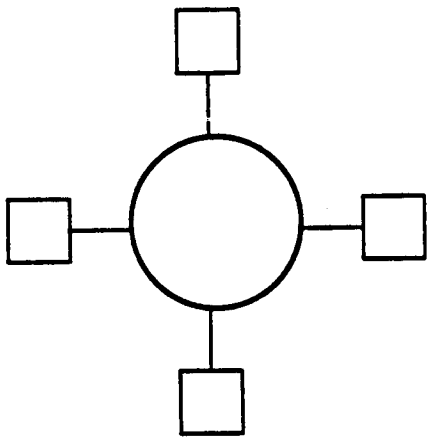
Activity	Separation	Bit Rate
Remote Networks	>10 Km	< 0.1 Mbps
Local Networks	10 - 0.1 Km	0.1 - 10 Mbps
Multiprocessors	< 0.1 Km	> 10 Mbps

2.2.2 Network Topology

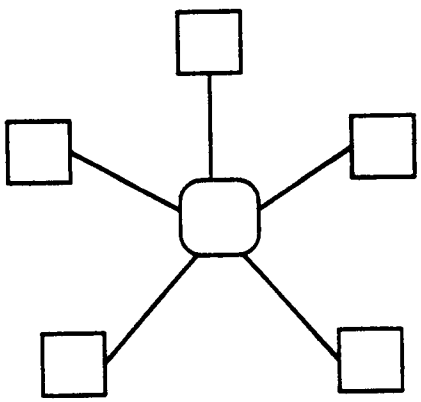
The structure of the three most popular LAN topology types, the topologies of which are discussed in the following three sections, are illustrated in Figure 2.1.



(1) Bus Network



(2) Ring Network



(3) Star Network

Figure 2.1 Network Topologies.

1) Bus Network. This topology is a very popular approach to LAN installation. A bus network consists of a common channel for the transportation of information between processing elements connected to the channel. The transmission medium for the channel can be twisted pairs, coaxial cable, or optical fiber. All nodes attached to the channel share the transmission bus with a unique address attributed to each node. Since there is only one transmission channel, this topology uses time multiplexing or random techniques to determine transmission precedence. Ethernet is one of the most popular LANs using the bus network topology.

2) Ring Network. The ring network topology interconnects processing elements via a loop, but without a specifically responsible loop controller [15,19]. The major features of the ring are equal time sharing between nodes, a simple ring control scheme, and a simple medium interface. Each node receives a token with messages from one of its neighbors and sends messages to its other neighbors. Hence, messages are circulated around the loop from their

source to their destination. The IBM PC network is one of the most popular ring topology networks.

3) Star Network. This topology is a common scheme used for installation of networks connected to a mainframe computer. Hence, the central system is the only shared resource and it has only a limited number of devices attached. This scheme is not similar to true local area networks, however, the modified star topology form uses a hub as a switch for the connection of processing elements. AT&T's Starlan is a popular star topology LAN. The advantage of this topology is inexpensive installation, made possible by the use of telephone wire communication connections.

2.2.3 Medium Access Schemes

There are three general medium access schemes in LANs: CSMA/CD, token-bus, and token-ring.

2.2.3.1 CSMA/CD: Carrier Sense Multiple Access (CSMA) is a contentious method of communication, presented in two modified forms: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). All CSMA devices on a network listen to the transmission medium before accessing it; if there is no

transmission, then any of the devices may transmit a message. This method necessitates that some devices retransmit their message packets, due to the late recognition of message collisions. Hence, the system does encompass a great deal of wasted time.

The most common networking medium access scheme used for bus LANs is the CSMA/CD. In CSMA/CD all devices listen to the medium before and during transmissions, minimizing wasted time [15].

CSMA/CA is a combined form of a slotted time division multiplexing system and CSMA/CD. The slotted time division is usually used for transmitting packets. However, when no devices are ready to transmit in their appointed time slots, the network is said to be in the CSMA/CD mode.

2.2.3.2 Token-Bus: This scheme is a non-contentious medium access method. The token-bus uses a unique frame, called the token, to attribute to a device the right to use the medium. This scheme provides a logical ring configuration with a physical bus structure, but it includes problems which must be solved, such as token loss and ring configuration [13].

2.2.3.3 Token-Ring. The token-ring medium access scheme uses a physical ring topology. A control token is a packet of information passed through the network. Any node that has the token uses the medium exclusively and every node retransmits received packets until a

free token received, indicating availability of the medium. If a node has a data packet ready when a free token is received, it sends the packet, accompanied by a token, to the next node, which then continues retransmission [8,13,18]. Token loss also constitutes a problem for this network.

2.3 The ISO Reference Model

The International Standards Organization (ISO) has produced a reference model for computer networks called the Open Systems Interconnection (OSI) reference model.

2.3.1 Overview

The object of the OSI model is to remove technical obstacles to communication between different systems. This is accomplished by standardizing the rules of interaction between interconnected systems. Therefore, the model is not concerned with the internal operation of any single system, but only with the information exchange between interconnected systems [15].

The architecture of the OSI model is based on structured layers, viewing the network as a logical construct of seven separate layers as follows [2,13,22]: (1) physical layer, (2) data link layer, (3) network layer, (4) transport layer, (5) session layer, (6) presentation layer, and (7) application layer. The lower four layers (from the physical to the

transport layer) are charged with communications-oriented functions, or those functions and rules concerned with the use of the physical data transmission network [15]. The upper three layers (from the session to the application layer) are associated with processing-oriented functions, such as the exchange of information between similar OSI models. The rules for processing-oriented functions are referred to as high-level protocols. Conversely, the rules of the lower four layers are referred to as low-level protocols.

2.3.2 Seven Layers

1) Physical Layer. The physical layer provides electrical, mechanical, functional, and procedural characteristics for the initiation, maintenance, and release of physical connections.
2) Data Link Layer. The data link layer provides the functional and procedural means to initiate, maintain and release data links between network layers, i.e., the layer defines the rules for sharing the use of the physical layer among systems connected to the network. The data link layer ensures reliable data transfer through the physical link by providing the following functions: (a) frame synchronization and (b) error handling for transmitted data.

3) Network Layer. The network layer provides the means to exchange network service data between connected network systems. It performs buffering and routing operations for data packets to establish logical connections between systems.

4) Transport Layer. The transport layer provides a universal transport service independent of the physical medium in use, initiating the partitioning of messages sent from upper layers into data packets and reassembling into messages the data packets received from lower layers. This layer provides error detection and assures the quality of data transfers.

5) Session Layer. The session layer provides services as follows: (a) establish logical communication paths between two user processes running on two separate network nodes and (b) control of data exchange, delimitation, and synchronization.

6) Presentation Layer. The presentation layer provides machine-independent characteristics for higher and lower layers by converting messages into forms suitable to both parties. Moreover, presentation layer message conversion provides data compression and security.

7) Application Layer. The application layer serves as the medium between the user and the

OSI network, providing distributed information service for an application and its management and system management, such as activation, maintenance, and termination of distributed resources throughout the OSI network [2].

2.4 Ethernet and IEEE 802.3 Standards

Ethernet is a standard for LANs which is supported by Intel, Digital Equipment, and Xerox Corporations. Ethernet was experimentally developed in 1975 at the Xerox Palo Alto Research Center (PARC) [10].

IEEE Standards documents are developed within the technical committees of the IEEE societies and the Standards Coordinating Committees of the IEEE Standards Board. The IEEE 802.3 standard is part of the LAN family of standards for LANs, shown in Figure 2.2. The physical layer specifies medium access schemes, such as CSMA/CD for IEEE 802.3, token-bus for IEEE 802.4, and token-ring for IEEE 802.4 Standards. The standard IEEE P802.1 describes the relationship among these standards and their relationship to the OSI reference model.

2.4.1 Overview

The Ethernet LAN is intended for such purposes as office automation, distributed data processing, terminal access, and other situations which require inexpensive connections to local communications medium

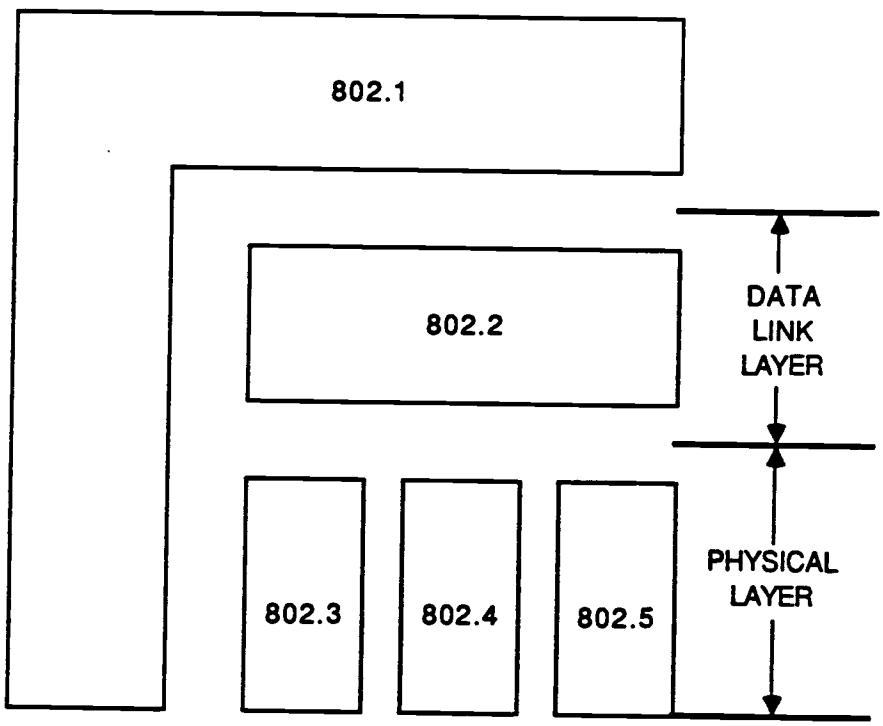


Figure 2.2 Family of IEEE LAN Standards.

carrying bursts of traffic at high peak data rates [12]. Ethernet operates at a data rate of 10 Million bits/sec (Mbps) over 50-ohm, shielded, coaxial cable. The network uses the CSMA/CD media access scheme on bus topology with a maximum number of stations of 1,024. Its primary characteristics include a physical and data link layer [12].

The IEEE 802.3 Standard describes a CSMA/CD scheme and bus topology, which also operates at 10 Mbps over 50-ohm coaxial cable. However, the IEEE 802.3 Standard is intended to describe CSMA/CD bus LANs operating at speeds between 1 Mbps and 20 Mbps over various media types [7].

2.4.2 Network Architecture

Figure 2.3 shows a comparison of layers between Ethernet and the IEEE 802.3 Standard with respect to the layered architecture of the OSI reference model. The Ethernet data link layer corresponds to the IEEE 802.3 Medium Access Control (MAC) and the IEEE 802.2 Logical Link Control (LLC).

2.4.3 Frame Formats

Though Ethernet and the IEEE standard share almost the same network architectures, their frame formats are different. Figure 2.4 illustrates the format of an Ethernet data link layer frame. Descriptions of each field are as follows.

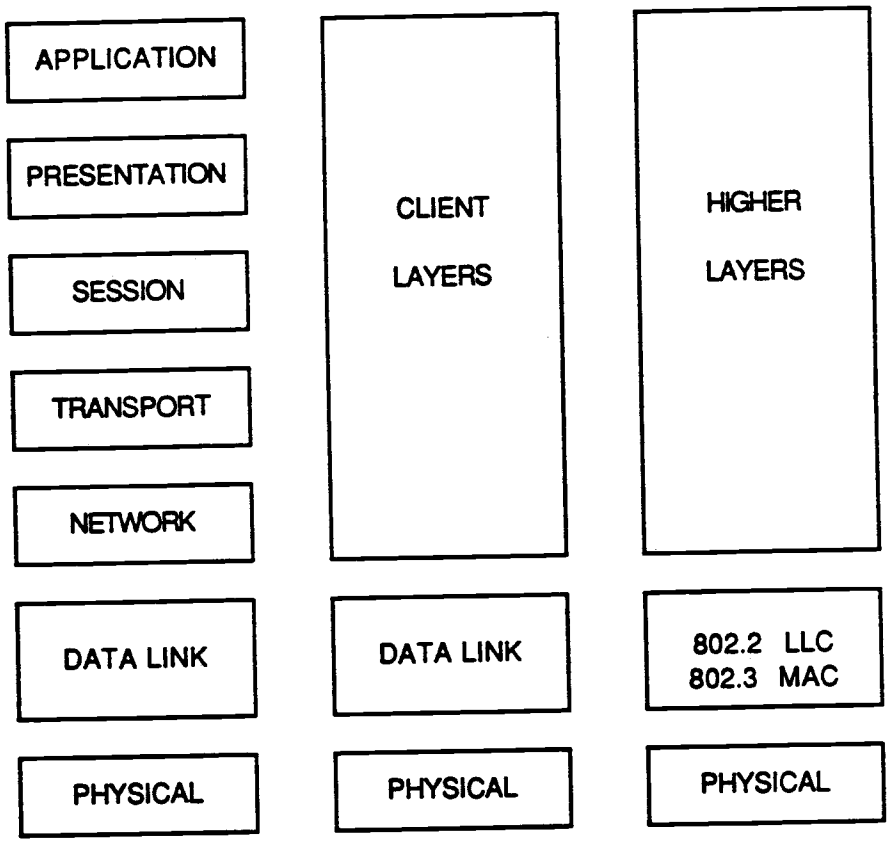


Figure 2.3 Layered Architecture of Ethernet and IEEE 802.3 Standards.

