

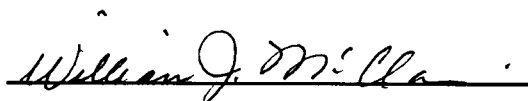
To the Graduate Council:

I am submitting herewith a thesis written by Emily Elizabeth Muly entitled "An Application-Specific Implementation of a Local Area Network." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.



Dr. David W. Straight, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Associate Vice Chancellor
and Dean of The Graduate School

**AN APPLICATION-SPECIFIC IMPLEMENTATION
OF A LOCAL AREA NETWORK**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Emily Elizabeth Muly

May 1994

ACKNOWLEDGEMENT

I would like to thank my advisor, Dr. David W. Straight, for his support and interest in this thesis topic, and the committee members, Dr. Donald W. Bouldin and Dr. William J. McClain, for their help in bringing this work to a conclusion.

I also would like to acknowledge the support of my husband, Brian Schmidt, who encouraged me in the pursuit of a Master's Degree.

ABSTRACT

In today's academic and commercial world, there are a variety of commonly used local area network implementations. Each network scheme has advantages and disadvantages with respect to particular applications or performance criteria. A network design that is right for one application may not be the best choice for a totally different application. For some specialized applications, none of the existing networks are appropriate. In particular, redemption games, which are simple, mechanical games found in arcades, have not been networked previously and are the focus of this thesis.

This specialized application requires only a subset of the functions provided in a full-function network protocol, and must be implemented at minimal cost. Ethernet, for example, provides excess functionality, which translates into more cost than is justified by this specific application. If the nodes of a network can be satisfactorily connected by a less robust network, then clearly it is worthwhile to implement a new and different network that will provide the benefits of simplicity and reduced costs.

A review of existing network protocols was made for the purpose of selecting the features to be included in the new design. Software was developed to implement the network protocol. A prototype of the network hardware was assembled, and the software was tested on this hardware implementation.

TABLE OF CONTENTS

CHAPTER	PAGE
1. Introduction	1
1.1 Design Considerations	2
2. Network Topology	5
2.1 Star	5
2.2 Ring	6
2.3 Bus	7
2.4 Tree	7
3. Network Transmission Media	9
3.1 Twisted Pair Cable	9
3.2 Coaxial Cable	10
3.3 Fiber-Optic Cable	10
3.4 Twisted Pair versus Coaxial Cable	11
4. Network Protocol	12
4.1 Physical Layer	12
4.2 Logical Link Control Layer	13
4.3 Medium Access Control Layer	13
4.4 LAN MAC protocols	13

CHAPTER	PAGE
4.4.1 CSMA/CD	14
4.4.2 Token Bus and Ring	14
4.4.3 Empty Slot	15
4.4.4 FDDI	16
4.5 Comparison of Protocols	16
4.5.1 Delay	16
4.5.2 Capacity	17
4.5.3 Number of Stations	17
4.5.4 Frame Length	17
4.5.5 Propagation Delay	18
5. Network Hardware	19
5.1 Topology	19
5.1.1 Star	19
5.1.2 Ring	20
5.1.3 Bus/Tree	20
5.2 Medium	21
6. Hardware Design	22
6.1 Development Hardware	22
6.1.1 CeSET Processor Controller and Audio Board	24
6.1.2 Network Board	25
6.1.3 ROM Emulator	27

CHAPTER	PAGE
6.1.4 Intel 386	28
6.1.5 Debugging Tools	28
6.2 Final Configuration	29
7. Software Design	30
7.1 Design Considerations	30
7.1.1 Real Time Network Performance	30
7.1.2 Flow of Information	31
7.1.3 Protocol Design	31
7.2 Software Implementation	32
7.2.1 Memory Management	32
7.2.2 Initialization Software	35
7.2.3 Network Communication	36
7.2.4 CeSET Microcontroller and Network Board Intercommuni- cation	47
8. Testing	50
8.1 Correctness of Shared Memory Matrix	51
8.2 Correctness of the Network Management Table	51
8.3 Network Communication Rate	52
9. Conclusion	53
LIST OF REFERENCES	56

CHAPTER	PAGE
VITA	57

LIST OF FIGURES

FIGURE	PAGE
6.1 Development Hardware Configuration	23
7.1 Memory Map of Network Shared Data	34
7.2 Network Management Table	38
7.3 Flow Diagram for Basic Operation	43
7.4 Hardware Interface Between Processor and Network Boards . . .	48

LIST OF TABLES

TABLE	PAGE
7.1 Contents of Memory Locations	34
7.2 Packet Field Description	39
7.3 Interface Byte Configuration	48

CHAPTER 1

Introduction

The Creative Electronics and Software Company (CES) in Elgin, Illinois, manufactures the CeSET microcontroller board. This board is used in the area of redemption games as a processor board. The microprocessor on this board is a Zilog Z80, 8-bit chip, and, although not currently state-of-the-art, this controller board continues to perform well in certain applications. This thesis consists of developing the hardware and software to connect up to sixteen (16) CeSET boards for the purpose of information sharing. To accomplish this, one must define the requirements for the network, and then make design choices based on these criteria. When striking out on your own, every aspect of the network must be carefully considered. Three essential things must be decided:

1. What will physically connect the boards together, i.e. what will the transmission medium be?
2. How will these boards be connected, i.e., what will the topology be?
3. How will the problem of contention for the network be decided, i.e., what will the protocol look like?

1.1 Design Considerations

When making the decisions regarding transmission medium, network protocol, and topology, it is necessary to define fully the network requirements of the application as well as the constraints inherent in the application. The network, hereafter referred to as Emnet, can be defined by the following criteria:

1. Shared Data

Each game will have data that it wishes to share with all other stations on the network. The application programmer can then use this data however he wishes. Fifteen (15) bytes was deemed the amount of data that each station would share with all other games on the network.

2. Maximum Number of Nodes

Sixteen (16) games will be the maximum number of nodes allowed on the network.

3. Simplicity

The network hardware must be as simple as possible and identical for every machine. This requirement is necessary to maintain ease of design, production, and sales. Also, it must be assumed that the end user will not be technically knowledgeable, and thus will not tolerate anything more complicated than plugging in the system and watching it work. The setup can not be difficult, and continued maintenance must be simple.

4. Reliability

When one station goes down it is essential that the remaining stations continue to function. It is highly likely with this application that individual stations will be going down intermittently.

5. Rate of Communication

Existing state-of-the-art technology more than provides an ample rate of communication for this application. Thus, a slower alternative based on human-reaction time can be implemented. The determined maximum for a full update of shared data, based on human-reaction time, is 500 milliseconds.

6. Broadcast Capability

A data packet sent by an individual station is of interest to all other stations or, at a minimum, a subset of the group of stations.

7. Cost

To create a commercially viable solution the implementation must be cost-effective. Since the CeSET board itself is not costly, the board designed to accompany the CeSET board to perform network functions must be inexpensive.

8. Lead Station

The network might be used by these games to issue commands to all games on the network. It is not desirable to have multiple games with the ability

to issue commands. Thus, one station must be identified as a lead station.

9. Transmission Errors

Transmission errors will be tolerated, thus complex error checking is unnecessary.

