AN ABSTRACT OF THE THESIS OF

Ravi C. Kumar for the degree of Master of Science in Electrical and Computer Engineering presented on June 2, 1988

Title: COLAN V: A High Performance Local Area Network for Control and Communication Utilizing a Communication Co-processor.

Redacted for privacy

Abstract approved:-----------------------------

James H. Herzog

This paper describes the design and implementation of COLAN V, a High Performance Local Area Network for Communications and Control utilizing a Communications Co-processor.

The main objective of this thesis is to improve upon the performance of previous COLANs in speed, switching techniques and error detection techniques. This was possible due to excellent capabilities of INTEL's 8344 a microcontroller with high performance serial interface unit (SIU). This SIU can operate on a concurrent basis with the core CPU which is an 8051 microcontroller.

Carrier Sense Multiple Access and Collision Detection technique (CSMA/CD) has been used by COLAN V for the bus access. COLAN V consists of nodes and a system bus. The system bus uses the EIA RS-485 standard for synchronous
communication. RS-485 has a higher data rate and a lower error rate than the conventional RS-232-C. Each node in COLAN V consists of an RS-485 circuit for bus access and RS-232-C circuit for communication with the host. Both these circuits are installed on SIBEC II single board computer.

The COLAN V software supports two interrupts. One for the host to communicate within the node and the other one is the SIU interrupt, which is for communication on the bus. Nodes in COLAN V can communicate with each other and can execute tasks for control purposes.
COLAN V: A High Performance Local Area Network
for Control and Communication Utilizing a
Communication Co-processor

by
Ravi C. Kumar

A THESIS
submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of
Master of Science

Completed June 2, 1988
Commencement June 1989
date thesis is presented: June 2, 1988

Typed by Sunitha Chilak for Ravi C. Kumar
Acknowledgements

My master's degree program at Oregon State University gave me a new direction in my life.

I would like to express my thanks to all of those who extended their cooperation in completion of my thesis and my master's program. I am grateful to Dr. G. C. Alexander who had offered me admission and financial support for my Master's studies at Oregon State University.

I thank my major professor Dr. J. H. Herzog who gave me lots of encouragement and support during my thesis work.

Special thanks are extended to Dr. Bella Bose who is my minor professor and Dr. S. J. T. Owen and Dr. E. G. Wolff for serving on my graduate committee and giving me useful advice.