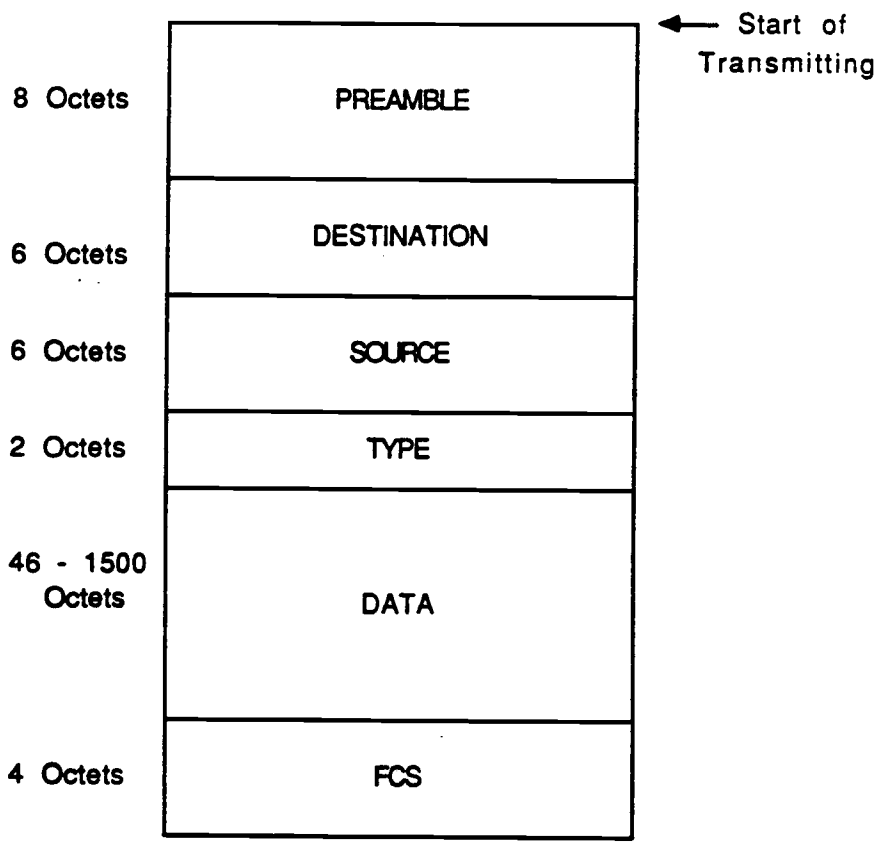


Figure 2.4 Ethernet Data Link Layer Frame Format.

2.4.3.1 Ethernet Frame Format

1) Preamble. This field is for synchronization and framing;
2) Destination Address. The destination address contains the address of the station which will receive the frame;
3) Source Address. This field is the address of station sending the frame;
4) Type. The type field is reserved for use by higher levels to identify the client layer protocol associated with the frame;
5) Data. The data field contains a sequence of 46 to 1500 octets; and
6) Frame Check Sequence. This field contains the Cyclic Redundancy Check (CRC) value, computed as a function of the contents of the source, destination, type and data fields.

2.4.3.2 IEEE 802.3 Standard Frame Format (Figure 2.5). Fields in the format include:

1) Preamble. This field is for the synchronization with bit pattern 10101010 for each octet;
2) Start Frame Delimiter (SFD). This field indicates the start of the frame indicator. The preamble and the SFD field are the same as the preamble field of Ethernet;
3) Destination Address. This field contains the address of the station to receive the frame,

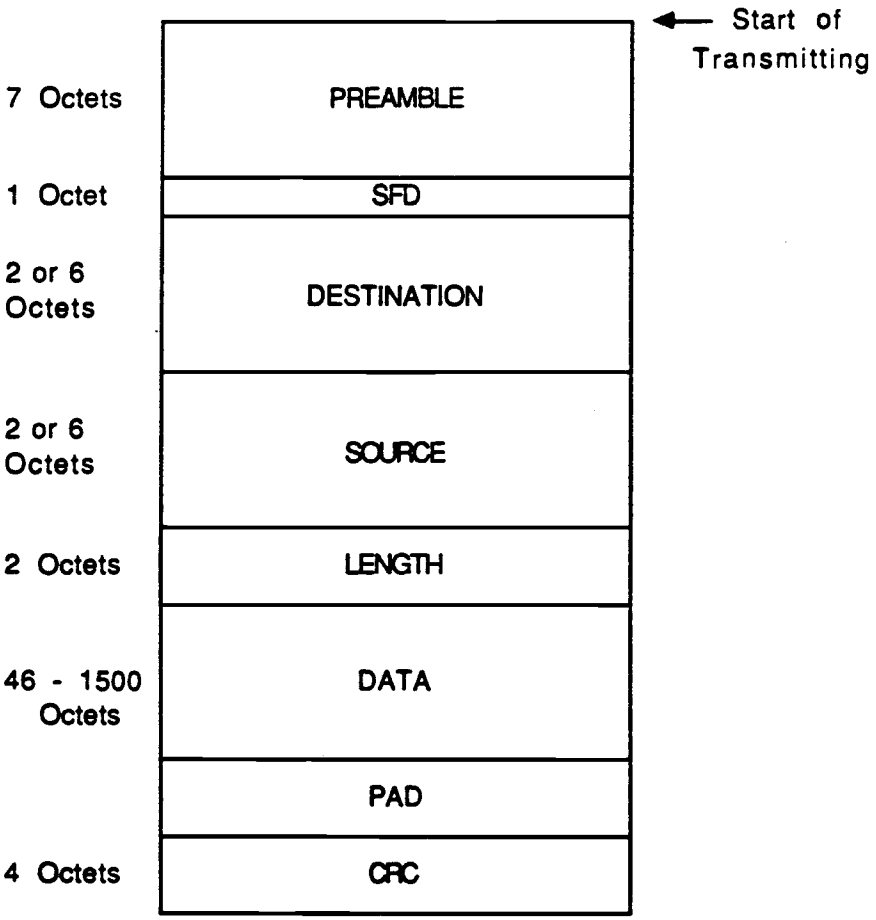


Figure 2.5 IEEE 802.3 Frame Format.

using two address formats, either of 2 or 6 octets, with different forms [7];

4) Source Address. This field contains the address of the station sending this frame. The type 10BASE5 (Ethernet version) specifies the use of 6-octets;
5) Length (LEN). The LEN field specifies the length of the data field;
6) Data and PAD. The length of both these fields depends upon the maximum and minimum frame size and the length of the address field. The type 10BASE5 version specifies that these two fields will be between 46 and 1,500 octets in length [7]; and
7) CRC. This field uses the CRC-32 polynomial.

2.4.4 Ethernet Implementation

Various channel media, such as twisted pair, shielded coaxial cable, radio, fiber optics, and diffuse infrared, may be used for Ethernet connections. A typical Ethernet system using coaxial cable is shown in Figure 2.6 [14]. The controller board discussed in this study is in control of the physical and data link layer in the network architecture. Functions of the controller include encoding and decoding, serial-to-parallel conversion, address recognition, error detection, buffering, CSMA/CD channel access management, and

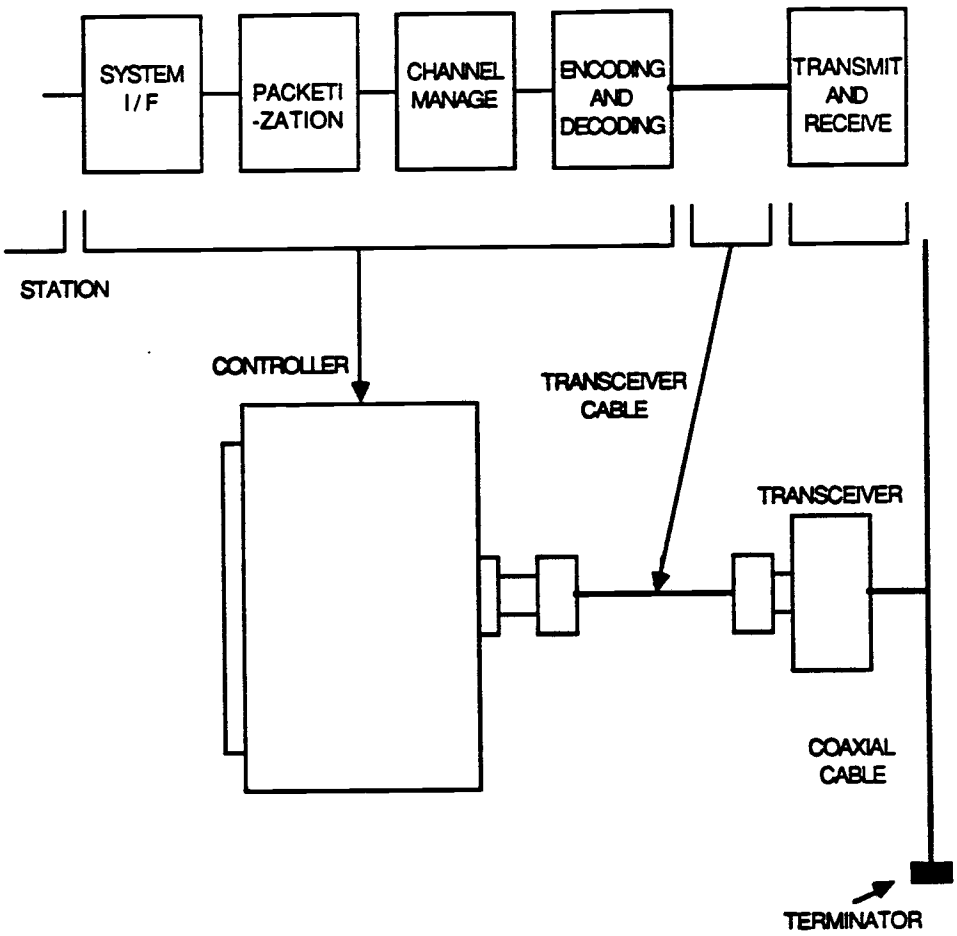


Figure 2.6 General Ethernet Implementation [14].

packetization. The transceiver contains electronic circuits to transmit and receive signals over the coaxial cable of a maximum length of 50 meters, recognizing carriers on the channel. A network configuration, which illustrates the topology of the branching non-rooted tree, as detailed in the Ethernet specification, is shown in Figure 2.7.

2.5 The Macintosh II System

This section provides a basic description of the Macintosh II (Mac II) system, one of the most powerful and recently developed systems of the Macintosh family.

2.5.1 Architecture

The Mac II computer system consists of a Motorola MC-68020 microprocessor with a 15.6672 MHz clock, an optional memory management processor, memory, ROM, six NuBus expansion slots, and the Apple DeskTop Bus (ADB) for connecting the keyboard and mouse [9,20]. A block diagram is shown in Figure 2.8. The system encompasses installation of peripherals, such as a hard disk, its own internal 3.5 inch floppy disk drives, a modem, and a laser printer.

2.5.2 NuBus Interface

The NuBus interface, shown in Figure 2.8, is a bidirectional, 32-bit wide, multiplexed address/data

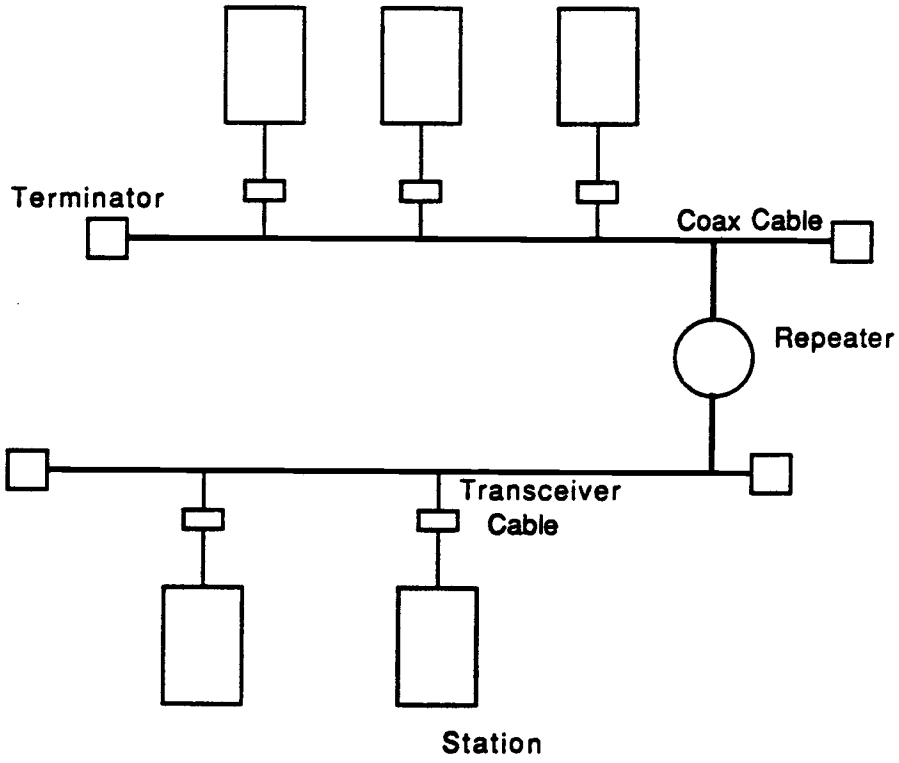


Figure 2.7 Medium-Scale Ethernet LAN Network Configuration.

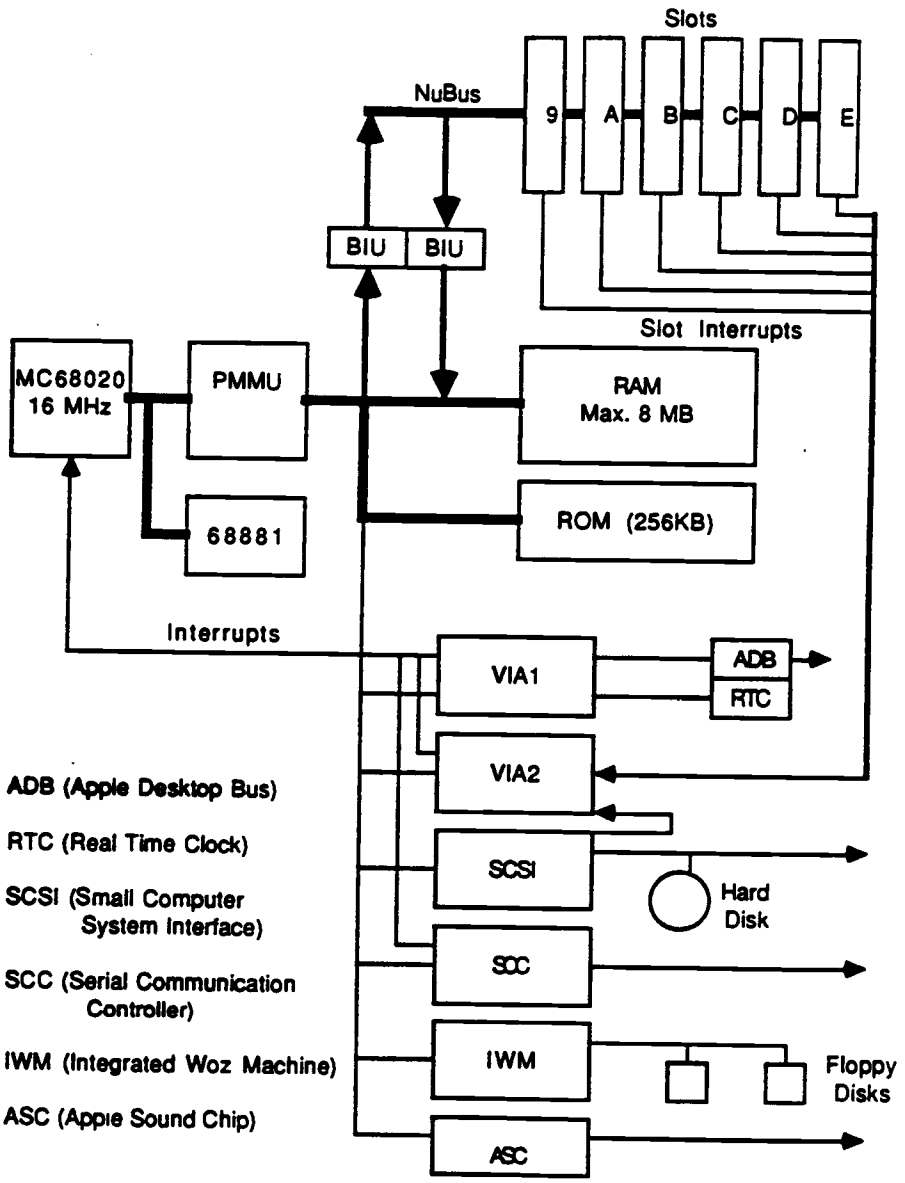


Figure 2.8 Macintosh II Hardware Architecture [9].

bus. Its design objectives are to obtain an independent system architecture regardless of the microprocessor in use, high data transfer speeds, simplicity of protocol, ease of system configuration, and small pin count. The NuBus is a synchronous bus, with all transactions and signal samplings synchronized to a central system clock at 10 MHz [4].