CHAPTER 2

Network Topology

To define the topology of a network is to define the way in which nodes or stations of a network are connected. Topology defines the path that data must take between any two nodes. The network topology is an essential element in determining the ability of a LAN to perform its particular task. The topology selected for a network influences its reliability, availability, computing power, and rate of information transfer. Several topologies exist which can be used as a model. Each topology has its pros and cons, and, depending on the application, a good choice can be made. The most common topologies include: star, ring, bus and tree. The following is a brief discussion of the topologies considered.

2.1 Star

In a star topology, each station is connected by a point-to-point link (a link that connects two nodes) to a central hub. The hub performs the routing functions for all stations in the system based on the destination address of the message packet. This centralized communications control strategy requires a complicated central switch or controller. The benefit of having such a complex node is clear: the communications burden on all other stations is minimal. However, the central

hub in a star network can be a limiting factor for growth. A hub can only support a predefined number of connecting stations: if a host is to be added to the network, there must be a spare port for the link at the hub. This can be a problem if growth is important for the particular application. The hub, also, should be a more powerful machine than the nodes it connects to handle the traffic and must possess extra hardware to guarantee reliability. The simplicity of communication of all non-hub nodes is a plus; however complexity and other issues related to the hub may be a strong deterrent to choosing this network topology for this application.

2.2 Ring

A network based on a ring topology consists of active stations attached to the network and a set of repeaters joined to form a closed loop. A repeater is a device that is capable of receiving data on one link and transferring it to another link without having to buffer the data. Typically, data flows in one direction around the ring. With this ring/repeater scheme the network devices are relatively simple, and the stations must provide packetizing and access control logic. To send a message on this type of network, the sending station places the message on the medium, and the message travels around the ring until reaching its destination. At this point two possible schemes exist: (1) the receiver can take the message off the network or (2) the message can propagate until reaching the sender, which

removes the message from the network. Addition of a station can be easy, because all that is needed is to unplug a connector and insert it into the new station that, in turn, connects into where the old station was connected. Routing is not a problem with the ring topology because all messages follow the same path. If a station that is connected to the ring goes down and no means exists for the path to bypass the down station, then the whole network goes down.

2.3 Bus

In the very common bus topology, a transmission from any station can be received by all other stations. There are no switches and no repeaters (in the simplest form). The medium is referred to as multipoint or broadcast. Because only one station can transmit at a time, some form of access control is required to determine whose turn it is. The method of bus control is not inherent to the topology and can be either centralized or decentralized. Centralized control would indicate that stations receive the right from one station the right to gain access, and decentralized control implies that each station determines when it can access the network. As is the case with a ring, bus topology removes the need for routing.

2.4 Tree

The tree topology is a generalization of the bus topology. The cable can be made to branch by the use of cable splitting. As with the bus topology, the network

is relieved of the entire communications processing burden. The transmission medium is passive. Pros and cons associated with a bus topology would also apply to the tree topology.

CHAPTER 3

Network Transmission Media

Transmission medium refers to the physical path between the transmitter and receiver. The transmission medium can be point-to-point or broadcast/multipoint. A point-to-point link connects two nodes directly without passing through any intermediate nodes. A multipoint link is a single line that is shared by more than two nodes. The most common transmission media used in LAN's are twisted pair cable, coaxial cable, and fiber optic cable.

3.1 Twisted Pair Cable

A cable of twisted pair consists of two wires, usually copper, which are twisted. This method has been in use for a long time and is inexpensive. It is also one of the most common transmission media for analog and digital signals. This medium can be used for both point-to-point and multipoint transmission. Twisted pair cable is a less expensive alternative for multipoint broadcast than coaxial and is the least expensive medium overall. As usual, a lower-cost alternative can translate into a lower performance alternative. Objectives must be weighed to determine if the lower performance is acceptable, since low cost is so desirable. The lower performance is seen in the reduced bandwidth supported and higher error rates

due to the wire absorbing and emitting a high amount of electrical interference.

3.2 Coaxial Cable

Coaxial cable is the most popular choice of transmission medium. This cable offers high speeds over moderate distances at moderate prices. As with twisted pair cable, coaxial cable is comprised of two wires. However, the design of the coaxial cable offers a higher bandwidth than the twisted pair cable. Coaxial cable provides excellent isolation from external noise and crosstalk. It is used for both multipoint and point-to-point communication. There are two types of coaxial cable: 75 Ohm broadband, used for both digital and analog transmission with frequency modulation, and 50 Ohm baseband, which is used for digital transmission only. The 50 Ohm baseband cable can run at speeds of 10 Mbps and the 75 Ohm broadband cable can run at speeds of 50 Mbps. In general, coaxial cable is superior to twisted pair cable for higher frequencies when noise immunity is considered. An advantage of broadband coaxial cable is the availability of low-cost components developed for the cable television industry, although baseband connections and interfaces are even less expensive.

3.3 Fiber-Optic Cable

A fiber-optic cable consists of a very thin, flexible, glass or plastic strand through which light can be transmitted by reflection off the tube's internal walls.

Fiber-optic cable operates with much greater speeds and over greater distances than does coaxial cable. Fiber-optic cable offers the lowest error rate of all other media, which is about 1 bit in every 10^{10} bits transmitted. This transmission medium also offers immunity to electrical interference that can disrupt communications on other types of cables. In addition to these advantages, the cable itself is small and lightweight. As might be expected, there is a higher cost associated with fiber-optic cable, although the price continues to decrease with time. There is some difficulty associated with splicing fiber optic cable. In addition fiber optic is less amenable to abuse.

3.4 Twisted Pair versus Coaxial Cable

Fiber-optic cable was not considered as a medium choice in this system, because of the cost factor. Twisted pair and coaxial were considered, although, in general, coaxial cable is more expensive by a factor of two or three than twisted pair. Another significant difference between coax and twisted pair cable is in the effective data rate that can be supported. Coax is capable of coexisting with strong rf (electromagnetic fields), while this will cause interference and disrupt communication on twisted pair cable.

CHAPTER 4

Network Protocol

The ISO OSI model for networking specifies seven layers, which together provide services associated with networking. The seven layers are referred to as the physical, data link, network, transport, session, presentation, and application layers. Each layer has a well-defined purpose and function.

The Local Area Network (LAN) standard, however, is not in need of all seven layers and can essentially function with only the first two layers, as long as the data link layer performs a few functions normally associated with the network layer. Corresponding to the OSI reference model is the LAN reference model. Basically, the LAN reference model has three layers that correspond to and perform the functions of the two lower layers and part of the third layer in the ISO reference model. The Physical Layer of the LAN model corresponds to the Physical Layer of the ISO model. The medium access control (MAC) and the logical link control (LLC) of the LAN model corresponds to the data link layer of the ISO model.

4.1 Physical Layer

The Physical Layer is concerned with transmission of a bit stream over a physical link. The physical layer incorporates the mechanical, electrical, and

procedural characteristics necessary to establish, maintain, and deactivate the physical link. Clearly, any definition of a network protocol would need such a capability.

4.2 Logical Link Control Layer

The Logical Link Control (LLC) provides a medium-independent protocol for both connectionless and connection-oriented communication. This layer attempts to make the physical link reliable and provides the means to activate, maintain, and deactivate the link.

4.3 Medium Access Control Layer

The Medium Access Control (MAC) layer provides a means of sharing a common medium among a number of different devices. This control is needed to avoid deadlock and to allow efficient utilization of bandwidth. This layer is medium dependent.