I thank my parents Sri C. Varadacharyulugaru and Smt C. Lakshmi and my brothers and sisters for their support and blessings. Finally I thank my wonderful wife Mrs. C. Sunitha who has burnt midnight oil along with me and has been a constant source of inspiration during my thesis work and my master's program.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Overview of COLAN V</td>
<td>3</td>
</tr>
<tr>
<td>2.0 LAN related materials and definitions</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Protocol</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 Open Systems</td>
<td>8</td>
</tr>
<tr>
<td>2.3 The OSI Model</td>
<td>9</td>
</tr>
<tr>
<td>2.3.1 The Application Layer</td>
<td>10</td>
</tr>
<tr>
<td>2.3.2 The Presentation Layer</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 The Session Layer</td>
<td>10</td>
</tr>
<tr>
<td>2.3.4 Transport Layer</td>
<td>11</td>
</tr>
<tr>
<td>2.3.5 The Network Layer</td>
<td>11</td>
</tr>
<tr>
<td>2.3.6 The Data-Link Layer</td>
<td>12</td>
</tr>
<tr>
<td>2.3.7 The Physical Layer</td>
<td>12</td>
</tr>
<tr>
<td>2.4 IEEE 802 Standards for LAN</td>
<td>13</td>
</tr>
<tr>
<td>2.5 IEEE 802 Token Bus</td>
<td>13</td>
</tr>
<tr>
<td>2.6 IEEE 802 Token Ring</td>
<td>14</td>
</tr>
<tr>
<td>2.7 IEEE 802.3 Standard (CSMA/CD)</td>
<td>15</td>
</tr>
<tr>
<td>2.8 Asynchronous Mode</td>
<td>16</td>
</tr>
<tr>
<td>2.9 Synchronous Mode</td>
<td>17</td>
</tr>
<tr>
<td>2.9.1 Externally Clocked mode</td>
<td>17</td>
</tr>
<tr>
<td>2.9.2 Self Clocked mode</td>
<td>18</td>
</tr>
<tr>
<td>2.10 The communication subnet</td>
<td>19</td>
</tr>
<tr>
<td>3.0 COLAN V SYSTEM</td>
<td>23</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Overview of TASKMASTER</td>
<td>23</td>
</tr>
<tr>
<td>3.2.1 Weaknesses of TASKMASTER</td>
<td>25</td>
</tr>
<tr>
<td>3.2.2 Contributions of COLAN II</td>
<td>26</td>
</tr>
<tr>
<td>3.2.3 Contributions of COLAN III</td>
<td>26</td>
</tr>
<tr>
<td>3.2.4 Contributions of COLAN IV</td>
<td>27</td>
</tr>
<tr>
<td>3.3 Overview of COLAN V</td>
<td>28</td>
</tr>
<tr>
<td>3.3.1 Packet Switching</td>
<td>29</td>
</tr>
<tr>
<td>3.3.2 Error Detection Technique</td>
<td>29</td>
</tr>
<tr>
<td>3.3.3 Concurrency of Operation</td>
<td>30</td>
</tr>
<tr>
<td>3.3.4 Speed of Transmission</td>
<td>30</td>
</tr>
<tr>
<td>4.0 Hardware Of COLAN V System</td>
<td>35</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>35</td>
</tr>
<tr>
<td>4.2 Intel’s 8044 Architecture</td>
<td>35</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>21</td>
</tr>
<tr>
<td>2.2</td>
<td>22</td>
</tr>
<tr>
<td>3.1</td>
<td>31</td>
</tr>
<tr>
<td>3.2</td>
<td>32</td>
</tr>
<tr>
<td>3.3</td>
<td>33</td>
</tr>
<tr>
<td>3.4</td>
<td>34</td>
</tr>
<tr>
<td>4.1</td>
<td>45</td>
</tr>
<tr>
<td>4.2</td>
<td>46</td>
</tr>
<tr>
<td>4.3</td>
<td>47</td>
</tr>
<tr>
<td>4.4</td>
<td>48</td>
</tr>
<tr>
<td>4.5</td>
<td>49</td>
</tr>
<tr>
<td>4.6</td>
<td>49</td>
</tr>
<tr>
<td>5.1</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>58</td>
</tr>
<tr>
<td>5.2.1</td>
<td>59</td>
</tr>
<tr>
<td>5.2.2</td>
<td>60</td>
</tr>
<tr>
<td>5.3</td>
<td>61</td>
</tr>
<tr>
<td>5.3.1</td>
<td>62</td>
</tr>
<tr>
<td>5.3.2</td>
<td>63</td>
</tr>
<tr>
<td>5.3.3</td>
<td>64</td>
</tr>
<tr>
<td>5.4</td>
<td>65</td>
</tr>
</tbody>
</table>
COLAN V: A HIGH PERFORMANCE LOCAL AREA NETWORK FOR
CONTROL AND COMMUNICATION UTILIZING A COMMUNICATION
CO-PROCESSOR.

CHAPTER 1

INTRODUCTION

1.1 Motivation:

The purpose of this research was to implement a high
speed COLAN with superior error detection and packet
management techniques. This was possible due to excellent
capabilities of INTEL 8344, a microcontroller with an on-
chip communication co-processor.

1.2 Background

As the technology is advancing the demand for high
performance local area networks is increasing at a rapid
rate. Many organizations already have a substantial number
of computers in operation, often located far apart. For
instance, a company with many factories may have a computers
at multiple locations to keep track of inventories, monitor
productivity, do the payroll and so on. Initially each of
these computers may have worked in isolation, but at a
certain time, management may have decided to extract and
correlate information about the entire company. Therefore a
current philosophy is that resources should be available to
anyone on the network without regard to the physical
location of the resource and the user.
With unconnected computers, if a machine goes down due to hardware failure, the user has to either wait till his computer is repaired or go elsewhere to work on another working computer. With a network, the temporary loss of a single computer is much less serious. In industrial process control, banking, military, and other real time applications, a complete loss of computing power for even a few hours due to some problems could be disastrous.