The NuBus uses only read and write operations in a transaction. Interrupts are detected through an /NMRQ line for every slot. The rights of any board placed in NuBus slots are even, i.e., no board or slot position is the master or a special slot [4]. NuBus signals are grouped into seven classes according to their functions: utility, control, address/data, arbitration, parity, slot identification, and power/ground.

2.5.3 Macintosh II Address Allocations

The NuBus is accessible to all of the existing Mac II address space. The Mac II has two options to allocate addresses in accordance with the operating systems in use: 24-bit and 32-bit addresses. The typical Macintosh operating system uses 24-bit addresses mode, while recent Apple/Unix operating systems use 32-bit addresses. A NuBus to Mac II address map is shown in Table 2.2.

Table 2.2 NuBus to Mac II Address Mapping.

24-Bit Addresses from MC68020	24-Bit Addresses from MC68020	NuBus Addresses	Description
$xx00 0000 - $xx7F FFFF	$xx00 0000 - $007F FFFF	$xx00 0000 - $007F FFFF	Present RAM
	$xx80 0000 - $3FFF FFFF	$xx80 0000 - $3FFF FFFF	Future RAM
$xx80 0000 - $xx8F FFFF	$4000 0000 - $4FFF FFFF	$F080 0000 - $F0FF FFFF	ROM
$xxF0 0000 - $xxFF FFFF	$5000 0000 - $5FFF FFFF	$F000 0000 - $F070 FFFF	I/O
	$6000 0000 - $8FFF FFFF	$6000 0000 - $8FFF FFFF	Unused
	$9000 0000 - $EFFF FFFF	$9000 0000 - $EFFF FFFF	Super Slot
$xxF0 0000 - $xxFF FFFF	$F000 0000 - $F0FF FFFF	$F000 0000 - $F0FF FFFF	Slot $0
	$F100 0000 - $F8FF FFFF	$F100 0000 - $F8FF FFFF	Unused
$xxS0 0000 - $xxSF FFFF	$FS00 0000 - $FSFF FFFF	$FS00 0000 - $FSFF FFFF	Slot Space S is in the Range $9-$E
	OR	OR	
	$FS10 0000 - $FSFF FFFF	$FS10 0000 - $FSFF FFFF	
	$FF00 0000 - $FFFF FFFF	$FF00 0000 - $FFFF FFFF	Unused

2.5.4 NuBus Byte Structure

The NuBus bit structure is different from the MC-68020 microprocessor bus, therefore the Mac II performs byte swapping of data between the Nubus and the MC68020. A diagram for byte lane mapping is shown in Figure 2.9. The routes by which bytes are transferred between the NuBus and the MC-68020 microprocessor are called byte lanes [4]. The order of bits within a byte lane is not changeable. The Mac II system automatically performs the byte swapping as shown in Figure 2.9.

2.6 Intel 82586 LAN Coprocessor

The 82586 is an intelligent local area network coprocessor using the CSMA/CD medium access scheme.

2.6.1 Features

The 82586 supports IEEE 802.3 type 10BASE5/Ethernet and IEEE 802.3 type 10BASE2/Cheapernet specifications. The coprocessor manages the entire process of receiving and transmitting frames. Hence, it relieves the host system from managing communication transactions.

The 82586 has four on-chip Direct Memory Access (DMA) channels for data transfers, status, and commands

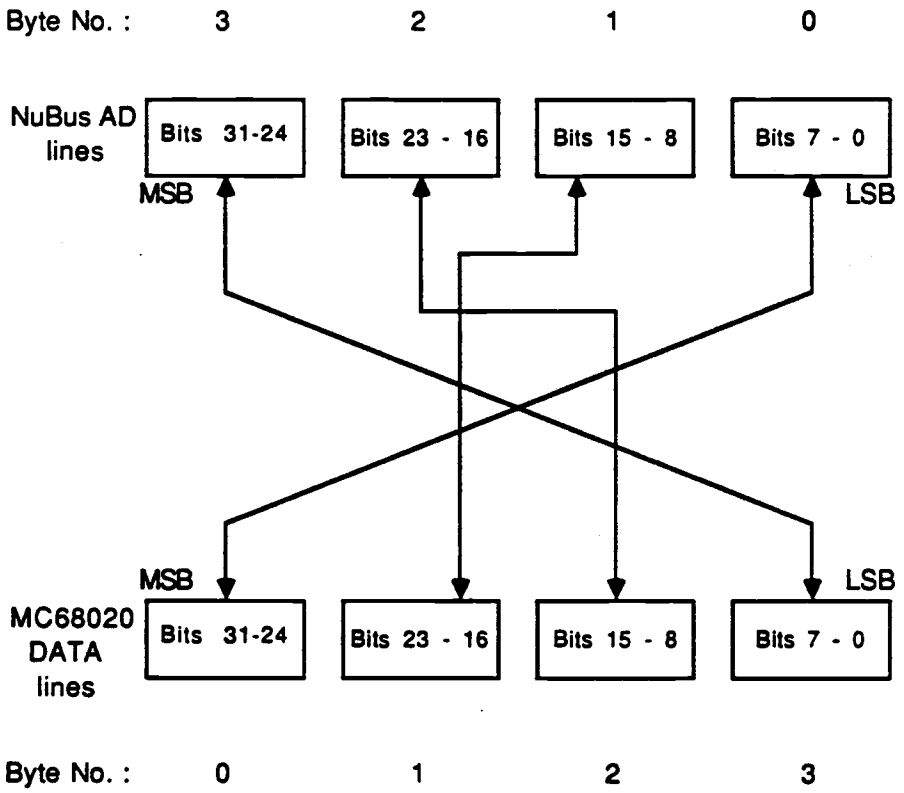


Figure 2.9 Diagram of Byte Lane Mapping

and may be used for non-Ethernet applications. The 82586 includes two independent 16-byte First-In, First-Out (FIFOs).

2.6.2 Operations

Communication between the 82586 and the host system is performed through the shared memory, which contains the data structures for transmitting and receiving operations. After power-up, the host system sets up data structures for action commands, transmission and receiving operations, then initializes the 82586 to initiate further transactions. The 82586 reads data in specific locations of the shared memory and automatically initiates itself, performing transmission and receiving transactions independently. The 82586 is composed of two independent units, the Command Unit (CU) and the Receive Unit (RU). The CU obtains and executes commands sent from the host system, while the RU stores received frames in the shared memory. The transmitting operation is provided by the CU. The only hardware signals used between the host system and the coprocessor are INTerrupts (INT) and Channel Attention (CA) signals. Interrupts are asserted by the 82586 for the attention of the host system and CA is generated by the host system to notify of initiation of a command. A more detailed discussion follows in Chapter 4.

CHAPTER 3: MACINTOSH II ETHERNET INTERFACE CARD

3.1 Introduction

The Mac II Ethernet interface card provides direct connectivity for Ethernet-based networks running on the Apple Macintosh II computer system. The principal object of this board is to use shared memory for communications between the host system and the local coprocessor. This chapter provides an overview of the Ethernet Mac II board, including the basic concepts of its purpose and design. In addition, layout considerations for Federal Communications Commission (FCC) testing is described.

3.2 Overview

The Ethernet interface card is compatible with the Mac II NuBus and requires a full-size slot for installation. It utilizes Intel's Ethernet/Cheapernet chipset, consisting of the 82586 LAN Coprocessor, the 82C501 Ethernet Serial Interface (ESI) and the 82502 Ethernet Transceiver Chip (ETC). The block diagram shown in Figure 3.1 illustrates the architecture of the interface card. The 82586 LAN Coprocessor, which acts

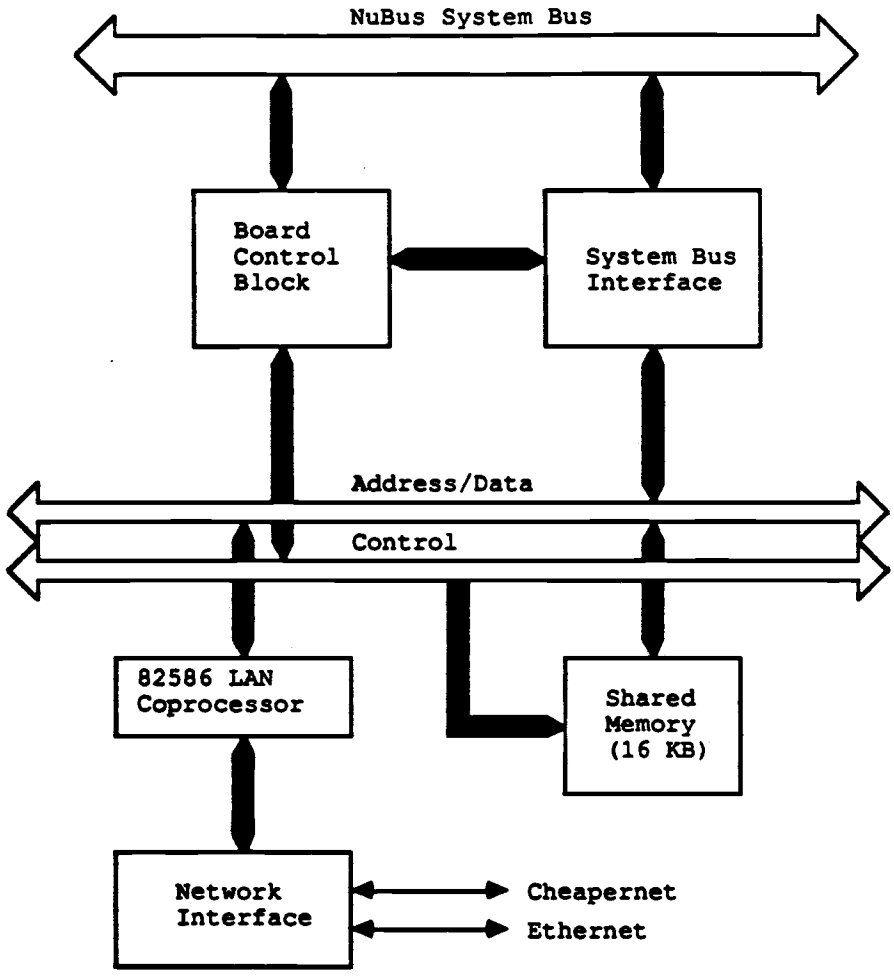


Figure 3.1 Block Diagram of the Interface Card.

as a master, always has priority on the local bus, i.e., when the host system tries to access local shared memory simultaneously with the 82586, the host system must wait until the completion of the coprocessor access.

The board control block contains six Programmable Array Logics (PALs) to provide the NuBus interface and the bus arbitration and hardware command registers. The host system's processor is relatively faster than the LAN coprocessor on the interface board. Therefore, careful consideration in the design of the control block has been required in order to maintain system performance with the least loss of time.

The 16 Kbytes of shared memory is partitioned into receive and transmit sectors and data transfer between the host system and the interface board is based on single-cycle transactions. The interface board uses a NuBus system interrupt line (/NMRQ) to provide the handshaking mode between the host system and the board.

All incoming and outgoing message packets are assembled or disassembled in the network interface. For this purpose, the 82C501 ESI chip works directly with the coprocessor for IEEE 802.3 LAN applications and the 82502 ETC provides a direct interface to the coaxial cable. Major functions include transmission, reception, and collision detection.

3.3 System Bus Interface Logic Design

The Mac II system has six expansion slots in its backplane, each of which has an unique slot number wired into its slot ID pins. Therefore, each card can be addressed by its geographical position. The top 256 Mbytes of the system address range is assigned to accommodate 16 slots, i.e., 16 Mbytes space for each slot (see Figure 3.2). (Since there are only 6 expansion slots, addressed as 9 to E in hexidecimal, and not 16 as provided for by the system, the balance of the memory is not utilized in the existing system.)

3.3.1 Memory Mapping

The shared memory, consisting of two static RAMs (8 K by 8 bits each) to provide the data structure for the 82586 coprocessor and data buffers for receiving and transmitting frames, is accessible to both the host system and the 82586 coprocessor. This memory is located between the addresses 0000 and 3FFF for 82586 local access and between addresses Fss00000 and Fss03FFF for system bus access. Figure 3.3 illustrates address mapping for the shared memory, access to which is available as either 8-bit or 16-bit data transfers. However, 16-bit access is preferable in order to reduce the cycle time of data transfer transactions.

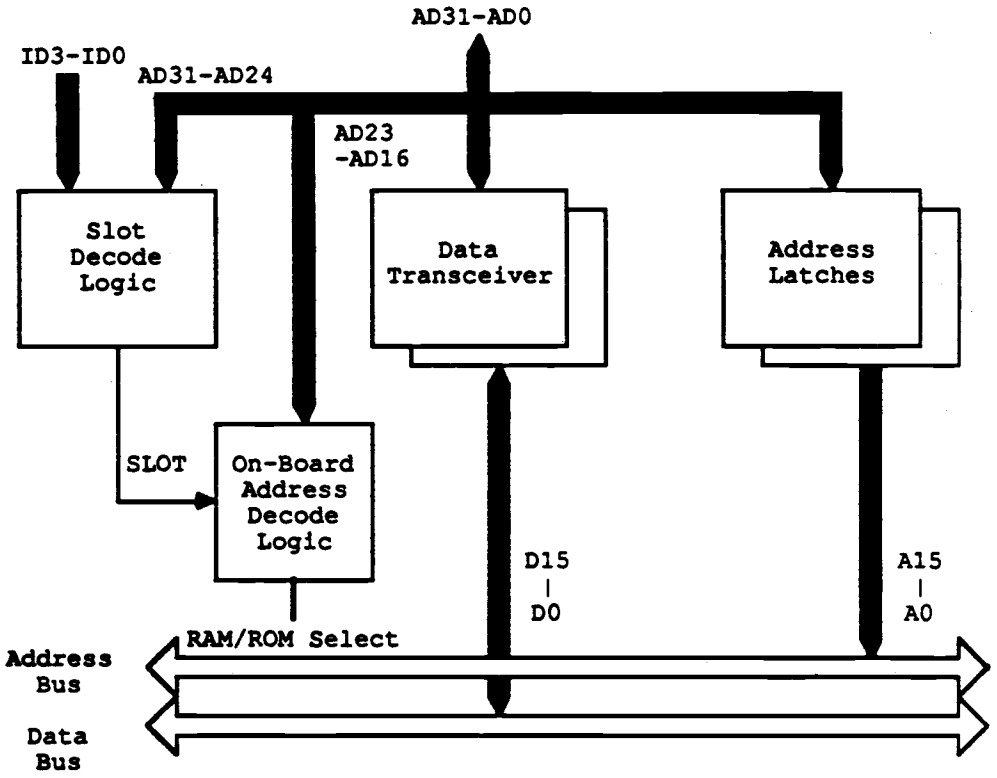


Figure 3.2 System Bus Interface Block.