4.4 LAN MAC protocols

The majority of LANs have been based on the bus and ring topologies. Because the communication medium is shared among attached devices, an access scheme must be provided to work out conflicts. LANs most often use distributed control

in that they allow the devices, with equal status, to resolve cable contention. These schemes are the MAC protocols. The most prominent MAC protocols are CSMA/CD, token bus, token ring, and empty slot.

4.4.1 CSMA/CD

CSMA/CD is an acronym for "Carrier Sense Multiple Access with Collision Detect". With this MAC protocol, specified by the IEEE 802.3 standard, any station wishing to transmit information first listens for any traffic on the line. If the line is busy, the station waits a random amount of time and tries again. If the line is free, the station transmits the packet. While transmission takes place, the station listens for a potential collision with another station's packet. A collision occurs when two or more stations attempt to use the medium at once. If a collision takes place, each station stops the transmission of the remainder of the packet, waits a random amount of time and tries again. The random delays serve to separate the access attempts of the colliding stations, hopefully precluding further collisions.

4.4.2 Token Bus and Ring

The token bus and ring operate in a similar way. A token circulates among the stations on the network, carrying with it the right to transmit. In a bus, the stations are in a 'logical' ring, where every station knows the address of the station that precedes it and follows it. The stations of a token ring network are physically connected in a ring by a series of point-to-point links. When a station

receives the free token in the token ring scheme, it may mark the token as busy and begin transmission of a packet. After it has completed its transmission, it passes the token to the next station in line, marked as free.

In the case of the token bus network, the token must be sent to the station that is logically next. This method is fair in that all connected stations get a turn in a round-robin fashion. The token bus IEEE 802.4 standard specifies a choice among three physical layers. Two of the standards are baseband systems similar to Ethernet (CSMA/CD) and one is a broadband system. Between these three standards, data rates of 1, 5, or 10 Mbps are offered. In all three systems, 75 ohm coaxial cable is the medium used. The token ring IEEE 802.5 standard specifies that data may be inserted on the ring at one of two data rates. The two data rates are 1 or 4 Mbps, however IBM does offer a 14 Mbps ring.

4.4.3 Empty Slot

With the empty slot principle, or Cambridge Ring, fixed-length minipackets continuously circle the physical ring. This minipacket is marked as full or empty. A station wishing to transmit may use a minipacket if it is marked as empty. The minipacket is not allowed to be used by the same station twice, unless the minipacket has gone all the way around the ring once after being used the first time. The minipacket tends to be very short, about 40 bits, in order to fit on the closed ring. The medium used is normally twisted pair wires.

4.4.4 FDDI

FDDI, or Fiber Distributed Data Interface, is another kind of ring. FDDI is designed to be a very fast ring that can have considerable physical extent. Fiber optic cable is used to connect nodes or repeaters, at a distance of up to 2 Km. A total fiber length of up to 200 Km can be achieved. The network can run at a very high speed of 100 Mbps.

4.5 Comparison of Protocols

There are many ways to compare these different protocols: delay, capacity, number of stations, frame length, and propagation delay.

4.5.1 Delay

Delay in a network refers to the time that a station must wait before accessing the medium when it has a frame to transmit. This is particularly important for time-critical applications. In a network implemented with the CSMA/CD protocol, there is no upper bound on a station's access time, because a station is never guaranteed the use of the network. When there is high traffic on the network and a time critical application, CSMA/CD is not the best choice. Instead, a token passing protocol is a better choice. In a token passing scheme there is an upper bound on the access time to the network, which is the time it takes for every station to transmit a packet, and for the token to be available. The lower bound

for the access time is longer than for the CSMA/CD protocol, however.

4.5.2 Capacity

The capacity of a network refers to the maximum raw data rate (R) in bits per second of the medium. The standard CSMA/CD network has $R = 10$ Mbps, which means that every station transmits its frames at this rate. As stated earlier the data rate for the token bus can be 1, 5, or 10 Mbps, depending on which physical layer standard is used. In the token ring standard, rates of 1 or 4 Mbps are specified although it can run at a speed of 14 Mbps. The standard Cambridge Ring operates at 10 Mbps. Work has been done to implement a "fast ring" that operates at 100 Mbps. FDDI operates at 100 Mbps and faster Ethernets are being developed.

4.5.3 Number of Stations

When considering the number of stations to include in a network, it is clear that the more stations on the network, the less access time made available to each station. It is clear that any of the protocols considered can accommodate the (16) sixteen nodes for this application.

4.5.4 Frame Length

Frame length can also affect the performance of a network. Frame length is the number of bits in the transmitted frame. The CSMA/CD protocol has a minimum

frame size of 72 bytes and a maximum of 1526 bytes. The token bus standard specifies a 8191 byte maximum. The token ring has a 21 byte minimum data packet size and no specified maximum. There is a token holding timer that limits a station to the amount of time it may hold on to the token and thus affects the amount of time it has to transmit a data packet.

4.5.5 Propagation Delay

Propagation delay is defined as the time taken for a frame to travel the length of the network. Propagation speeds are much the same for the different networks, and therefore the propagation depends on the length of the network.

CHAPTER 5

Network Hardware

A bus topology utilizing twisted pair was chosen for the Emnet. This decision was based on flexibility in physical characteristics and cost.

5.1 Topology

Most local area networks utilize a ring or bus topology. However, since the network designed for this application has somewhat different criteria than other LANs, all topologies were considered initially. Factors considered are discussed below.

5.1.1 Star

The star topology was not utilized, because the network hardware in the Emnet design must be the same on every node. In a star topology this is not the case. A hub is needed with complicated hardware, and even though the other stations have a minimal hardware burden, the hub complexity can not be overlooked, nor can the non-uniformity of hardware between the hub and the nodes. Another disadvantage of the star topology is the dependence of the network on the reliability of the hub. If there is a hardware problem with the hub, then the network is disabled until

a replacement is received. It is assumed with our application that the user will not be knowledgeable about the network system and will be unable to correct problems that occur; so the less complicated the equipment the better. Thus, the complexity of the hub is a major disadvantage based on problems that may occur and the inability of the user to deal with network issues.

5.1.2 Ring

One of the more important design criteria was not to have the entire network suffer because one node goes down. It is important to have the network in continuous operation, even though some nodes on the network may be down. The inability of the ring topology to deal with such a case removed it from consideration.

More complicated rings may be designed to deal with this problem, by maintaining a backup ring, by automatically by-passing the inoperative node, or other schemes. Adding such things to a ring make it complex and more expensive to implement and, thus, cost made this scheme unacceptable.

5.1.3 Bus/Tree

Bus is a specific case of the tree topology and is easier to implement. While the bus topology is not always the first choice for a local area network, it was selected as the best for this application. Bus topology is synonymous with broadcast. Broadcast is appropriate for this application because all packets that are "information" packets, as opposed to "control" packets, must be received by ev-

ery station on the network. Another advantage was that twisted pair cable and baseband coaxial cable can both be used with this topology, and, therefore, there is a choice of medium.

5.2 Medium

Fiber optics and broadband coaxial cable are more expensive media than baseband coaxial and twisted pair cable. They are also more complicated from a hardware standpoint. Based on these factors, both fiber and broadband coaxial are undesirable. In addition, the immense capacity of these two media is greater than needed for this application. Both baseband coaxial cable and twisted pair can be used with the bus topology. Twisted pair was selected, based on the following criteria.