Thus we see a reason to define a local network at this point [STAL 84].

A local network is a communications network that provides interconnection of a variety of data communicating devices within a small area.

The goal of this work is to implement a reliable local area network for control and communications with the abovementioned advantages. In our case we chose IBM PC’s as our host computers.

Another aspect of this thesis is to implement the abovementioned local area network at a very economical price using commonly available SIBEC boards. SIBEC boards are custom built printed circuit boards. SIBEC boards will be more fully described in Chapter 4.

In our definition of local area networks we have mentioned "variety of data communication devices". This provided impetus for establishment of industry standards. Industries wanted "open systems" which allow end users to purchase equipment from several vendors and integrate them
for an efficient solution for a given application. Thus ISO (International Standards Organization) established the open systems Interconnect (OSI) reference model which will be more fully developed in Chapter 3. Similarly the Institute of Electrical and Electronics Engineers (IEEE) formed IEEE 802 Standards Committee with regards to LAN standardization. Description of these standards is done in the later chapters. This project utilizes a Carrier Sense Multiple Access and Collision Detection CSMA/CD technique for media access and network topology is BUS.

1.3 Overview of COLAN V:

Considerable amount of research has been done on implementation of a low cost and high performance LAN for control and communication purposes in the Electrical and Computer Engineering department at Oregon State University.

To begin with, modification of the daisy chain structured TASKMASTER system [HERZ 87] into a LAN was attempted by Y.P. Zheng [ZHEN 86]. It was called COLAN (a Control Oriented Local Area Network). It used hybrid medium access method of token passing and CSMA/CD. It was not fully implemented. Later COLAN II was developed by S.K. Kao [KAO 87] who used token passing for medium access. COLAN III was implemented by D.H. Eum [EUM 87]. He used CSMA/CD scheme for bus access. COLAN IV was developed by Yong Thye [Thye 88] who also used CSMA/CD scheme and developed the system to the fullest extent including a powerful user interface.
All the above methodologies used an asynchronous mode of communication. This involved byte-by-byte transfer of information between hosts. This results in relatively low speeds of communication. This project, COLAN V, is a first trial in implementing COLAN using a synchronous mode of communication. There are several advantages to using synchronous type of communications. COLAN V utilizes the Intel 8344 microcontroller with an integrated high performance communications co-processor.

Networking capability has been added to this microcontroller which improved the performance of the communications system. The 8344 reduces the cost of networking microcontrollers without compromising performance. It contains all of the hardware components necessary to implement a microcomputer system with communications capability, plus it reduces the CPU and software overhead of implementing the HDLC/SDLC communication protocols. Figure 1 shows a functional block diagram of the 8344.

The 8344 integrates the powerful 8051 microcontroller with an intelligent Serial Interface Unit (SIU) to provide a single chip solution which efficiently implements a distributed processing or distributed control system. The microcontroller is a self sufficient unit containing ROM, RAM, ALU and its own peripherals. The 8344's architecture and instruction set are identical to the 8051's. The serial Interface Unit (SIU) uses bit synchronous HDLC/SDLC protocol
and can communicate at bit rates up to 2.4 Mbps, externally clocked, or up to 375 Kbps using the on-chip digital phase locked loop. The SIU contains its own processor, which operates concurrently with the microcontroller.

The CPU and the SIU, in the 8344, interface through 192 bytes of dual port RAM. There is no hardware arbitration in the dual port RAM. Both processor's memory access cycles are interlaced; each processor has access to alternate clock cycle. Therefore, there is no throughput loss in either processor as a result of the dual port RAM, and execution times are deterministic. Since this has always been the method for memory access on the 8051 microcontroller, 8051 programs have the same execution time in the 8344.

This chapter served to introduce COLAN V, explaining its background as a control oriented LAN and summarizing its objectives and most important features. Chapter 2 discusses, in brief, the LAN related materials and definitions. Chapter 3 discusses the COLAN V system as an improvement over previous COLAN's. Chapter 4 discusses the hardware structure of COLAN V. Chapter 5 discusses the software implementation of COLAN V. Chapter 6 discusses in brief about the suggestions for future improvements of COLAN V.