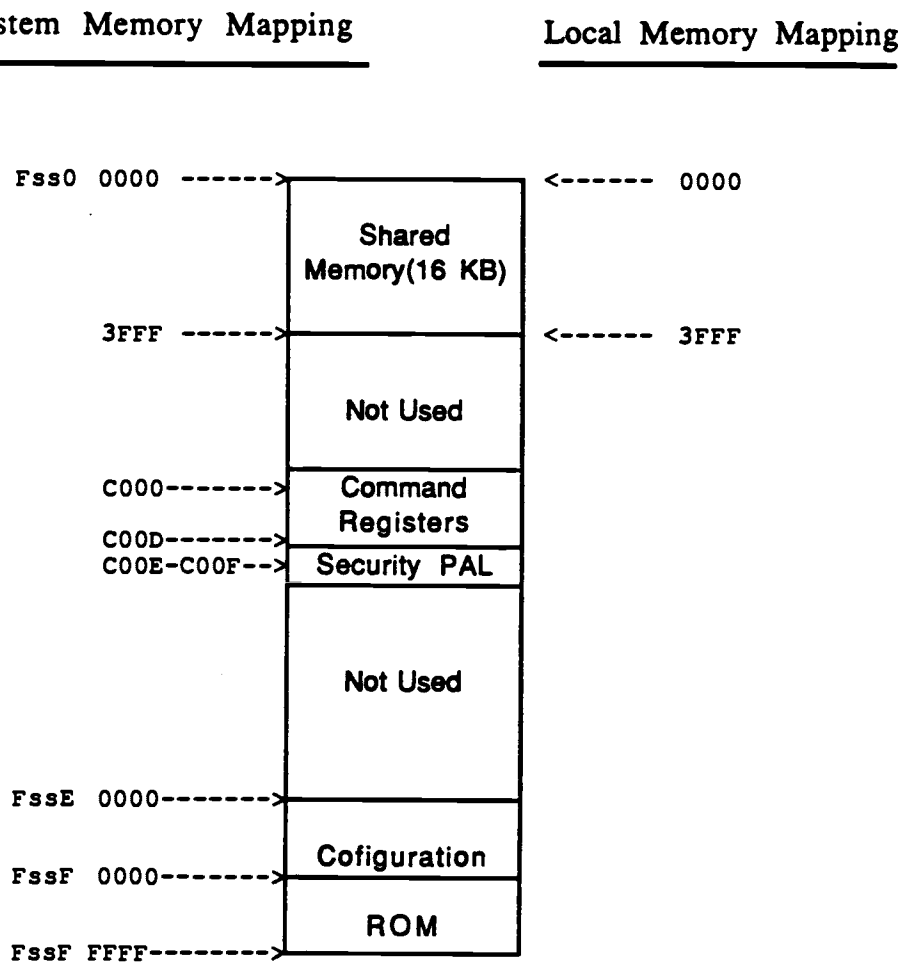


Figure 3.3 Shared Memory Address Mapping.

Therefore, the 82586 is programmed to the 16-bit access mode with even address boundaries. On the other hand, the host system may access the shared memory in either the 8-bit or 16-bit modes to maintain compatibility with higher level communication protocol layers since these layers require byte access to shared memory. The 32-bit data (long word) access from the host system is broken into two 16-bit data transfers due to the data line limits of shared memory.

3.3.2 Configuration ROM

The configuration ROM, called declaration ROM, is implemented in 8-bit physical width sectors on the card and its address space is assigned to FssE0000 through FssFFFFF (with a 32 Kbyte address range). The ROM, which provides the data structure, including a format block, a slot resource directory, a board slot resource list, and other slot resource lists, is always located at the top of the slot space. The format block bears a hexadecimal value of 78, providing information that the host system is using the fourth byte lane to communicate with card configuration ROM. The slot resource directory lists all the slot resource lists, providing offsets for each resource in the card firmware. Each slot resource list, in turn, contains references bearing information about a single resource, including vendor name, a card function defined as the networking

board, and the name of the card and driver used by the operating system (defined as "sMacOS68020").

3.3.3 Address/Data Bus

The address/data bus communicates with multiplexed, bidirectional interface signals and the interface card uses both 32-bit address and 16-bit data lines. As shown in Figure 3.2, the slot decode logic interprets any physical address whose upper four address bits (AD31-AD28) are in the top 256 Mbyte address space. The next four address bits (AD27-AD24) are decoded to select each slot. The address lines, AD23 to AD16, are used to divide the address space of the card into two sections, one for configuration ROM, in which the properties of the card and usable information for the operating system are defined, and the other for the hardware command register and shared memory address space. The hardware register is described in greater detail in Chapter 4.

The data lines (AD31-AD0) are divided into two sections (D15-D0): Address/data lines AD31-AD24 and AD15-AD8 correspond to data lines D7-D0 and address/data lines AD23-AD16 and AD7-AD0 to data lines D15-D8. While the NuBus has a 32-bit address/data bus, the card has a 16-bit data bus. Therefore, the system

bus interface control block maps the system address/data lines to the on-board data lines as shown in Table 3.1.

Table 3.1 Address and Data Bus Correspondence.

NuBus Address/Data Lines	On-Board Data Lines
AD31-AD24 and AD15-AD8	D7-D0
AD23-AD16 and AD7-AD0	D15-D8

3.3.4 NuBus Interface Signals

The signals used for the board are described in this section. Eleven signals, for all functions except the address/data bus and the power lines, are used for the board.

3.3.4.1 Utility Signals

1) /RESET. The reset signal is asserted asynchronously to the NuBus clock. This signal is combined with the software reset signal to generate the internal 82586 coprocessor reset signal.

2) /CLK. The clock signal is driven from the host system and is used to synchronize data transfers between system cards. The signal has a constant nominal frequency of 10 MHz, with an

asymmetric duty-cycle of 75% high and 25% low; it is also used for arbitration signals.

3) /NMRQ. The non-master request signal is asynchronous to /CLK, providing an interrupt driven scheme. The 82586 coprocessor on the board sends its interrupt signal to the host system CPU through this line, asserting the interrupt signal whenever it needs the system CPU to report its status, i.e., after each command from the host system.

3.3.4.2 Control Signals

1) /START. The transfer start signal is driven for only one clock period by the system at the beginning of a transaction. This signal enables the comparator of the slot decode logic (see Figure 3-2) to generate a slot select signal.

2) /ACK. The transfer acknowledge signal is driven for one clock period by the card to indicate completion of a transaction.

3) /TM0 and /TM1. These transfer mode signals are driven for one clock cycle by the system at the beginning of each transaction to provide the type of bus operation being initiated. The signals are also driven at the end clock period of each transaction to denote the type of acknowledgement for the transaction.

3.3.4.3 Slot ID: Slot identification signals (/ID0-/ID3) specify the physical location of each card. The upper four address signals (/AD31-/AD28) are used to specify the top 256 Mbyte address space and the next four signals (/AD27-/AD24) are used to compare with the identification signals to produce a slot select signal on the card. The slot select signal is used to initiate any system bus access transaction, generating the first state of the NuBus slave state machine (PAL 16R6) and the address latch signal for the NuBus interface.

3.4 Internal Bus Arbitration Logic Design

The local bus is shared by the 82586 LAN coprocessor and the host system. The arbitration logic consists of two states, one for the local bus and the other for the system bus.

3.4.1 The 82586 to Memory Interface

Since the 82586 HOLD signal is connected directly to its HLDA, i.e., whenever the 82586 needs access the to shared memory, the 82586 is allowed access without restrictions. The 82586 can be programmed to 8-bit or 16-bit data access mode and the 16-bit access mode is used for interface card data transfers. Therefore, the 82586 is connected to the shared memory via a 16-bit data path through two 8-bit data transceivers. The address lines A1-A13 of the 82586 are connected to the

static RAMs through the address latches, with tri-state output. The address A0 is neglected due to use of the 16-bit data mode. Thus, the accessible address space is 16 Kbytes.

The 82586 coprocessor operates at 6 MHz with clock periods of 167 nano-seconds (ns) and each 82586 memory cycle consists of four clock periods, T1, T2, T3 and T4. The static RAM used on the board requires 100 ns access time. Therefore, the actual clock periods to access shared memory are T3 and T4. Considerations for accessing memory are discussed in the following section.

3.4.2 The Local State Machine

Since the HLDA and READY are always active, the 82586 is always available to access memory. The control logic requires a state machine for the read-or-write-cycle timing. See Figure 3.4 for the timing diagram to access memory. The ALE (Address Latch Enable) signal is provided by the 82586 to latch the address onto the octal D-type latch and to initiate the LQ0 signal in the state machine. In turn, the LQ0 is initiated prior to the end of ALE, i.e., it is started in the middle of the 82586 T1 clock, remaining active during 210 ns and restarting at the beginning of the T4 clock period to finalize the read- or write-cycle.

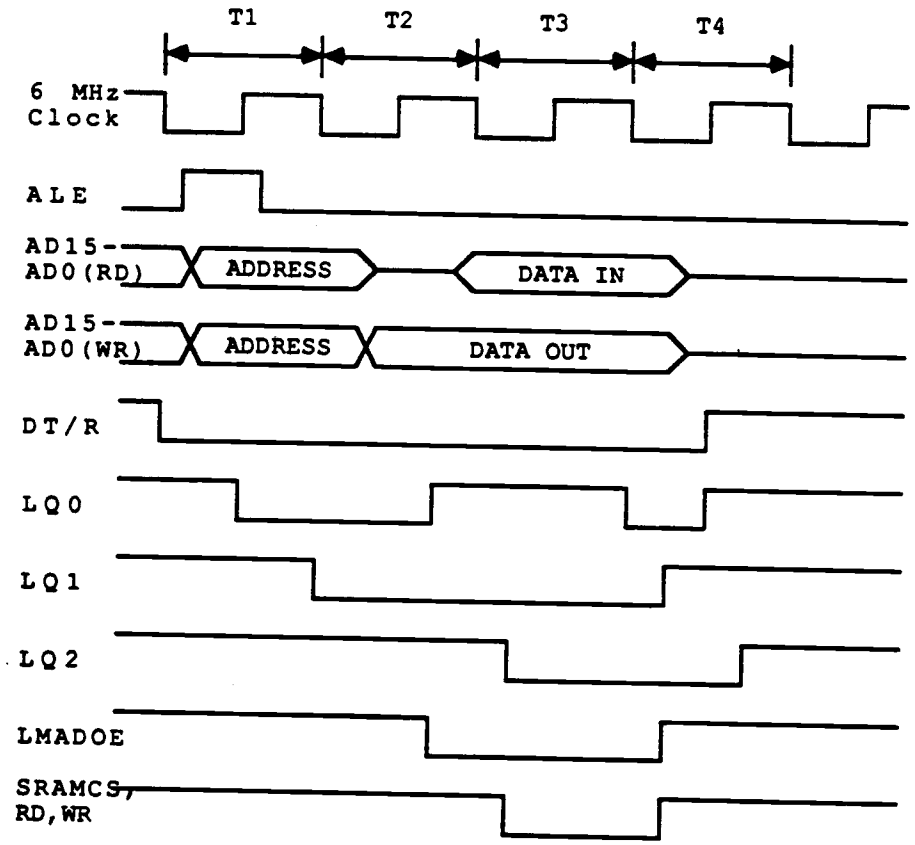


Figure 3.4 Timing Diagram of the 82586 Read, Write, and Local State Machine.

The LQ1 signal is initiated at the beginning of the T2 clock to generate the arbitration signal between the local state machine and the system state machine. The pulse period, from the beginning of LQ1 to the generation of the address latch and the data transceiver output enable control signal (LMADOE) for the 82586, is 125.25 ns. This is because the system clock period for the Mac II system is 100 ns and the arbitration time to avoid the contention between the system and the 82586 requires at least 100 ns (a more detailed discussion is provided in the following section).

The end of LQ1 active indicates when the LMADOE signal should be disabled. At this point the LQ2 signal specifies the beginning time for the static RAM chip select signal (SRAMCS0 and SRAMCS1), its trailing edge denoting the end of a read or write transaction cycle. The SRAMCS0 and SRAMCS1 signals remain active until the second leading edge of LQ0 is passed. The approximate period for the SRAMCS is 125 ns, which is sufficient to access the static RAM.