- The cable will not be in the presence of significant rf fields.
- Twisted pair is less expensive than coax.
- The bandwidth provided by twisted pair is suitable for this application.

CHAPTER 6

Hardware Design

6.1 Development Hardware

A block diagram of the development hardware configuration is shown in Figure 6.1. This system includes three each of the CeSET microcontroller board and a network board, along with a ROM emulator and an Intel 386 machine.

Although the ultimate goal of this project is to provide a network for multiple CeSET microcontroller boards, special features of the microcontroller are utilized during the development of the network board. The final configuration for the network board has a Z80 microprocessor to execute the the network software. However, for the design phase, the Z80 microprocessor on the CeSET board was utilized, in lieu of the network board being equipped with a microprocessor.

This scheme was chosen because functions already available on the CeSET microcontroller board provided important software debugging capabilities. But, it should be noted that in the final design a Z80 on the network board will be dedicated to network operations, while the Z80 on the CeSET microcontroller board will be utilized only for the specific applications software.

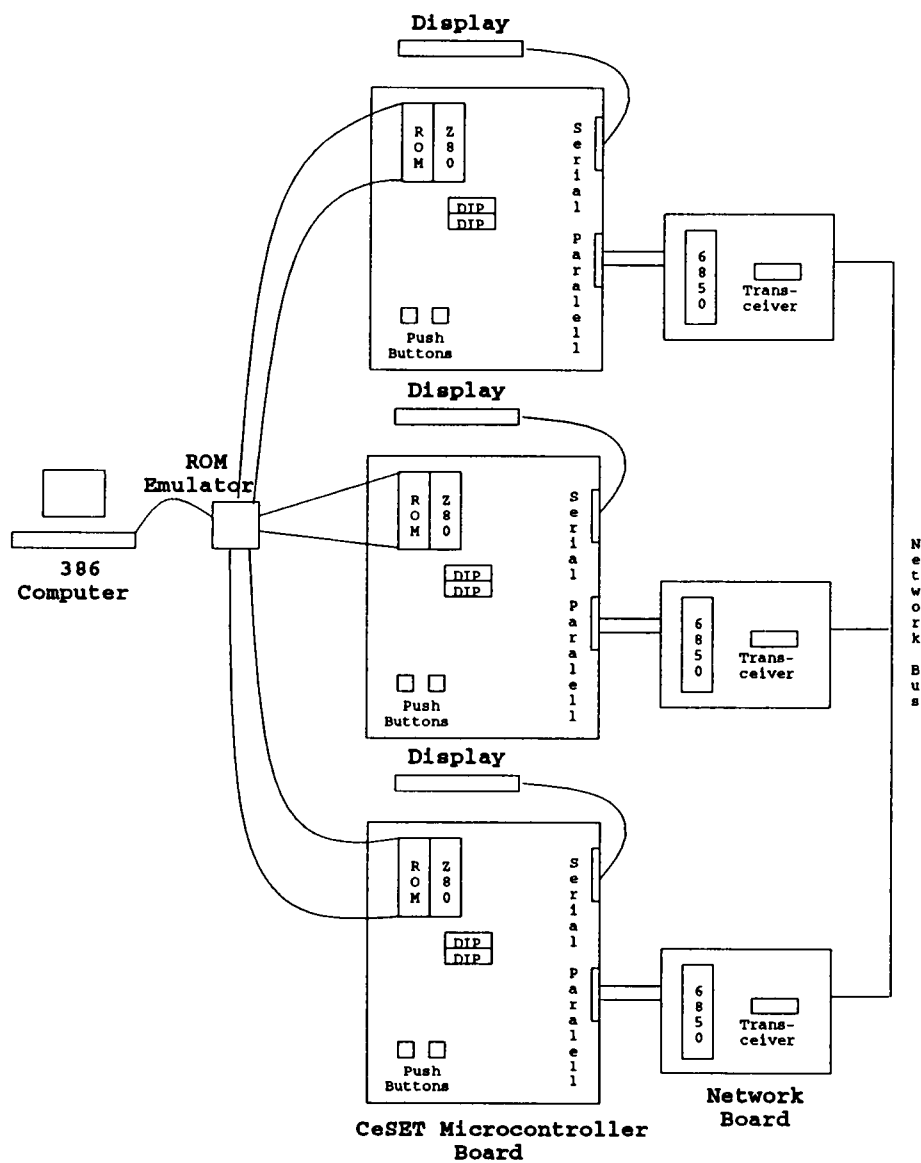


Figure 6.1: Development Hardware Configuration

6.1.1 CeSET Processor Controller and Audio Board

CPU and Memory

This microcontroller board can accommodate either a 6 MHz Zilog Z80 microprocessor or a 1.5 MHz Motorola 6809 microprocessor. The board is equipped with 56K of EPROM (Erasable Programmable Read Only Memory) and 8K of battery-backed RAM (Random Access Memory).

Switch Inputs

The use of switches is a means to allow the input of information and to have that information available to the CPU.

On the board, there are two banks of eight DIP switches, for a total of sixteen (16), and two push buttons. Eight (8) dedicated switch inputs allow a physical hookup between a switch on the game to a pin on the board. Also available are 54 matrix switches that involve a more complicated and time consuming method of reading inputs.

Input/Output

Physically the processor board has two (2) ports: serial and parallel. The serial port is used to send data to a display. The parallel port has eight (8) bits for data and 4 bits for the address. The four address bits allow for multiplexing of the port so there are logically (effectively) sixteen (16) parallel ports. Two of the ports (number 0 and 1) are used for communication to the network board

described later.

Sound

The board is equipped with a BSMT2000 sound processor.

Other Features

Other features of the board include a periodic interrupt, watchdog reset and power monitor circuits, three on-board LED's and an optional real-time clock.

6.1.2 Network Board

The network board used in development has two major components: a serial I/O chip, the Motorola 6850 ACIA, and a transceiver.

Motorola 6850 Asynchronous Communications Interface Adapter (ACIA)

- Communication of data between the Z80 on the microcontroller and the 6850 chip is in parallel. The ACIA interfaces to the Z80 with an 8-bit bidirectional bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line. These signals permit the Z80 to have complete control over the 6850.
- Data is transmitted to and received from a serial network bus, therefore; the ACIA's interface to the bus is serial.

- The 6850 provides four registers available to the Z80. The registers are used to transmit and receive data and to obtain information about current conditions or set values that affect current conditions. If access is needed to one of these registers for the purpose of reading or writing information, the register must be selected by using two input control lines.

- Transmit Data Register

A byte is transmitted onto the network bus from this register. When a new byte is received into this register from the Z80 microprocessor, it is transferred to a shift register where it is serialized and transmitted.

- Receive Data Register

A byte that is received bit-by-bit is shifted into a shift register, where it is then transferred to the Receive Data Register. It is here that the Z80 can read the new data.

- Control Register

The Control Register consists of eight bits of write-only buffer. This register allows the user to select word length, parity, number of stop bits, or other communication characteristics.

- Status Register

Information on the status of the ACIA is available to the Z80 by reading the Status Register. This is a read-only register which will give such information as the status of the Transmit Data Register, the status of

the Receive Data Register, and the status of the error logic.