Summary:

The main objectives of this thesis are summarised below.
1) Build a high performance, low cost Local Area Network utilizing CSMA/CD protocol.

2) Implement synchronous mode of communication.

3) Increase the speed of operation to 300 Kbits/sec.

4) use packet switching technique for data transfer.

5) Improve error detection techniques using CRC-CCITT polynomial available on INTEL 8344 microcontroller.
Figure 1.0
Simplified 8044 Block Diagram
CHAPTER 2

LAN RELATED MATERIALS AND DEFINITIONS

2.1 Introduction

Various terminologies and concepts have been introduced in chapter 1 regarding the implementation of a local area network. These are discussed below.

2.2 Protocol

The communications functions are partitioned into a hierarchical set of layers. Each layer performs a related subset of functions required to communicate with another system. Layer n of one machine carries on a virtual conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer n protocol.

Physical or actual communication takes place at the lowest level through twisted cable or optic fibre etc. In the remaining higher levels, no actual communication takes place. Each layer, instead, passes data and control information to the layer immediately below it until the lowest layer, i.e. the physical layer, is reached.

2.2 Open Systems

Implementation of a LAN in different environments such as factory, office etc., has created the problem of incompatibility among various communicating devices. This has given rise to the definition of "Open Systems". Open
systems allow end users to purchase equipment from several vendors in order to realize an optimal solution for given application. In an effort to encourage open networks, the International Standard Organization (ISO) [ZIMM 80] developed the Open Systems Interconnect (OSI) Reference Model.

The Institute of Electrical and Electronics Engineers (IEEE), with regards to a need for standardization in the area of LANs, formed the IEEE 802 standards committee. These are discussed briefly in the following sections.

2.3 THE OSI MODEL

The design of an open system has proven to be a very difficult task because of (1) all the manufacturers and standards organizations involved and (2) the size of the problem.

A local network must be thought of as a single entity. The system has many parts, but they all interrelate. The OSI model provides names for the parts of a communications network. It doesn't matter in a general sense if the network is local or international. The size of the network may limit or increase the importance of certain of its parts, but they will all be there in one form or another. The OSI model uses specific definitions to describe the various portions of the network. It speaks of "layers" of functions arranged in a hierarchy. Each layer performs a related subset of functions required to communicate with another system. It relies on the next lower layer to
perform more primitive functions and to conceal the details of those functions. The task of the ISO subcommittee was to define a set of layers and the services performed by each layer. Various layers are described as follows with reference to Figure 2.1

2.3.1 The Application Layer

The OSI model has seven functional layers. The top function, is the end-user application. This includes applications that are to be run in a distributed environment. It would typically include vendor-provided programs of general utility, such as electronic mail, a transaction server, a file transfer protocol, and a job manipulation protocol.

2.3.2 The Presentation Layer

The next down in the model (layer 6) is the presentation function. The presentation layer prepares the information for the application. Services that this layer would typically provide include Data translation, Formatting and Syntax selection. Examples of presentation protocols are text compression, encryption, and virtual terminal protocol. A virtual terminal protocol converts between specific terminal characteristics and generic or virtual models used by application programs.

2.3.3 The Session Layer

The Session Layer (layers 5) is a coordinating function. It establishes the communications link between units and gradually feeds or buffers information to the
devices or program performing the presentation function. At a minimum, the session layer provides a means for two presentation entities to establish and use a connection, called a session.

2.3.4 Transport Layer

The Transport Layer (layer 4) functions to provide a common face to the communications network. It translates whatever unique requirements the other higher layers might have into something the network can understand. It also makes the most effective use of whatever various communications media it is connected to and selects the best or most logical route for transmission. It detects and corrects errors in transmission and provides for the expedited delivery of priority messages. It checks the data, puts it into the proper order if it arrived in incorrect sequence, and usually sends an acknowledgment back to the originating transport layer. It attempts to reestablish contact in the event of a network failure.