3.4.3 The System State Machine

The state machine for the system has three state signals, MQ0, MQ1, and MQ2. Each is synchronized by the 10 MHz system clock. As shown in Figure 3.5, the triggering signal is a slot select signal, SLOT. The address lines, START, and the TM0 and TM1 signals are

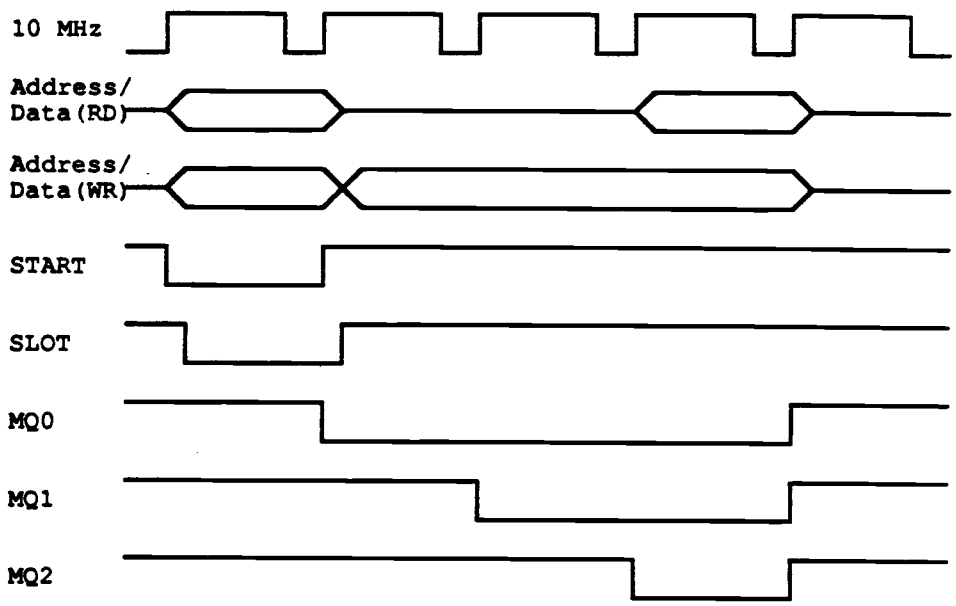


Figure 3.5 NuBus Address/Data Bus and System State Machine.

provided during the first clock period of each transaction by the host system via the NuBus. (System interface logic, as discussed in the section 3.3, generates several control signals.) Host system access to shared memory is interleaved between 82586 accesses. The first state signal, MQ0, is initiated following the first clock period of a transaction cycle, remaining active low until the end of the cycle. If the 82586 is not using shared memory during the second clock period, the second state signal, MQ1, is automatically generated in the system state machine (16R6 PAL). The third state signal, MQ2, is also provided in the following clock period if the 82586 does not initiate memory access and the actual memory access should be completed during the MQ2 active phase.

The interface card sends the ACK and the TM0/TM1 signals to the host system via the NuBus. In normal communications, each interface card transaction must be completed in 400 ns. However, if simultaneous transactions between the host system and the 82586 have occurred, memory access time from the host system should be prolonged to avoid contention. A detailed case study is described in section 3.4.4.

3.4.4 Arbitration Control Logic

The actual signal to arbitrate between the host system and the 82586 is the LQ1 in the local state ma-

chine PAL and the period for effective arbitration is 125 ns (see section 3.4.2). This signal is synchronized by the inverted system clock (10 MHz), generating a new signal and a hold host system (HOLDS) signal. The LQ1 signal is then triggered by the rising edge of the inverted clock in a D flip-flop. Therefore, the host system state machine of the interface card confirms whether the 82586 has tried to access shared memory. The marginal time for decoding the status is 25 ns and the actual spending time is a maximum of 9.2 ns in the fast speed D flip-flop. Thus, the state signals from the system state machine PAL has sufficient marginal time and in the worst case the itemized timing period required to decode the status is 100 ns for detecting the LQ1 signal, 9.2 ns for the transparent time of the D flip-flop, and a maximum of 6 ns for inverting the system clock. The total time is therefore 115.2 ns and the marginal time for safe operation is 9.8 ns in the worst case (i.e., 125 ns - 115.2 ns). Timing considerations for this design were proven in a real time situation using the Hewlett-Packard Model 1431D Logic Analyzer.

There are four cases which may occur in the arbitration control logic. The first is that shared memory is already being used by the 82586 before the assertion of MQ0. In this case none of system state signals are generated during the period of access of the 82586.

The second case is the initiation of LQ1 during the first MQ0 clock period, which causes a delay in the generation of MQ1 until completion of the 82586 transaction. This means MQ2 is unavailable until the completion of the 82586 transaction. The LQ1 signal may be started only during MQ1, which is the third case of contention arbitration. In this case MQ1 is inactivated until the expiration of the LQ1 signal. Thus, the inactive state of MQ1 prevents MQ2 from going to active state, thereby preventing or delaying the host system from accessing local memory. On the other hand, when the LQ1 signal is activated during the MQ2 active state, MQ2 holds the active state in order to access shared memory, which is why in this design consideration 125 ns of marginal time are required between the activation of LQ1 until initiation of the active state of LMADOE. This period is sufficient for the system to access shared memory, with 100 ns access time during MQ2 (which has a 100 ns clock period).

3.5 Hardware Registers

There are four commands, mapped in memory and installed in 16L8 PAL, that can be issued to the interface board by direct software action. A list of hardware command registers is shown in Table 3.2. All data used in the registers are in hexadecimal code.

Table 3.2 List of Hardware Command Registers.

Commands	Address	Data	
		Set	Reset
Loopback Mode	FssOC000	1	0
Channel Attention	FssOC002	0	1
Software Reset	FssC004	1	0
Interrupt Enable	FssC006	1	0

3.5.1 Software Reset

The reset command to the 82586 is activated by writing 0001 as data into the address FssOC004. To de-activate the reset signal the system should write a 0000 into the same address, after a minimum of four 82586 clock cycles.

3.5.2 Loopback Mode

The loopback mode is enabled by writing the data 0000 to the address FssOC000; conversely, writing 0001 to the same address will disable the loopback mode. This feature is used for self-diagnosis of the 82586 network interface, as described in section 3.6.6.

3.5.3 Channel Attention

The channel attention (CA) signal to the 82586 is enabled by writing the data 0001 to the location of FssOC002. The CA operation provides a handshake between the host system and the card. To deactivate the signal the host system writes the data 0000 to the same address. Figure 3.6 provides a diagram of 82586 and host system interaction.

3.5.4 Interrupt Enable

The Mac II system has an interrupt signal (/NMRQ) in a tri-state for each slot. To enable the interrupt signal the board writes the data 0001 to the address of FssOC006; the host system disables the signal by writing the data 0000 to the same address.

3.6 Network Interface

The network interface provides either a Cheapernet or an Ethernet Carrier Sense Multiple Access/Collision Detection (CSMA/CD) scheme, supporting bit encoding and decoding, carrier sense, collision detection, and loopback mode.

3.6.1 Interface Circuit

The interface circuit consists of the Intel 82C501 Ethernet Serial Interface (ESI), the 82502 Ethernet

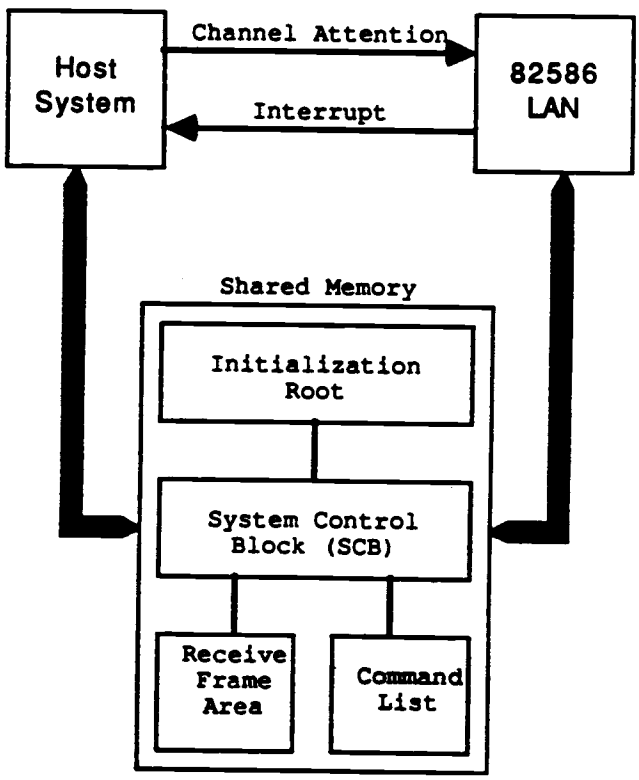


Figure 3.6 82586 and Host System Interaction [11].

Transceiver (ETC), a DC/DC power source, and an isolation transformer. The ESI is compatible with IEEE 802.3 10BASE5 (Ethernet) and 10BASE2 (Cheapernet) specifications, interfacing directly to the 82586 and the ETC. The major functions of the ESI are to perform Manchester encoding/decoding for the transmitted/received frames, to provide an electrical interface for the Media Attachment Unit (MAU) for Ethernet, and to generate the transmit clock for the 82586 coprocessor. The ETC provides the interface with the coaxial cable, and its major functions are message transmission, reception and collision detection. The ETC can drive a transceiver cable up to 50 meters in length, according to Ethernet specifications.

The 82C501 chip provides for Manchester encoding/decoding, diagnostic loopback for network node fault detection and isolation, and generation of the 10MHZ transmit clock for the 82586. The DC/DC power source generates +5V DC and +10V DC for the 82502 ETC, driving the transceiver circuit for Cheapernet. The input voltage of the power source, completely separated from output voltages to provide the isolation of the transceiver circuit, is +12V DC. The ground for the input voltage is a logical (digital) ground. On the other hand, the ground for output voltages is an analog ground.

The isolation transformer provides high isolation voltages to comply with international safety requirements for LANs. The purpose of this chip is to create an isolation buffer between the 82C501 ESI and the 82502 ETC. Figure 3.7 illustrates a block diagram of the network interface.

3.6.2 Encoding and Decoding

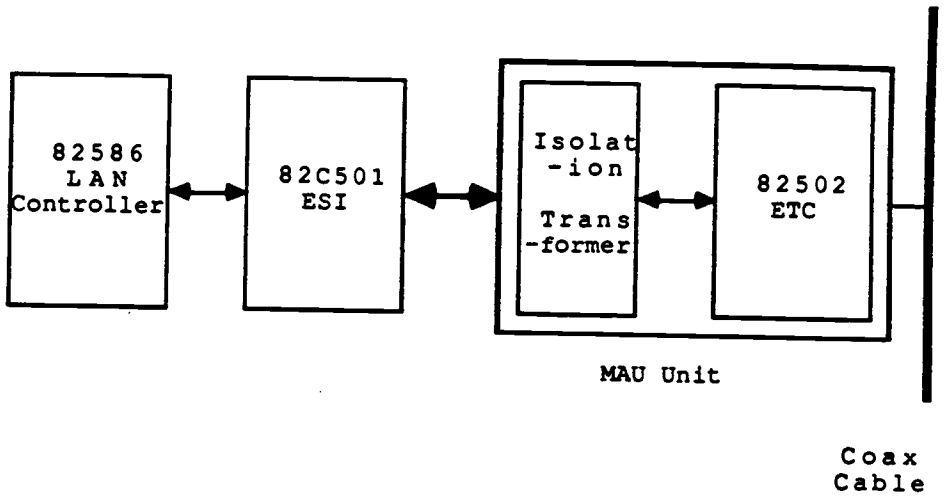
The 82586 receives a transmit clock from the 82C501 ESI, which performs Manchester or NRZ encoding for the transmitted data and decoding for the received data. The configuration is specified in the configuration command block.

3.6.3 Carrier Sense

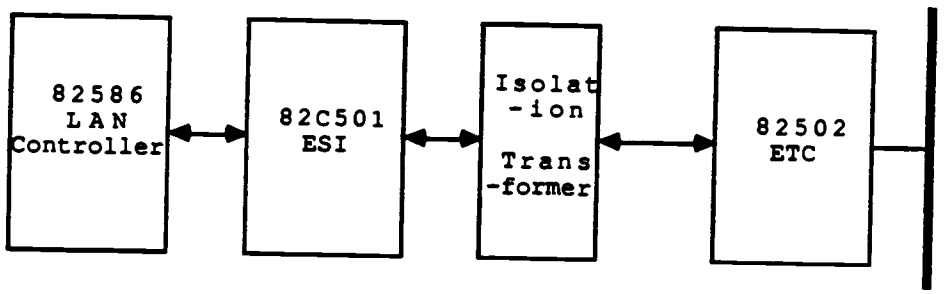
Carrier Sense detects activity on the network. The 82586 can choose the carrier source from either an external (82C501) or an internal generation, however, the IEEE 802.3 Standard is configured for external generation. The 82586 provides two behaviors, transmission and receiving, as follows [11]:

3.6.3.1 Carrier Sense Transmission Procedure

1) The 82586 determines whether Carrier Sense is active and if the network is in use, the 82586 defers;
2) If Carrier Sense goes inactive, the 82586 sets its Interframe Spacing (IFS) timer to 9.6 μs for the IEEE 802.3 Standard;



(1) Ethernet Configuration



(2) Cheapernet Configuration

Figure 3.7 Block Diagram of the Network Interface.

3) When the IFS timer expires, the 82586 initiates its transmission; and

4) If the Carrier Sense signal goes inactive anytime after it transmission of the message preamble, the 82586 aborts transmission.

3.6.3.2 Carrier Sense Receiving Procedure

1) The 82586 starts the IFS timer when the Carrier Sense signal goes inactive; and

2) The 82586 receives the data on the receive clock after the Carrier Sense signal becomes active.

3.6.4 Collision Detection

The collision presence signals are connected to the collision pair of the 82502 transceiver, indicating one of three conditions: (1) there have been simultaneous transmission attempts by two or more stations on the coaxial cable, (2) a Signal Quality Error (SQE) test after the completion of each transmission, which confirms collision circuitry operation, and (3) a frame transmission time which exceeds time limits.

3.6.5 Serial Interface

The handshake between the 82586 and the 82C501 ESI is accomplished by the use of Request-To-Send (RTS) and Clear-To-Send (CTS) signals. RTS notifies the 82C501 that the 82586 has data to be transmitted, while CTS confirms that the 82C501 is ready. On the interface

board the CTS is connected directly to the ground. Therefore, the 82586 deactivates the RTS after each transmission is completed.

3.6.6 Loopback

Three loopback modes, called by setting the respective configuration bits, are used to diagnose network operations: (1) an internal loopback that logically disconnects the 82586 and the ESI unit, directly connecting Transmit Data to Receive Data and Transmit Clock to Receive Clock signals in the 82586; (2) an external loopback that logically connects the 82586 to the 82C501, diagnosing functional operations of the 82C501; and (3) an external transceiver loopback that logically connects the 82586 to the 82C501, the 82502 and the coaxial cable for full link diagnostics.

3.7 Printed Circuit Board Layout

To satisfy the FCC regulations, the following special considerations were provided for board layout.

3.7.1 EMI

Electromagnetic Interference (EMI) is undesirable radiated, conducted, or coupled energy that degrades the reliability or performance of electronic circuitry at specific frequencies. Current transients are the basic cause of these phenomena and the main source of

current transients are integrated circuits. Signals from digital integrated circuits form square waves, generally causing harmonic conflict among components [3]. Therefore, careful consideration in the choice of logic circuits is required. In general, TTL devices cause greater emission problems due to a totem pole output. while CMOS devices have good emission characteristics. To reduce the emissions problem, several factors must be considered: (1) pay careful attention to selection of the logic family, (2) select a grounding plan that reduces inductance, (3) use bypass capacitors, (4) use small inductors, such as ferrite beads, to reduce the number of high frequency components and improve the noise margin, and (5) select multilayer boards to secure a bypassing effect. The printed circuit board used in this project is a four layer board which has obtained the FCC Class A grade. Figure 3.8 shows the horizontal emissions characteristics and Figure 3.9 indicates the vertical emissions characteristics, as obtained from an Hewlett-Packard 8590A Spectrum Analyzer.