Transceiver

The board also has a transceiver chip that directs the flow of information to and from the bus. In other words, it dictates whether the 6850 is using the bus for transmitting or for receiving.

A command that originates from the Z80, SET-RECV-MODE, will configure the transceiver to cause information to flow from the bus to the Receive Data Register on the I/O chip. A command that originates from the Z80, SET-XMIT-MODE, will configure the transceiver to direct the flow of information from the Transmit Data Register on the I/O chip to the bus. The bus in this emulation consists of two wires, one is tied to ground and one is the data line.

6.1.3 ROM Emulator

On the CeSET microcontroller, there is an EPROM socket for storing the code to be executed by the Z80 microprocessor. When the microcontroller is powered on, it starts running code starting at address zero (0) of the EPROM chip. It was decided to use a ROM emulator rather than the EPROM for the software development phase to accommodate the frequent changes made in the software during development. Therefore a ROM emulator was used to load the program into the ROM socket.

6.1.4 Intel 386

The code was written on an Intel 386 machine in the C language. A cross compiler was used to compile the source code into Z80 assembly language.

6.1.5 Debugging Tools

Dip Switches and Buttons

The CeSET microcontroller is equipped with some very useful tools that were utilized during the software development phase. These tools, including the dip switches and buttons already available on the microcontroller board, were used extensively in testing and debugging. One such use was to assign host addresses through the use of the dip switches during early software development, before the address initialization process had been implemented.

LED Display

In addition, for testing the output during the development, an eight-character display was connected to the serial port on each microcontroller. The display was configured to show four two-digit hexadecimal numbers. This was used, for example, to confirm that the data sent from one host was correctly received by another host. Values to be displayed were sent to the display during a periodic interrupt of the Z80.

6.2 Final Configuration

In the final configuration a network board will be manufactured combining a dedicated Z80 chip with the Motorola 6850 Input/Output device and transceiver. The final hardware configuration was emulated by the development setup as described in Section 6.1.

The CeSET processor board will send data to the network board for communication to all other hosts on the network. The network board also must send the data it has received from other stations back to its processor board.

CHAPTER 7

Software Design

7.1 Design Considerations

With the choice of a bus topology, two access protocol standards may be considered: the token bus and the CSMA/CD protocols. The token bus protocol was selected for this application primarily for two reasons. The first reason concerns the real-time requirement of the application and the second concerns the need for an equal flow of information from each station.

7.1.1 Real Time Network Performance

One basis for comparison of the token bus with CSMA/CD as an access method deals with the notion of a probabilistic versus a deterministic scheme. CSMA/CD is seen as a probabilistic scheme because an upper bound can not be identified for the time period of a station trying to access the bus. The token bus scheme is said to be deterministic because an upper bound for access time can be calculated. It is this difference that leads users to choose the token bus method in time-critical applications, because it is deterministic. On heavily loaded systems, a station is guaranteed, within a certain time, access to the bus to transmit its

information. Clearly the problem associated with a real-time system based on a CSMA/CD access method is that if a message needs to be sent within a certain time, CSMA/CD can not guarantee it.

7.1.2 Flow of Information

The application described in this thesis requires that stations transmit all updated, shared information. This information will be changing often. Because of this, the stations need to have regular turns, which is another good reason to select token passing. Also, as the amount of traffic increases, the delay in access time goes up; therefore the performance decreases.

7.1.3 Protocol Design

The Emnet protocol access scheme is decentralized. A station must possess the token in order to transmit. When it does transmit, it broadcasts not only the data packet, but the token as well. The protocol is written such that each station determines, independently, when it has access rights and when it does not. However, there are potential problems in making every station equal in status. Therefore, this protocol does implement a "master" station and a "back-up" station. These stations are not there to implement a centralized control scheme, but to respond to possible problems and to ensure the continuous operation of the network.

7.2 Software Implementation

There were three distinct parts to the software that was developed: network communication, network initialization, and a network board interface with the microcontroller.

Network communication is the most significant part of this project. It involves everything associated with passing around data smoothly and accounting for errors.

Network initialization involves the assigning of network addresses to stations. For this project, assigning addresses through hardware was not feasible, and so a way for a station to obtain an address was developed in software.

Interface with the processor board refers to the passing of data values between the microcontroller and the network board. The network board updates the microcontroller on the current data values of other stations and the processor board updates the network board on its current data values.

Memory management is also an important part of the software development and will be discussed along with these three parts.

7.2.1 Memory Management

The point of a network is to share information, and, in this case, each station wants an updated copy at all times of a set of values from all other stations on the network. Essentially, it is the idea of distributed memory. Every station will

have a local copy of shared data values from other stations, and the network is responsible for updating those local copies.

The overall goal of this network was to have a machine share certain data values (up to 15) with all other stations on the net, and, also, for that station to hold the data values of all other stations of the net that were marked as "shared". Therefore, every station is to hold a copy of shared data values that are constantly updated by the network. Each station also holds a copy of some network-related information, as shown in Figure 7.1. The last row (15) is dedicated to holding this information. These fifteen (15) bytes of data are divided into two parts.

Bytes 0 to 9 are application data. For this special application, one of the CeSET boards is designated as the lead machine for the application software. For simplicity, this machine is chosen to be the one associated with the network master. The leader is the only one that writes to these locations. These locations are used, for example, to tell all other stations to perform some function. These memory locations are at the disposal of the application programmer and are not used for network operations.

Bytes 10 to 15 are reserved for network status information. These data values are assigned only by the network board and inform the CeSET board of current conditions. Examples of data items held in this part are whether or not a machine is the master, or whether or not a machine is the only one on the network. This data is not shared among stations since these values are of specific interest only to the machine on which they reside and describe.

Figure 7.1 shows the memory allocation for any station.

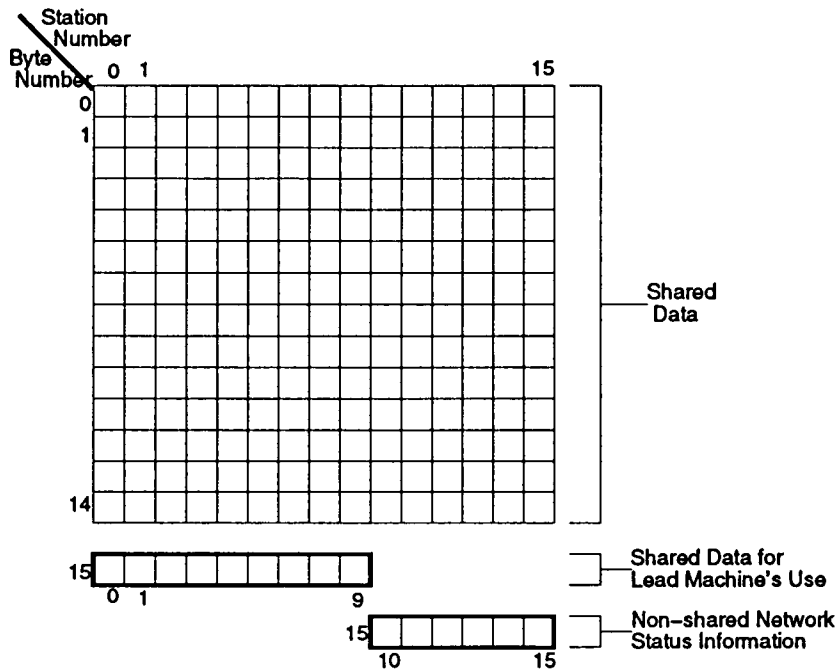


Figure 7.1: Memory Map of Network Shared Data

Table 7.1 shows the individual memory location values, for those locations in which a dedicated purpose has been assigned.