2.3.5 The Network Layer

The network layer (layer 3) sets up a logical transmission path through a switched network. In local networks this path may only be theoretical, since the individual units are almost always electrically connected into the circuit and the paths are defined by the network topology. But in large systems, several transmission paths and even alternative media (dialled telephone service versus leased service, for example) may exist. The transmission
path may be temporary in nature lasting only long enough to transfer a packet of information or it may provide continuous connection for two users of the network.

In a local network, the network control function can exist in one place (star network) or be distributed (bus or ring.) It is essentially made up of software that recognizes various conditions on the network and reacts to them.

2.3.6 The Data-Link Layer

The data-link layer (layer 2) does the accounting and traffic-control chores needed to transfer information on an electrical link. It puts every piece of information into the right place and checks it out before putting it on the bus. Information is broken down into packets. Then frames are prepared by attaching address bytes, control bytes, frame acknowledgements and frame check sequences for error detection to these data packets. Then it properly routes the data within the receiving device.

The data link was originally thought of as being a function of the software in a device. It is increasingly being done by special-purpose integrated circuits requiring little external programming. A fitting example would be Intel's 8344 with a communications coprocessor which is described in Chapter 4.

2.3.7 The Physical Layer

The physical layer is concerned with transmission of a unstructured bit stream over the physical link. It involves
such parameters as signal voltage swing and bit duration. It also deals with other mechanical, electrical, and procedural characteristics to establish, maintain, and deactivate the physical link (RS-232-C, RS-449, X.21).

2.4 IEEE 802 Standards for LAN:

The ISO open systems interconnect model has been adopted by the IEEE Standards Committee to define the specifications for the data link layer and the physical layer. In response to a need for standardization, the IEEE 802 Standards Committee drafted the specifications of a LAN based on widely accepted industry standards. The IEEE 802 Standards (IEEE 82a, 82b, 82c) classify LANs according to their network topologies and media access methods.

The IEEE 802 Standards for LANs consists of five parts: 802.1, 802.2, 802.3, 802.4, and 802.5. As depicted in Figure 2.2, IEEE 802.1 describes how each part of the 802 Standards fits into the OSI Reference Model. The IEEE 802.2 standard describes the functions and protocols of the logical link control sublayer in the LAN.

2.5 IEEE 802 Token Bus:

The Token Bus technique is more complicated than CSMA/CD. For this technique, the stations on the bus or tree form a logical ring; that is, the stations are assigned logical positions in an ordered sequence, with the last member of the sequence followed by the first. Each station knows the identity of the stations preceding and following
it. The physical ordering of the stations on the bus is irrelevant and independent of the logical ordering.

A control frame, known as the token, regulates the right of access. This token includes a destination address and stations receiving the token, control the bus for a specified time. The control station may transmit frames to other stations, poll other stations, and get responses. When these actions have been completed or time has expired, the control station must pass the token frame on to the next station in the logical sequence. That station then assumes temporary control of the bus. Two phases are required: one for data transfer and another for token transfer.

The main disadvantage of Token Bus is its complexity. The second disadvantage is the overhead involved. Under lightly loaded conditions, a station may have to wait through many fruitless token passes for a turn.

2.6 IEEE 802 Token Ring:

The Token Ring technique is based on the use of a single token that circulates around the ring when all stations are idle. A typical example of a token is an 8-bit pattern such as '01111111'. A station wishing to transmit must wait until it detects a token passing by. It then changes the token from "free token" to "busy token." This can be done by changing the last bit of the token (e.g., from '01111111' to '01111110'). The station then transmits a data frame immediately following the busy token.
There is now no free token on the ring, so other stations wishing to transmit must wait. The frame on the ring will make a round trip and be purged by the transmitting station. The transmitting station will insert a new free token on the ring when both of the following conditions have been met:

* The station has completed transmission of its frame.
* The busy-token has returned to the station.

The principal disadvantage of Token Ring is the requirement for token maintenance. Loss of the free token prevents further utilization of the ring. Duplication of the token can also disrupt ring operation. One station must be elected monitor to assure that exactly one token is on the ring and to reinsert a free token if necessary.