3.7.2 Considerations

It is recommended that the card have an analog ground plane for the 82502 and its interface circuit with the coaxial cable. The reference ground is the

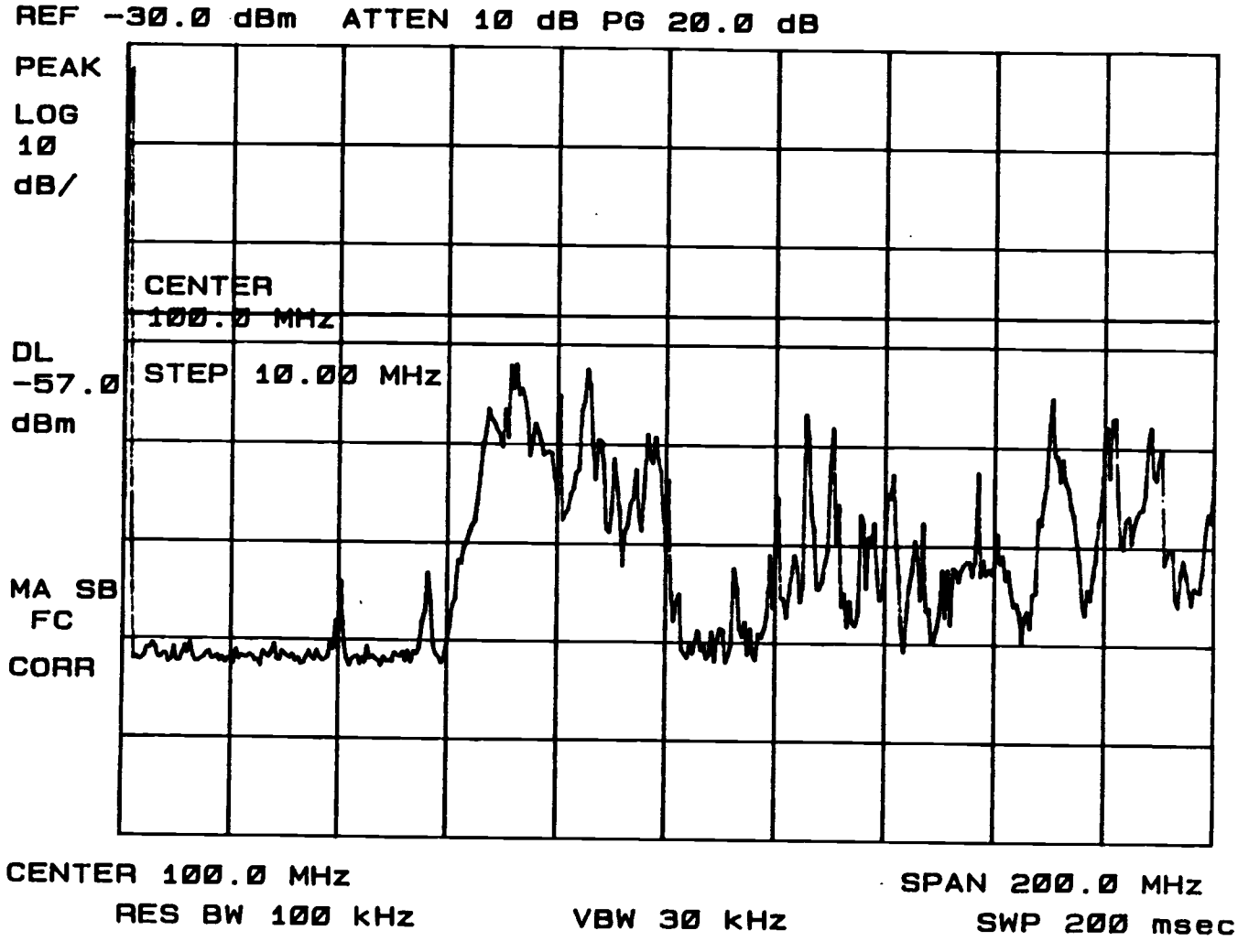


Figure 3.8 FCC Test, Horizontal Emission Characteristics.

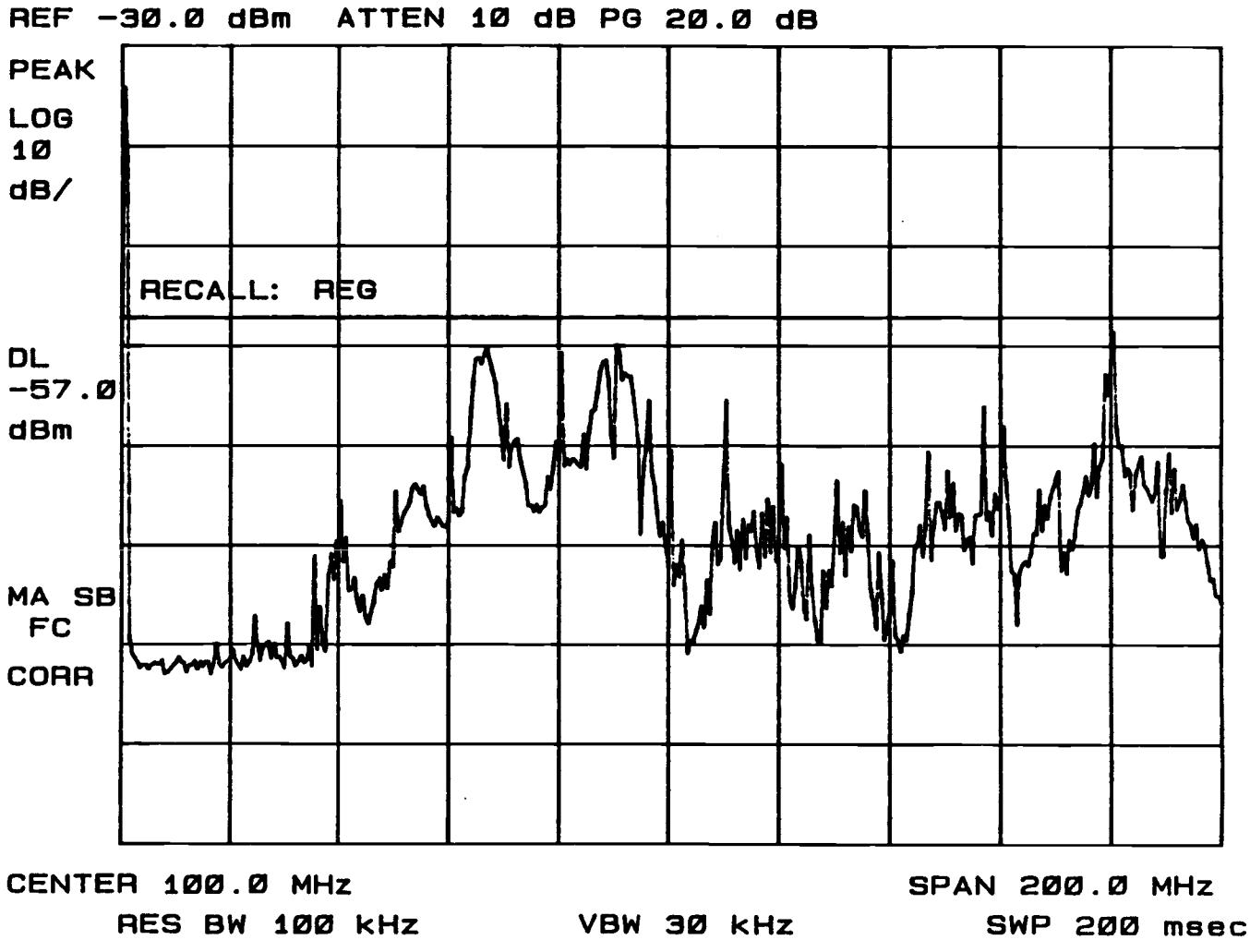


Figure 3.9 FCC Test, Vertical Emission Characteristics.

"0 V" output of the isolated DC/DC power source. The ground Vss and the analog ground AVss of the 82502 is connected to the analog ground plane, which isolates the rest of the card circuitry from the coaxial cable interface in order to assure compatibility with IEEE 802.3 specifications.

High frequency signal lines are not crossed with any other signals and are guarded with ground lines to avoid interference with other signals. The use of components with high currency variations is limited to avoid EMI problems, i.e., limited use of high speed logic devices, such as the 74S or 74AS series. If possible, device card fanouts are limited to a maximum of three or four in number to assure low currency variation. In addition, the path length from the coaxial cable transmit data (CXTD) signal of the 82502 to the center conductor of the cable is minimized for reliable data transfer.

CHAPTER 4: IMPLEMENTING THE INTERFACE BOARD

4.1 Introduction

The 82586 is programmed to serve as the controller for CSMA/CD LAN's. All parameters, including framing parameters, may be configured as desired. This chapter describes the operation and programming of the interface board 82586 processor and is divided into initialization and programming sections.

4.2 Initialization

The board requires three initialization phases, as follows: (1) Memory initialization, (2) 82586 initialization, and (3) self-test diagnostics. This section describes the 82586 initialization phase.

The 82586 requires four steps for normal operations: (1) After power up, the host system resets the board, (2) the 82586 performs the initialization procedure, (3) the host system issues a command, and (4) the 82586 sets interrupts to report the status of each command.

4.2.1 Procedure

The purpose of the initialization procedure is to establish communication between the host system and the interface board 82586 processor. After reset, the 82586 starts the initialization procedure by accessing the System Configuration Pointer (SCP), a data structure (see Figure 4.1) in shared memory, while the host system provides the 10-byte SCP contents prior to the reset operation. The data structure contains a "sysbus" byte and an Intermediate System Configuration Pointer (ISCP) [11]. The "sysbus" byte specifies the data bus width to be accessed by the 82586 through either an 8-bit or 16-bit path. The rest of the block contains the ISCP, which includes the status of initialization and the address of a System Control Block (SCB). The data structure for the ISCP is illustrated in Figure 4.2.

Initialization is initiated by the reset-CA sequence, after which the following occurs:

1) The 82586 reads the hexadecimal location 0FFFFF6, the "sysbus" byte. On the board the actual address is 3FF6 because the internal address range is 0000 to 3FFF (16 KBytes). The upper address bits of the 82586, A16-A20, are not connected to the local address bus.

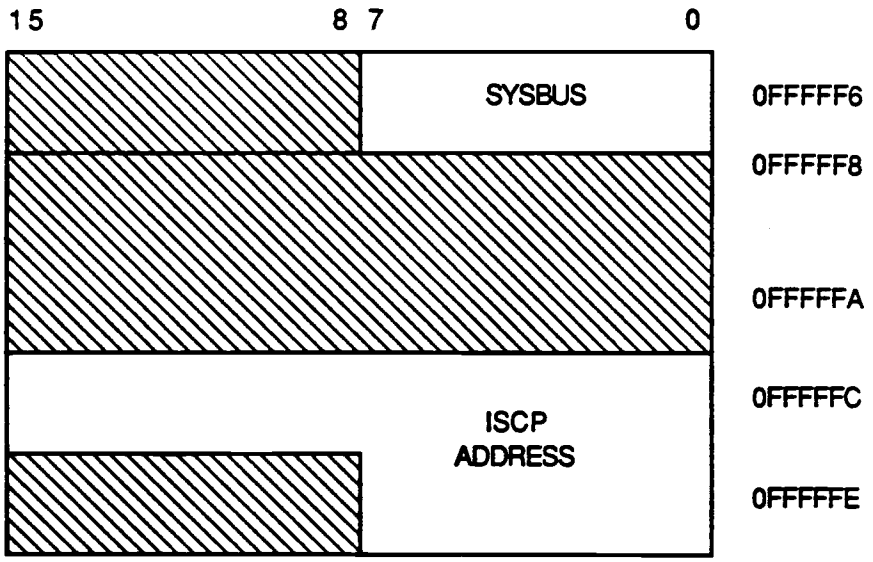


Figure 4.1 System Configuration Pointer.

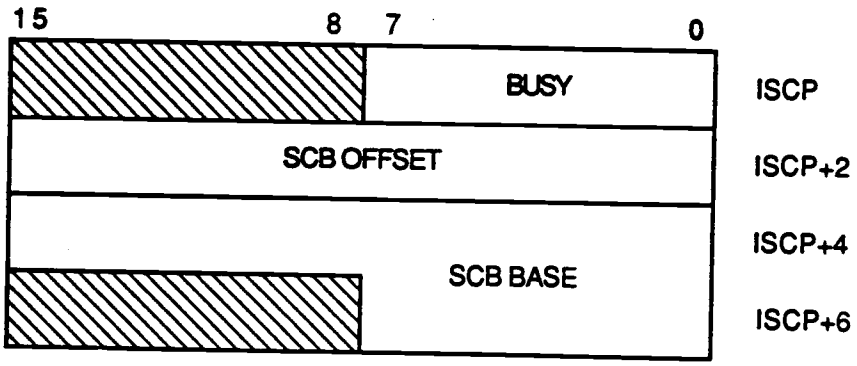


Figure 4.2 Intermediate System Configuration Pointer.

2) The 82586 reads the ISCP address from the location 3FFC to 3FFE.
3) The 82586 obtains the SCB OFFSET and SCB BASE fields.
4) THe 82586 clears the BUSY byte located in the ISCP.
5) The 82586 saves the SCB BASE in the internal register.
6) The 82586 writes the SCB STATUS word, specifying that the command unit has finished command execution and the inactive state of the command.
7) The 82586 enables the INTERRUPT signal to the host system.

4.2.2 Data Structure

Before issuing commands, the host system prepares an appropriate data structure for recognition by the 82586. The data structure is built in shared memory, its size determined by the command format of the 82586. A block diagram for the shared memory structure is shown in Figure 4.3 [11].