Table 7.1: Contents of Memory Locations

Memory Location	Value
[15][9]	Number of stations that are currently active
[15][13]	=1 if station is the only one up =0 if there is more than one
[15][14]	=1 if station is the master =0 if station is not the master
[15][15]	=0 if there is nothing wrong =1 if checksum bad, five times in a row

7.2.2 Initialization Software

When machines are first received by the end user, they do not have network address numbers to identify them. Therefore an initialization process must be performed.

When stations are first powered on, they are placed in an initialization mode, and they wait either for an interrupt from a button on the machine to be activated or to receive data from another station claiming an address number.

For the first machine on which the button is pushed, the number zero (0) is assigned as an address. This machine then uses the network to inform all other stations that a node with a zero (0) address exists. Zero also claims to be the "master" or "monitor" of the network. There is only one master and it is always

zero upon initialization.

For the second machine on which the button is pushed, the number one (1) is assigned as an address. That station then informs all other stations that node number one exists. Node number one also claims to be the "backup" to the master. There is only one backup and it is always one upon initialization.

Thereafter, stations are assigned a number in the order that their buttons are activated. Again, when the button is pushed, the station communicates with all other stations to announce that another number has been taken.

When every station has been assigned a number, to finish the process, the button of the master must be pushed again. The master then lets everyone else know that the initialization process is complete.

After this process, stations may be powered on and off and will still keep their identity in battery backed RAM; however the operator may reinitialize the network if so desired.

7.2.3 Network Communication

Station States

An individual station will always be in one of three states: standard, master, or backup.

1. Standard - A station in the standard state will be listening to the network.

When a packet is transmitted, the station receives this message. If it is a data packet, the information is processed. The whosenext field is also

examined. If it is not the station's turn, it simply continues to listen to the network. However, if it is the station's turn, it packages up the data it wishes to send, along with the header, transmits the packet, and then continues to listen. If the packet received was not a data packet, but a control packet, then that information is processed differently. A complete description of the control packet will be provided in a later section.

2. Master - The master performs similarly to a station in the standard state. However, the master is responsible for updating the netcontrol data part of the "global" memory, in addition to updating other stations on its station's shared memory. The master is also responsible for responding to network problems that might occur. For example, what if a station, whose turn it is, becomes inactive? How would the next station in the round-robin know that it was its turn? The answer is that the station in the master state is equipped with a timeout when listening. If this period of time goes by and the master hears nothing on the network, it assumes that a station is down. The master keeps track of which station is currently allowed access to the network and, so, after a timeout, it makes note of the station that is down and sends a control packet informing all other stations of this. Also, in that same control packet is an indication of whose turn is next.
3. Backup - The backup is there solely for the purpose of handling the situation of when the master goes down. The backup operates similarly to the master,

but has a much longer timeout. If the backup times out it assumes that the master is down. It changes state to become the master and sends a control packet to a normal-state station informing it that it must change state to become the new backup.

Software

A Network Management Table (NM Table) is used to help with network maintenance. The NM Table holds the order in which stations transmit data, and it also holds information as to which stations are active on the network and which are down. If the status of a station changes, the table that every station keeps current will have to be modified. Figure 7.2 shows the NM Table for a network where 1, 4, and 6 are the only stations that are active. The data circulates around the network from 1 to 4 to 6, and back to 1.

Station Number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D O W N	4	D O W N	D O W N	6	D O W N	1	D O W N	D O W N	D O W N	D O W N	D O W N	D O W N	D O W N	D O W N	D O W N

Figure 7.2: Network Management Table

Packet Field Description

Information on Emnet travels on the network in one of two types of packets, data and control. The packet format is identical for both types of packets, as shown in

Table 7.2. Each field is 8 bits.

Table 7.2: Packet Field Description

Source	Type	Destination	Whosenext
Data Byte 0	Data Byte 1	Data Byte 2	Data Byte 3
Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7
Data Byte 8	Data Byte 9	Data Byte 10	Data Byte 11
Data Byte 12	Data Byte 13	Data Byte 14	FF Hex
Checksum			

1. Source - The number in the source field indicates which station sent the packet.
2. Type - Type identifies whether the packet is a control packet or a data packet.
3. Destination - Since all data packets are to be read by every node, this field is irrelevant when "type" identifies the packet as a data packet. However, if the packet is a control packet, that is being used, for example, to assign a new back-up, the packet is only intended for the one node that is becoming the back-up. The destination field indicates the new back-up.
4. Whosenext - The whosenext field is used as an equivalent to passing the token in a token bus passing scheme. This field is relevant when the packet

is a data packet or a certain kind of control packet. In these cases, the station checks the number in the whosenext field. If that number matches a node's address, it assumes that the token has been passed to it, and it now has full right to send data.

5. Data Bytes 0-14 - There are 15 bytes designated for data.
6. End Of Packet (EOP) - This field indicates to the receiving node that the entire packet has been received and that the next byte is the checksum. The EOP has a value of FF hexadecimal.
7. Checksum - Checksum is the addition of all fields except the EOP field.

Control Packets

Control packets are necessary to keep the network running in the event that a problem occurs.

There are six types of control packets:

1. TABLEDOWN - A packet that contains information about altering the table to reflect the fact that a station has gone down.
2. TABLEUP - A packet that contains information about altering the table to reflect the fact that an additional station is on the network.
3. BKUPPACKET - A packet used to inform one station that it must change state to be the backup to the master. The use of this packet implies that

the previous backup has gone down or has changed states to master and is therefore accompanied by a TABLEDOWN packet.

4. RU-UP - It is possible that at some point, a station that was previously down could come back up, and although that station is listening and processing data it is not fully integrated into the network. Because the station is listed as DOWN in the table array for every station, the station is not included in the robin-robin transfer of data. To account for this possibility, the master occasionally will transmit an RU-UP packet to stations it has listed as DOWN to check on the status.
5. IAMUP - If a station that was previously down has come back up, it will not be fully integrated into the network again, until it receives a RU-UP packet. It would then respond with an IAMUP packet. The master then follows with a TABLEUP packet to let all stations on the network know about the change in the status of the network.
6. NETCNTRLDATA - In addition to a machine's data that it shares with all other machines, there are ten bytes of data that are considered network data, unspecific to one machine. This data is sent by the master, who is responsible for sending its data and the network data every time it takes its turn. In normal network operation, the master begins by transmitting the NETCNTRLDATA packet and then a data packet.

Data Packets

Most packets on the network will be data packets, which contain the data from each station that it wants all other stations to know. Data packets are initiated by each station in a round-robin fashion, and include a maximum of fifteen bytes of data.

General Network Operation

Data packets from stations are transmitted in an orderly round-robin fashion. If it is station X's turn, X sends a data packet. In the 'whosenext' field, X places the number of the station whose turn is next. X gets this number from the NM Table. If this number is Y, then station Y, after processing the data from X, transmits its own data packet.