2.7 IEEE 802.3 Standard (CSMA/CD)

The Carrier Sense Multiple Access with Collision Detection (CSMA/CD) media access method is the means by which two or more stations share a common bus transmission medium. To transmit, a station waits (defers) for a quiet period on the medium (that is, no other station is transmitting) and then sends the intended message in bit-serial form. If after initiating a transmission, the message collides with that of another station, then each transmitting station intentionally sends a few additional bytes to ensure propagation of the collision throughout the system. The station remains silent for a random amount of time (backoff) before attempting to transmit again (IEEE 82
al. Each aspect of this media access method is specified in details in IEEE 802.3 Standards.

The reasons for choosing CSMA/CD protocol are
* Simple algorithm
* Widely used
* Fair access
* Good performance at low to medium load.

2.8 Asynchronous Mode:

In asynchronous communication neither parties in a communication net can predict when the next bit will arrive. Therefore, an alert bit called the start bit is transmitted immediately before any character is transmitted. The start bit is a logical 0, or space, and is represented by the positive voltage level on the RS-232-C direct current line. The start bit tells the receiving system to look at the next bits as an ASCII character. An optional parity bit could be added as an eighth bit for error detection and correction. After the seven bits of ASCII character and the parity bit are sent, one or more stop bits (logic 1) are transmitted. These stop bits ensure that the receiver recognizes the next start bit, and the whole process starts all over again, many times a second. The rate of transmission determines how many stop bits are transmitted. Two stop bits are usually used at 110 baud and below, one during faster transmission.

Asynchronous mode is commonly used in local connections between computer and terminal equipment. Using RS-232-C
so far in previous COLANs a maximum speed of 9600 baud has been achieved between node and host computer. Further 50 feet of cable length is a standard for reliable communications.

2.9 Synchronous mode:

In synchronous communication, sender and receiver synchronize their clocks and expect bit changes only when the clock ticks. Hence data arrives at specified times. Synchronous mode of operation is faster and more efficient than asynchronous mode. It requires more precise timing and is better suited to high speed transmission systems handling thousands of characters per second. The possible data rates in this mode are discussed in the following sections. To prevent timing drift between the transmitter and the receiver, their clocks must somehow be synchronized. This can be done in two ways as discussed below.

2.9.1 Externally Clocked mode:

This mode has been discussed with reference to INTEL's 8044 microcontroller. In the externally clocked mode, a common Serial Data Clock (SCLK on pin 15) synchronizes the serial bit stream. This clock signal may come from the master CPU or primary station, or from an external phase-locked loop local to the 8044. Figure 3.3 illustrates the timing relationships for the serial interface signals when the externally clocked mode is used in point-to-point and multipoint data link configurations.
Incoming data is sampled at the rising edge of SCLK, and outgoing data is shifted out at the falling edge of SCLK.

2.9.2 Self Clocked mode:

This mode is also discussed with reference to INTEL's 8044 microcontroller. The self clocked mode allows data transfer without a common system data clock. Using an on-chip DPLL (digital phase locked loop) the serial interface recovers the data clock from the data stream itself. The DPLL requires a reference clock equal to either 16 times or 32 times the data rate. This reference clock may be externally supplied or internally generated. When the serial interface generates this clock internally, it uses either the 8044's internal logic clock (half the crystal frequency) or the "timer 1" overflow. Figure 3.4 shows the serial interface signal timing relationships for the loop configuration when the unclocked mode is used.

The DPLL monitors the received data in order to derive a data clock that is centered on the received bits. Centering is achieved by detecting all transitions of the received data and then adjusting the clock transition (in increments of 1/16 bit period) toward the center of the received bit. The DPLL converges to the nominal bit center within eight bit transitions, worst case.

To aid in the phase locked loop capture process, the 8044 has a NRZI (non-return-to-zero inverted) data encoding and decoding option. NRZI coding specifies that a signal
does not change state for a transmitted binary 1, but does change state for a binary 0. Using the NRZI coding with zero-bit insertion it can be guaranteed that an active signal line undergoes a transition at least every six bit times.

2.10 The communication subnet:

With characteristics such as simplicity, high speed, low transmission delay, the wide range of application areas, and use of real time control operations, the LAN protocol design philosophy is different from that of conventional long-haul networks.

The service layer for a typical LAN might include the following (FRAN 811):

1) Level 1, the physical layer, is concerned with transmission of the unstructured bit-stream over the physical link, specifying the electrical and functional characteristics, such as signal voltage level and bit-duration, necessary to interface an element to a network interface unit and a network interface unit to the network channel.