4.2.2.1 System Control Block. The SCB is used as a mailbox for communication between the host system and the 82586. The host system issues a control command to the 82586, obtaining status information on the command written by the 82586 via the SCB. The SCB format is

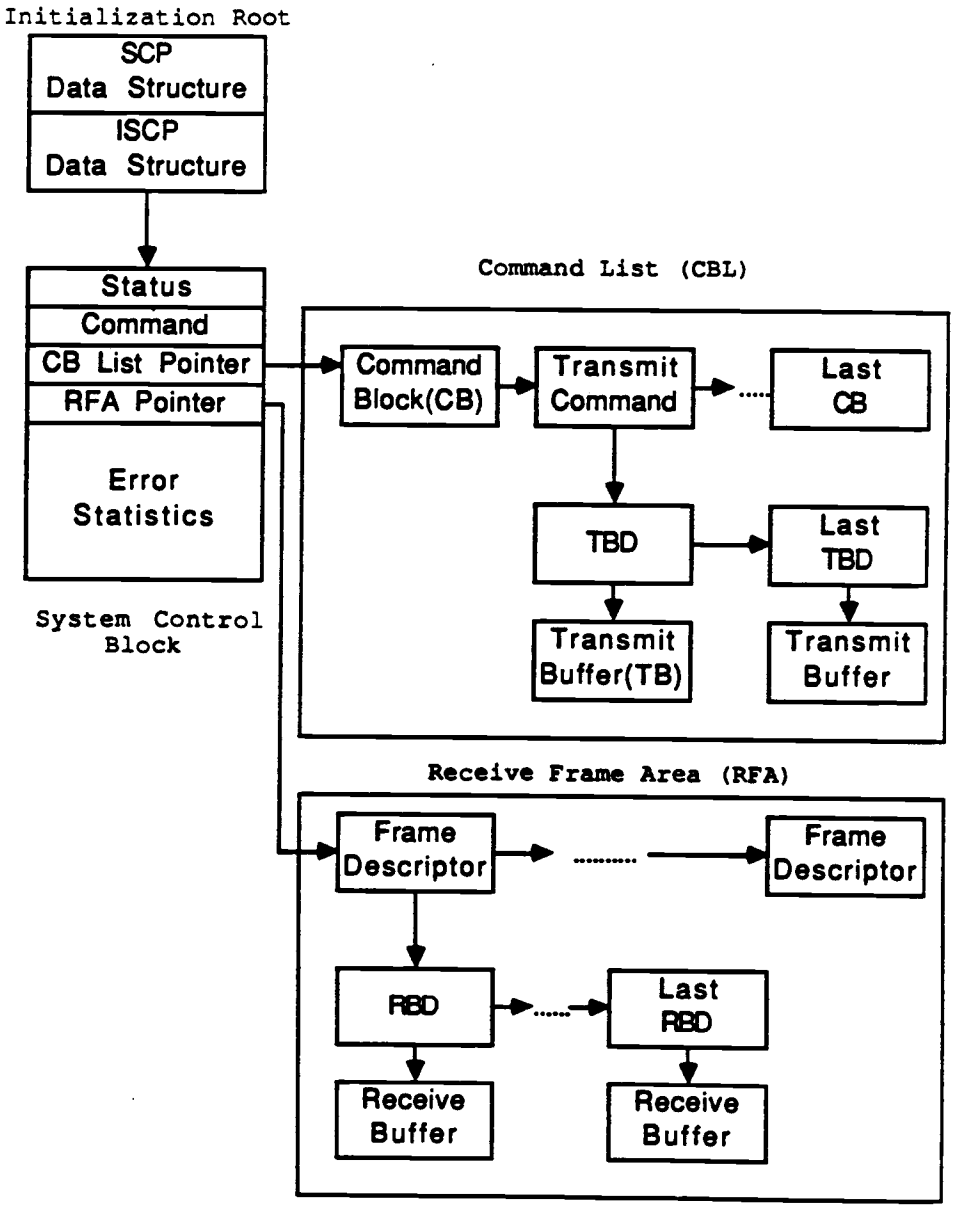


Figure 4.3 Shared Memory Structure [11].

illustrated in Figure 4.4. The SCB contains the following fields: (1) the status word, (2) the command word, (3) the 16-bit offset address of the first command, (4) the 16-bit offset address of the first Receive Frame Descriptor in the Receive Frame Area, and (5) error statistics [11].

4.2.2.2 Command List. The Command List (CBL) consists of command blocks, a list of Transmit Buffer Descriptors (TBD), and Transmit Buffers (TB). The command list has a chain structure, as shown in Figure 4.5. The previous Command Block (CB) indicates the current CB and the current CB points to the following CB. The transmit CB points to the TBD containing the actual data to be transmitted, the next TBD pointer, and the transmit buffer pointer. Each TBD has a TB, the size of which is subject to constant change in accordance with the programming of the configuration command block.

4.2.2.3 Receive Frame Area. The Receive Frame Area (RFA) consists of a list of Frame Descriptors (FD), Receive Buffer Descriptors (RBD), and Receive Buffers (RB). Each RFD is associated with a frame. Figure 4.6 shows the detailed RFA data structure.

Since the RB is relatively small in size, a receiving frame may be stored in a sequence of RBs chained into the complete frame. RBs are chained via the RBD, which points to a single RB.

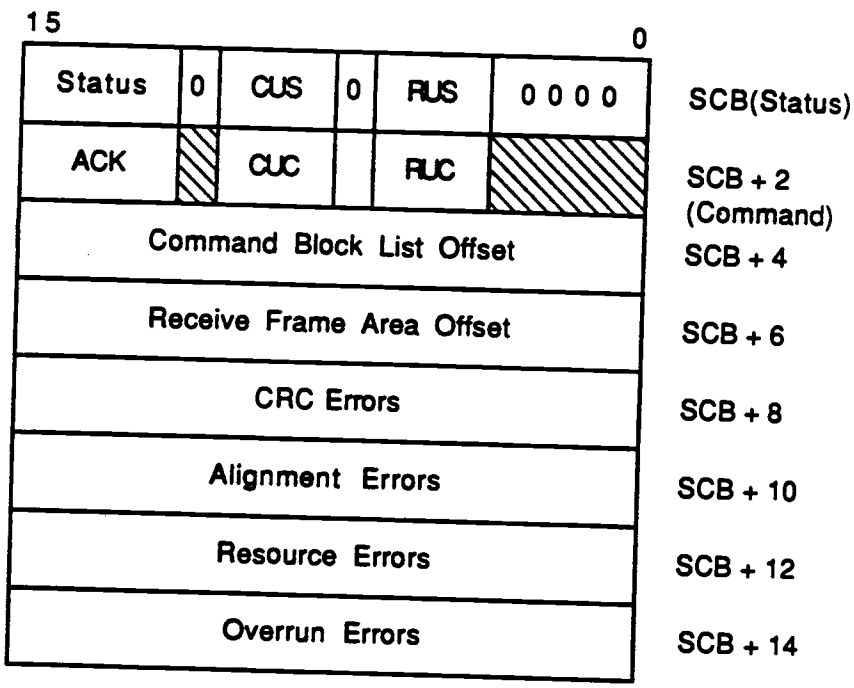


Figure 4.4 System Control Block Format.

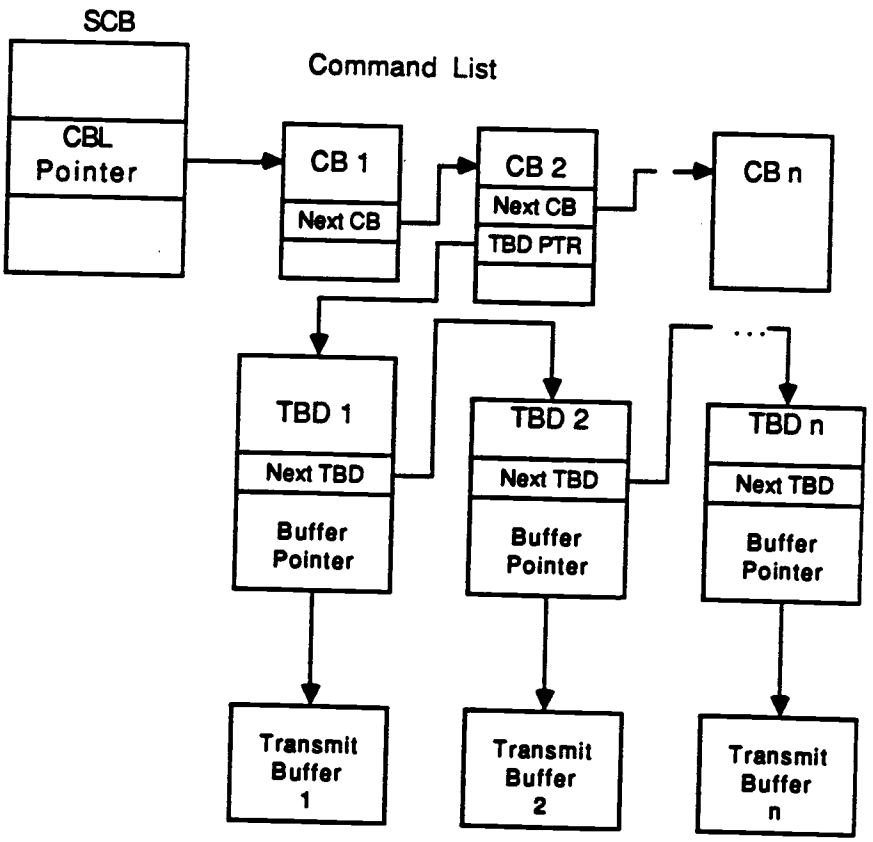


Figure 4.5 Transmit Command Data Structure.

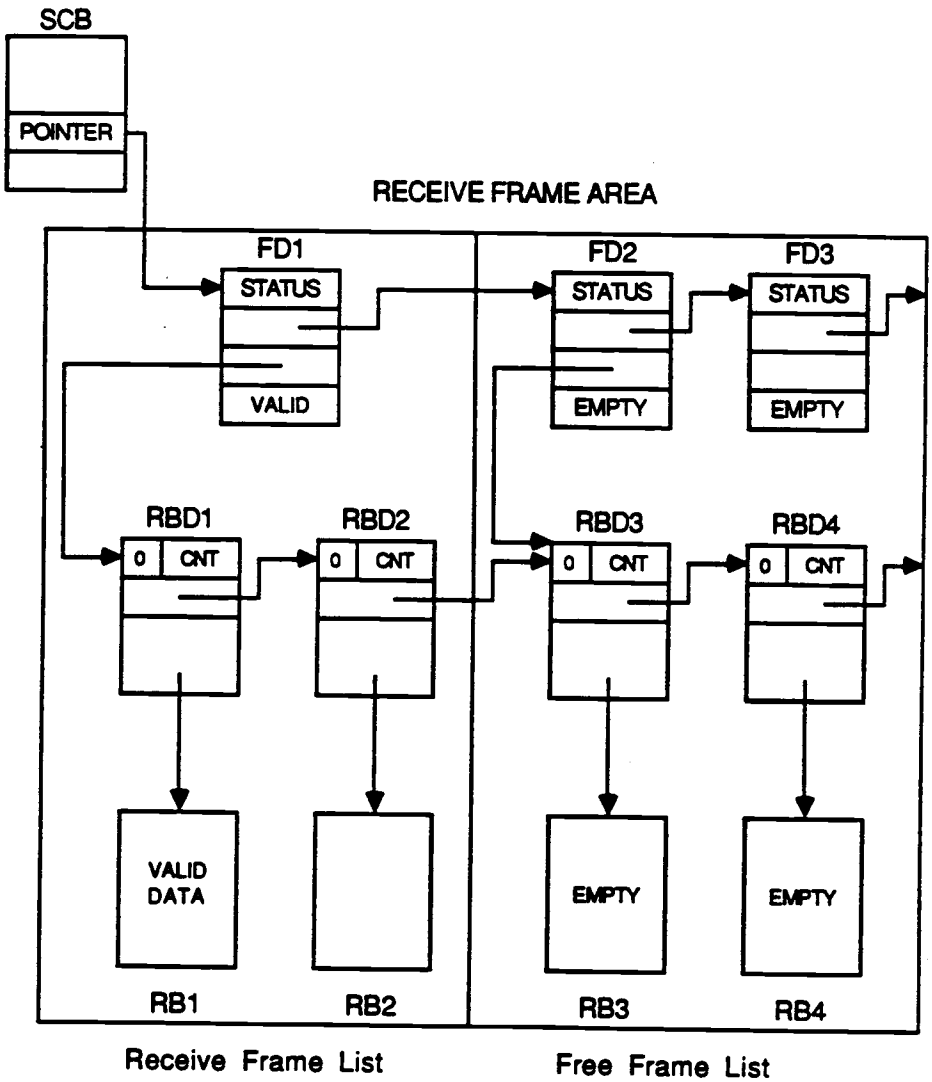


Figure 4.6 Receive Frame Area.

1) Frame Descriptor. The FD includes the following fields of 20 bytes each: (a) a status word, (b) a command word, (c) pointers containing addresses of the next FD and the first RBD for the frame, (d) a destination and source address of the frame, and (e) the length field of the received frame. The FD format is shown in Figure 4.7.
2) Receive Buffer Descriptor. The RBD contains information about the actual number of bytes occupied in the RB, the size of the RB, and pointers for the next RBD, current RB, and status word. The data structure of the RBD is shown in the Figure 4.8.

4.2.3 Control Units

The 82586 coprocessor has two control units, the Command Unit (CU) and the Receive Unit (RU).

4.2.3.1 Command Unit. Each action command has a CU, which is a logical unit performing action commands in three states: idle, suspended, and active [11]. In the idle state the CU does not execute commands and is not associated with a CB on the list. The suspended state specifies that the CU is not working, but that it is associated with a CB on the list. The active state signals that the CU is currently executing a command with its associated CB. The host system may issue a CU

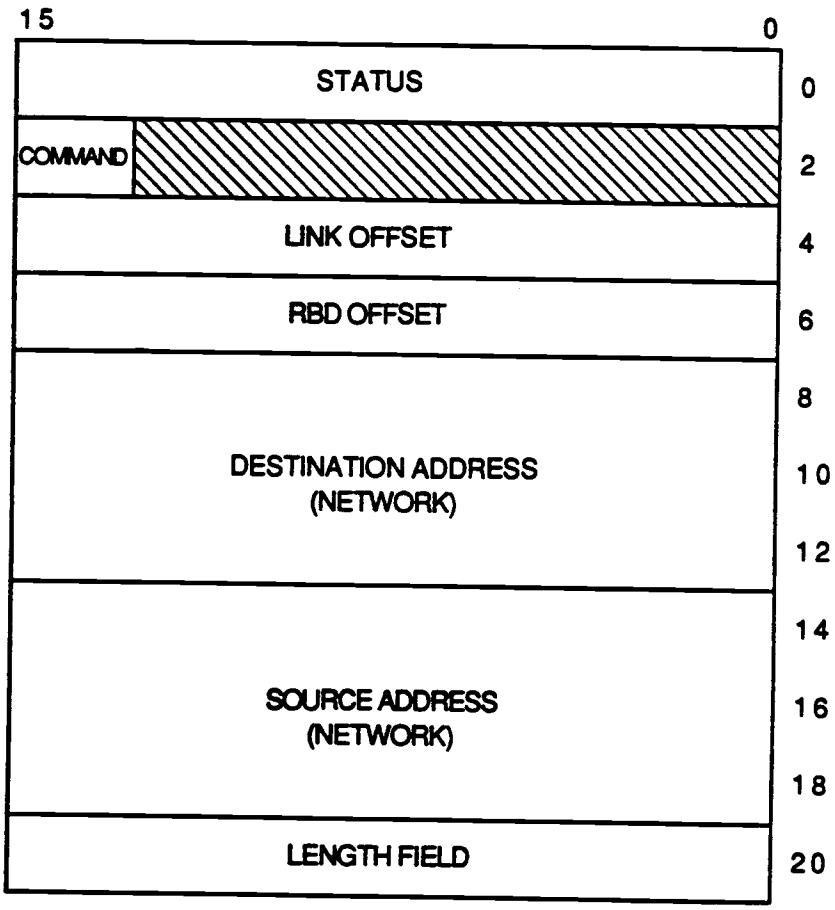


Figure 4.7 Frame Descriptor Format.