The only deviation from this is when the master has control of the network. Before sending its data packet, the master sends a NETCNTRLDATA packet, which includes general network information, non-specific to any one station.

Figure 7.3 gives a flow diagram for basic operation. This flow diagram does not take into considerations potential network problems that are compensated by Emnet and are discussed later in this chapter.

Periodically, the master will ask to see if any of the previously DOWN stations are back up. If the master is the only one that is up on the network, the master continuously circles the net looking for a station that is back up and listening. If a station does respond, the network goes back into normal operation.

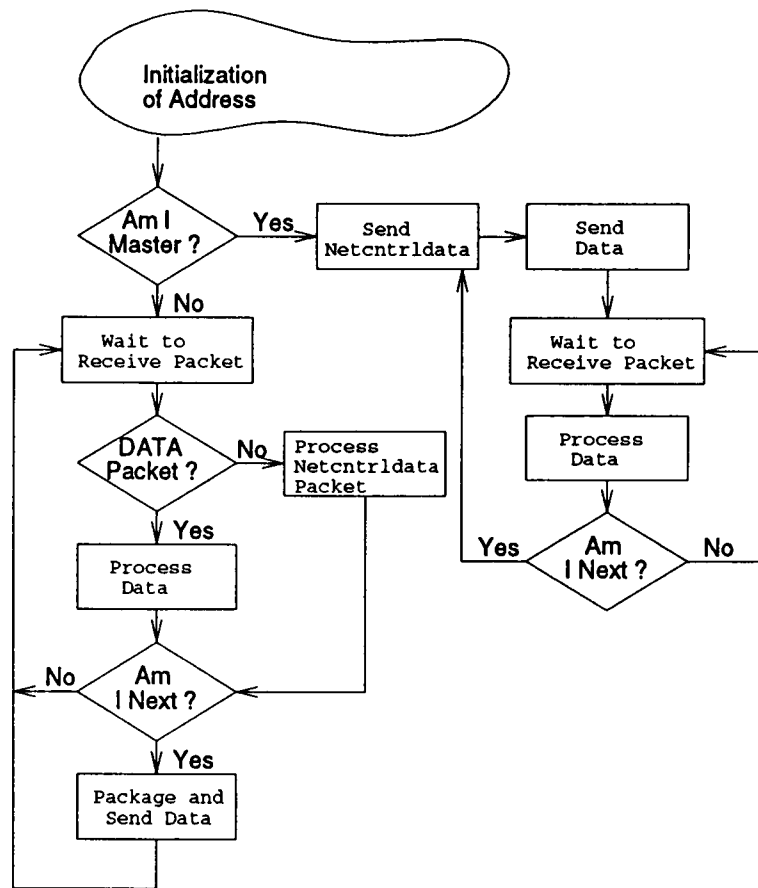


Figure 7.3: Flow Diagram for Basic Operation

Communication Problems Solved

There are many problems that can occur, however unlikely, to interrupt normal network operation. The Emnet protocol has been designed to compensate for these possible problems.

- Every packet contains a checksum. It is possible to have a transmission error and data may be received incorrectly. If the checksum included in the packet and the computed checksum on the receiving end are not the same, a station will not process the packet. If a station receives a packet in error five times in a row there may well be something wrong with that station's communication device and the gameboard can be notified by placing the appropriate value in memory location [15][15]. (See Table 7.1)
- A station may go down during the operation of the network. The only way other stations would know this is by the down station's not taking its turn. If it does not take its turn, the master will timeout, because no one will be transmitting data. The master sends a TABLEDOWN packet to the rest of the machines on the network to let them know about the change in the network.
- If the backup station is down, then the master must first assign a new backup by sending a BKUPPACKET, and then send the TABLEDOWN packet.
- If the master is down, the backup will timeout (he has a longer timeout), become master, assign a new backup and send a TABLEDOWN packet.

- If the original master (number 0) goes down, there is a new master. If number 0 comes back up later, it will assume that it is still the master. To avoid duplicate masters, when 0 comes up, it listens and if no one is transmitting, it assumes that it is the master and needs to get things going. If someone else is transmitting it assumes it has been replaced as master and changes into the standard state.
- If the original backup (number 1) goes down, there is a new backup. If number 1 comes back up later, it will assume that it is still the backup, so it will go into the backup state and when it receives an RU-UP packet it will realize that it was down, no longer the backup, and will go into the normal state.
- If a station goes down while transmitting a packet, stations will still be expecting an entire packet. The master (or backup, if master is down) in this case will complete the packet transmission. All stations will ignore this packet when received because the checksum will not compute.
- If a station receives a packet in error, i.e., the checksum does not compute, the station ignores the packet altogether. If this is a data packet, it will get the data the next time around since stations regularly transmit all their data even if there hasn't been a change in any of the data values. If the packet is a control packet and received in error then a station loses out on important information that could bring down a few stations on the network

temporarily. However, any station that is not part of the round-robin but is fully functioning and is receiving packets will eventually be back in the rotation when receiving an RU-UP packet.

- If a station transmits the hexadecimal value FF in a data field, this will be understood by the receiving station as an EOP. The subsequent computation of the checksum will be in error. However, this should not occur because the application programmer will be informed that hexadecimal FF is not a valid data value. To prevent random occurrences, the memory map of network shared data will be initially cleared on every station.

Tolerated Problems

While this software will compensate for a number of potential problems, there are scenarios that the software will not handle and, in most cases, will require that the network be restarted. The network protocol was not designed to compensate for problems considered to be either unlikely or tolerable.

- If a station goes down in the midst of transmitting a byte of the packet, and there is still power to the network board, the transceiver chip is left in transmit mode. Thus, the chip is still in transmit mode when the network resumes normal operation. Two transceiver chips will then be in transmit mode at the same time. Leaving two transceiver chips in transmit mode simultaneously for a long period of time will cause the chip to fail; however, this would be a very unlikely occurrence.

- If the master and backup go down at the same or approximately the same time, the network must be restarted.
- If the network has been previously initialized and is powered on without stations '0' and '1', the network will be inoperative. There can be no network activity until '0' and/or '1' are restarted, or the network is reinitialized to have a master or backup.
- If '1' goes down, a new backup node will be assigned. If at a later time '1' comes back up, there will temporarily be two backups on the network. This is resolved when the master sends an RU-UP packet to '1'. Number '1' then deduces it was down and that there must be a new backup on the network. So '1' changes state to the normal or standard state. If, however, the master goes down before '1' changes its state from backup, there will be two backups that timeout and try to become master. This situation is considered unlikely, and would require that the network be restarted.

7.2.4 CeSET Microcontroller and Network Board Intercommunication

When the hardware for the final version of the network board is complete, routines must exist to handle communication between the processor board board. The game board needs to share its information with its network board so the network board can send current station information. The network board needs to share the game board data coming in from all the different stations on the net. Physically

the processor board can send two bytes of data to the network board through the parallel port and visa versa. The physical connection between the two boards is shown in Figure 7.4.