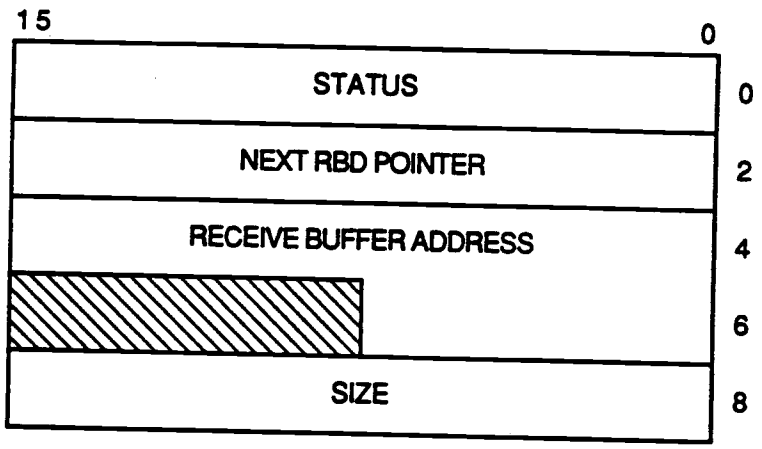


Figure 4.8 Receive Buffer Descriptor Format.

operation by setting the appropriate bits in the SCB format and setting bits in the command field of the action command.

4.2.3.2 Receive Unit. The RU is the logical unit that receives frames and stores them in shared memory. The RU utilizes unused RBDs and RBs provided in the initialization procedure by the host system and at any given time is in one of the following states:

1) Idle state, with no RBD or RB, indicating that the 82586 is not available for receiving incoming frames. This is the initial state of the RU;
2) No-Resources state with no memory resources, unlike the idle state. However, this state maintains statistics about discarded frames even though the 82586 is not able to receive incoming frames;
3) Suspended state, during which the RU ceases to store incoming frames though there are available memory resources;
4) Ready state, when the RU is ready to store incoming frames with free RBDs and RBs. The RU issues an interrupt after every received frame.

4.3 Programming

After initializing the 82586, the host system leaves all functions associated with data transfer between shared memory and the network to the 82586. The 82586 has various action commands and error statistics registers and programs are constituted by action commands in the command list.

4.3.1 Commands

An 82586 command set consists of eight types of action commands: Configure, Transmit, Diagnose, Dump, NOP, Multicast Address Setup, Individual Address Setup, and Time Domain Reflectometer (TDR).

1) Configure. The configure command is used to set parameters for the appropriate 82586 operating mode. The configuration format consists of 32 bytes as follows:
   a) The status field written by the 82586 to specify the command status;
   b) The command field prepared by the host system to set up the command;
   c) The next command block pointer;
   d) Controlling parameters for the 82586; and

e) Configuration parameters for Ethernet. The default values of the configuration are compatible with the IEEE 802.3 Standard.

2) Transmit. The transmit command initiates transmission and has the following fields:
   a) The status field, including the number of collisions experienced by a frame;
   b) The command field;
   c) The address of the next CB;
   d) The pointer for the first TBD;
   e) The destination address field with 6 bytes; and
   f) The length field of the frame.

3) Diagnose. The diagnose command initiates the 82586 self-test procedure, reporting the status of the internal test. The status field, the command field, and the address of next CB are included in the report.

4) Dump. The dump command allows the 82586 to dump its internal registers into shared memory. This command is used as a programming self-diagnostic tool and its format includes the status field, the command field, the link pointer to the next CB, and the pointer for the dump area. The size of the area is 170 bytes [11].

5) NOP. This command results in no action by the 82586, except as performed in normal command processing. The format includes the status, command, and pointer fields.

6) Multicast Address Setup. This command is used to set multicast addresses in the 82586. In addition to the status, command and pointer fields, the format includes the multicast address list and the multicast count fields to indicate the number of bytes in the list.

7) Individual Address Setup. This command allows the 82586 to load a specific address in the network. It is used to recognize the destination address during reception and the insertion of a source address during transmission. The format includes a 6-byte individual address in the normal format.

8) Time Domain Reflectometry. TDR is used to detect and locate cable faults caused by either a short or opening on the coaxial cable [11]. After transmission of "All Ones" to the serial link, the 82586 sets the internal timer to measure elapsed time from the time of transmission initiation until an echo is obtained. Command format includes the results field after testing, and status and command fields.

4.3.2 Receiving and Transmitting Frames

Figure 4.6 illustrates the RFA data structure, which is the initial structure for receiving a frame. A frame is received when its length is at least 6 bytes and its address matches the individual address or the multicast or broadcast address. All receiving operations are under control of the receive unit.

The RU sets up a frame descriptor and RBD for every received frame in shared memory. The receive DMA loads data from the receive FIFO in each RB and after a complete frame is loaded the RU closes the last RBD and the frame descriptor, setting up the data structure for the next receiving frame.

To transmit a frame, a data structure is provided (as shown in Figure 4.5) and a transmit action command, controlled by the command unit, is issued. The combined operations of receiving and transmitting frames are shown in Figures 4.9 and 4.10. The host system sets up data structures as previously described, sending a CA-RESET sequence to initialize the 82586. The host system then initiates the receive unit by enabling a card interrupt signal and running normal network applications.

The 82586 on the interface card sends an interrupt whenever it has completed any action commands. Hence, the 82586 determines whether the received interrupt has

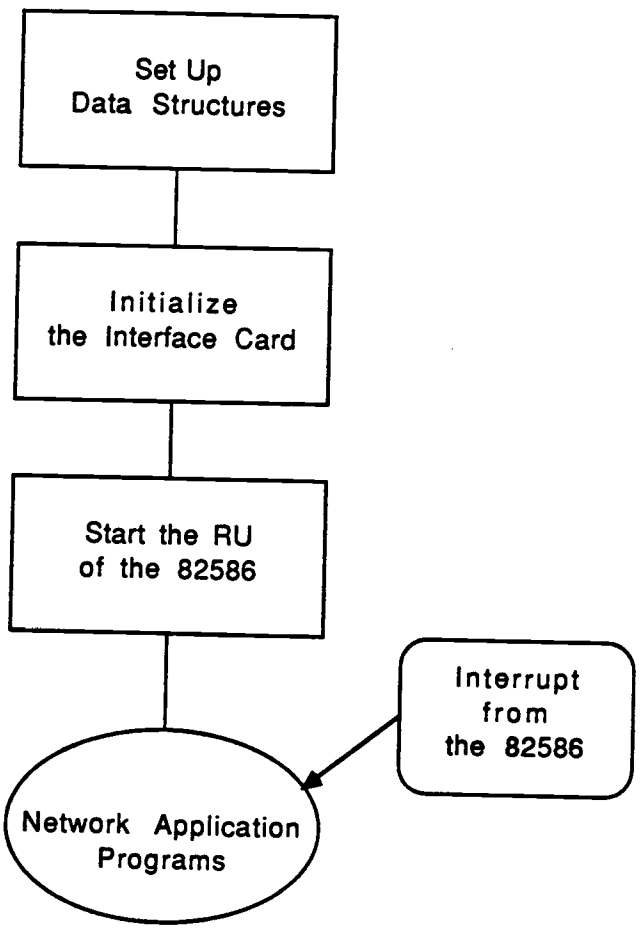


Figure 4.9 Flow Chart for Host System Operation.

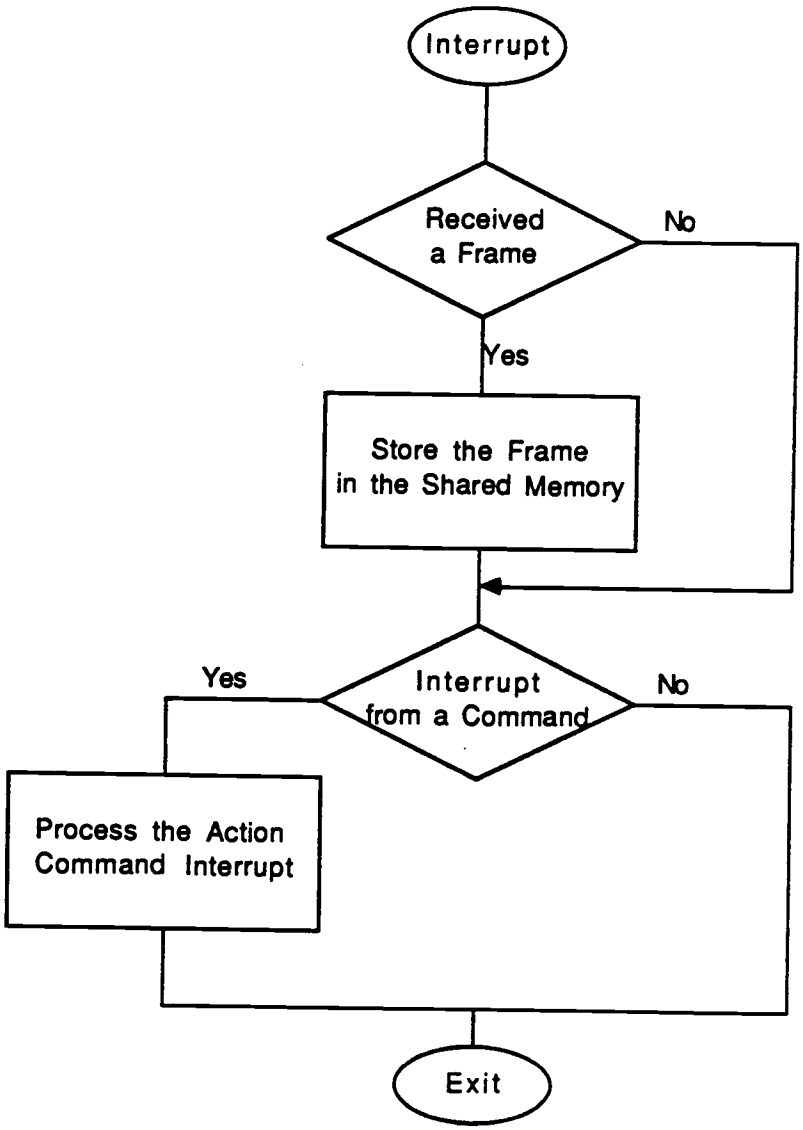


Figure 4.10 Flow Chart of Interface Card Operation.

been asserted for frames received or any other action command. If an interrupt for received frame has been initiated, then the 82586 stores the data in the receive FIFO in the shared memory RB. If no interrupt has been issued, the 82586 processes appropriate operations for the action command, which may be a transmit command.

CHAPTER 5: CONCLUDING REMARKS

This project has encompassed a practical implementation of an Ethernet LAN interface card. The general terminology of LANs, such as topologies, access schemes, and Ethernet and IEEE 802.3 standards, has been discussed and the entire design procedure, from the block diagram to the timing diagram, has been described.

The major design feature was the inclusion of an arbitration interface to avoid contentions between the host system and the interface board coprocessor. Almost all of the control logic consists of various PALs, allowing a reduction in the number of components and logic design flexibility. Moreover, use of the coprocessor data structure was also analyzed. The use of unshielded twisted pair cable as the communication medium provides for the low cost of this Ethernet installation.

BIBLIOGRAPHY

[1] *Advanced Peripherals IEEE 802.3 Local Area Network Guide*. National Semiconductor Corporation, Santa Clara, CA [n.d.].

[2] Cashin, J. "Open System Architecture." *Interface '82 Proceedings*. McGraw-Hill, New York 1982., pp.137-146.

[3] Dash, G. "Designing I/O to Reduce Emissions in Computing Equipment." *Symposium Record of EMC Expo 86 International Conference on Electromagnetic Compatibility*, Washington, DC, June 16-19, 1986, pp. T14.14-T14.16.

[4] *Designing Cards and Drivers*. Apple Programmer's and Developer's Association, Renton, WA, August 28, 1987.

[5] Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. *The Ethernet: A Local Area Network, Data Link Layer and Physical Layer Specifications*. Version 1.0, September 30. 1980.

[6] Jordan, L. and Churchill, B. *Communications and Networking for the IBM PC & Compatibles*, Rev. Ed., Prentice-Hall Press, New York, 1987.

[7] Kessler, G. "Ethernet Vs. IEEE 802.3." *LAN Magazine*, July 1987, pp.54-57.

[8] Lefkon, D. "A LAN Primer." *Byte*, Vol. 12, No.8, July 1987, pp.147-154.

[9] *Macintosh Family Hardware Reference*. Apple Programmer's and Developer's Association, Renton, WA, March 2, 1987.

[10] Metcalfe, R. M. and Boggs, D. R. "Ethernet: Distributed Packet Switching For Local Computer Networks." *Communications of the ACM*, Vol. 19, No.7, July 1976, pp.395-404.

[11] *Microcommunications Handbook*. Intel Corporation, Santa Clara, CA, 1987.

[12] *NuBus--a Simple 32-Bit Backplane Bus P1196 Specification*, Draft 2.0, IEEE, New York, 1986.

[13] Reiss, L. *Introduction To Local Area Networks with Microcomputer Experiments*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[14] Shotwell, R.E. (Ed.). *The Ethernet Source Book*. Elsevier Science Publishing Co., Inc., New York, 1985.

[15] Silva, E. da. *Introduction to Data Communications and LAN Technology*. Collins, London, 1986.

[16] Smith, H.T. et al. "Using the Macintosh on a Unix Network." *Byte*, Vol. 12, No.8, July 1987, pp.159-166.

[17] Stallins, W. *Local Networks*. Macmillan Publishing Co., New York, 1987.

[18] Tanenbaum, A.S. *Computer Networks*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[19] Tropper, C. *Local Computer Network Technologies*. Academic Press, New York, 1981.

[20] Williams, G. and Thompson, T. "The Apple Machine II." *Byte*, Vol. 12, No.4, April 1987, pp. 85-106.

[21] Wright, M. "Software." *EDN*, Vol. 33, No.5, March 3, 1988, pp.102-108.

[22] Zimmermann, H. "OSI Reference Model--The ISO Model of Architecture for Open System Interconnection." *IEEE Transactions on Communications*, Vol. COM-28, No.4, April 1980, pp.425-432.