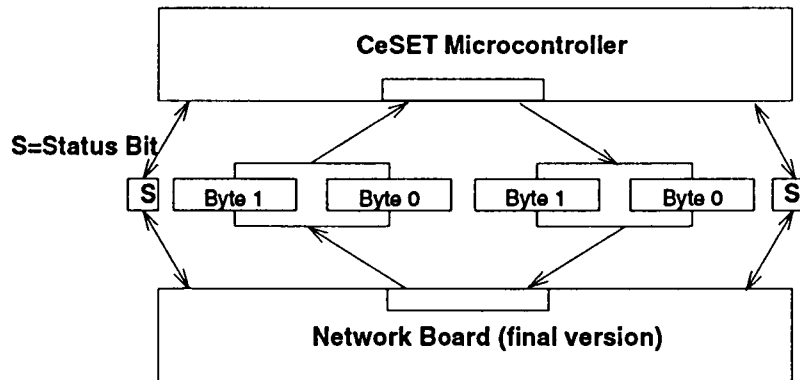


Figure 7.4: Hardware Interface Between Processor and Network Boards

The two bytes in both directions are used to specify which memory location is being updated and the value (See Table 7.3).

Table 7.3: Interface Byte Configuration

Byte	Bits 7 - 4	Bits 3 - 0
0	Game Number	Byte Number
1	Data	

There are a maximum of sixteen nodes, each having fifteen data bytes to be shared. In addition, there are ten bytes of shared data, unspecific to one machine, that the master updates.

The gameboard's interface program continually feeds the network board its fifteen bytes of game information. If it is the master, it must also give the network board some of the network information held in the sixteenth byte of the game board arrays (See Figure 7.1).

The network board regularly provides the CeSET board with application information from each of the other CeSET boards.

Either the processor board or the network board regularly calls the interface procedure. This interface procedure checks to see if the last byte of data sent to the network board has been read; if so, it sends another byte of data. The microcontroller keeps track of which byte it has just sent so it can know which byte is to be updated next. A non-master processor board updates a total of 15 bytes repeatedly. The master processor board sends a total of 25 bytes of data (15 of which are data, and 10 of which are global network data values) that the master keeps track of for the other stations.

The network board is responsible for updating fifteen (15) shared data values of all stations ($15 \times 15 = 225$), plus the network status information in memory locations [15][10] to [15][15] (see Figure 7.1), which brings the total to 231.

CHAPTER 8

Testing

A testing procedure was designed and implemented to confirm that the network was operating according to the design specifications and also to determine the network communication rate. The network operation was tested by monitoring

- the accuracy of the data values being received and stored in the Shared Memory Matrix (SMM), and
- the accuracy of the data in the Network Management Table (NMT).

The network was tested during normal operation, and, in addition, all potential problems, as discussed in Section 7.2.3, were simulated and the SMM and NMT values were observed.

In addition to verification of shared data values on the network, procedures were developed to determine the network refresh rate for these values to provide a measure of the network communication rate.

The testing procedures were developed with the goal of demonstrating the accuracy of the network operation of the three nodes in the development setup and thus to gain confidence in the robust operation of the network when expanded to the design limit of sixteen nodes.

8.1 Correctness of Shared Memory Matrix

The shared data values in the distributed memory (SMM) were verified as accurate, or current, in the following way: the DIP switches on the CeSET boards were used to select a row and column of the table of shared data values (See Figure 7.1), and the value at that location was displayed on the LED display device attached to each CeSET board. The displays were monitored simultaneously on each of the three CeSET boards, which represented Station 0, Station 1, and Station 2 in the hardware development setup. In this manner, all data values were verified as correct and current, while the network was in actual operation.

8.2 Correctness of the Network Management Table

It is essential for stable network operation that the values in the NM Table be correct and current, as this data array determines the round-robin order in which stations access the network to share their data. The NM Table also contains the list of all station numbers that are not participating in the round-robin. When stations are added or removed from the network, the NM Table must be updated in a timely manner so that the network will continue to run smoothly.

A similar method to that used for checking the Shared Memory Matrix data was used to check the values in the NM Table during network operation. DIP switches on the CeSET boards were used as input. The Table values were viewed on the displays connected to the CeSET boards (See Figure 6.1). In this way the

stable operation of the network was confirmed, both under normal operating conditions and during simulation of the various potential network problems discussed in Section 7.2.3.

8.3 Network Communication Rate

How fast are data values in the Shared Memory Matrix being updated? The communication rate was determined in the following way. A periodic interrupt is serviced by the Z80 microprocessor at a rate of 388 Hz. This interrupt was used as a counter to determine the length of time required for each station to have a turn, completing one round-robin.

The time elapsed between station's successive turns on a three node network was determined to be .106 seconds. Effectively, the time that each station takes to transmit a packet is .026 seconds because the master transmits two packets each time it has control of the network. On a fully loaded network of sixteen (16) nodes, the time elapsed between a station's successive turns is .45 seconds. This rate of communication, although relatively slow, is sufficient for the specific application for which the network was designed.

CHAPTER 9

Conclusion

Network software and hardware has been designed and implemented to meet the design criteria for an application specific Local Area Network. An inexpensive and simple solution has been devised for the network communication for up to sixteen (16) CeSET microcontroller boards. The CeSET board has immediate application for redemption games. Redemption games are ones in which a winner is awarded a number of tickets based on playing performance. These tickets are then redeemed for awards. With the network implementation, multiple game players may compete against each other, adding interest and excitement to the game.

A test facility of three CeSET nodes was built, and utilized for thorough testing, providing a high confidence level for the successful operation of a sixteen (16) node network. Potential network problems were addressed, and solutions were devised for the most common problems. For unlikely problem scenarios, it was decided not to compensate for the potential situation in order to keep the design simple and inexpensive.

It is expected that the Emnet Network board will be manufactured and marketed in the near term for use in redemption games. The Emnet Network may

be easily extended to other applications, which have a similar requirement for an inexpensive, simple-to-implement network, to share a minimal amount of data. Future development plans are to integrate the network functions, currently residing on the Network Board, directly on the CeSET Board. In addition, further work is anticipated to achieve a faster network rate.

LIST OF REFERENCES

LIST OF REFERENCES

1. Bartee, Thomas C., *Data Communications, Networks, and Systems*, SAMS, Carmel, IN, 1992.
2. Chorafas, Dimitris N., *Local Area Network Reference*, McGraw-Hill, New York, NY, 1989.
3. Currie, W. Scott, *LANs Explained*, Currie/Ellis Horwood Limited, Chichester, UK, 1989.
4. Digital Equipment Corporation, *Introduction to Local Area Networks*, Digital Equipment Corporation, 1982.
5. Fortier, Paul J., *Handbook of LAN Technology*, McGraw-Hill, New York, NY, 1989.
6. Hutchison, David, *Local Area Network Architectures*, Addison-Wesley, Cheltenham, UK, 1988.
7. Stallings, William, *Local Networks*, Macmillan Publishing, New York, NY, 1984.

VITA

Emily Elizabeth Muly was born in Needham, Massachusetts on April 24, 1969.

She attended elementary school in Pennsylvania, middle school in Florida, and high school in Connecticut and Tennessee.

She attended Northwestern University, Evanston, Illinois, and graduated in June 1991, with a B. S. degree in Computer Science from the Electrical & Computer Engineering Department, in the Northwestern McCormick School of Engineering.

She entered the University of Tennessee, Knoxville, in August 1991, to study in the Department of Computer Science, and expects to receive the M.S. Degree in May 1994.

During her graduate studies, she has been a Teaching Assistant in courses in programming and computer architecture.