2) Level 2, the access layer, specifies the mechanisms for network interface unit access to the channel, message format, transmission rules for outgoing message channel-monitoring and message-receipt rules, received message error detection rules, and transmission acknowledgement rules.

3) Level 3, a combination of levels 2 and 3, specifies
addressing information rules for outgoing messages, monitoring and regulation rules to control the flow of messages to nodes, and rules to initiate and terminate extended-message transfer sequences.

Although this layering is typical, it is not definitive since a standard for a wide variety of LANs does not yet exist. One of the main functions of level 3 is routing. With a direct link available between any two points, this is not required. Its other functions addressing, sequencing, flow control, and error control are also performed by layer 2. Therefore, though a network may provide services through level 3, the protocol can be implemented with two OSI levels, dependent upon LAN characteristics, such as the IEEE 802 standard [IEEE 82a, 82b, 82c] for the CSMA/CD Bus, the Token Bus, and the Token Ring protocol.

The first three layers of ISO OSI Reference model are concerned with the communication subnet. Since this project mainly deals with the design of the subnet, its attention is confined to only the first three layers.

In this project IBM PCs have been chosen as hosts. SIBEC II boards with INTEL's 8344 (microcontroller with an on-chip communications co-processor) have been chosen as nodes. CSMA/CD has been used for bus access.
Figure 2.1

Open Systems Interconnection Model
Figure 2.2 IEEE 802 Standards
CHAPTER 3
COLAN V SYSTEM

3.1 Introduction
This chapter discusses COLAN V and its improvements over previous COLANs.

3.2 Overview of TASKMASTER:
The COLAN II was based upon a daisy-chain structured system, called TASKMASTER. This section briefly reviews the TASKMASTER System.

The TASKMASTER System consists of one host and several microcontroller based boards called TASKMASTERs. The host is a master and the TASKMASTER boards are intelligent slaves capable of performing a variety of specified control tasks. The system is configured as a daisy-chain structure as shown in Figure 3.1.

The TASKMASTER system is a control oriented distributed system designed for real-time control purposes. Each control action is accomplished through the use of system tasks. Each task has a unique "name" or "task number" which can be specified by the host. Tasks reside in the TASKMASTER's program memory in the form of task library. Such a task library can be modified by users for specific applications.

Each TASKMASTER in the system has a unique device number. A command from the host must specify the device number, task number, and task types.

There are three types of tasks:
**Immediate tasks:** These are tasks of the highest priority. Upon receiving an immediate task, the TASKMASTER executes it immediately without delay.

**Queued tasks:** These tasks are to be put on the task queue, and be executed sequentially.

**Synchronized tasks:** These tasks are put on the task queue and be executed only if: a) It has moved to the head of the task queue, and b) A synchronizing signal from the host has been received by this TASKMASTER.

The command packet is in the following format:

```
(DN P TN Q DD/)
```

{ : Starting delimiter of the packet.

DN: Device number

P: Pre-execution control character. It indicates if the task called is an immediate task, a queued task, or a synchronized task.

Immediate task

Queued task

Synchronized task

TN: Task number

Q: Post execution control character. It indicates if the task should be requeued.

Non-requeued task

Requeued task

DD: Data field. This data field can contain up to 5 pairs of hexadecimal characters.
The operating system of the TASKMASTER system consists of two major parts, the main routine and the interrupt service routine. The flow chart is shown in Figure 3.2.

1) The main routine

The purpose of this routine is to initialize the TASKMASTER, open an interrupt window for the serial port interrupt, execute the queued task at the head of task queue, and manage the task queue.

2) Serial interrupt service routine

The purpose of this part is to receive a character from the host, process the command packet upon receiving a complete packet, put the modified packet on the task queue if a queued or synchronized task, or execute it if an immediate task.

3.2.1 Weaknesses of TASKMASTER:

TASKMASTER system was originally designed to operate with a master-slave structure utilizing a daisy-chain connection. This is an undesirable control structure for distributed control system. Also the capabilities of TASKMASTER's operating system are quite limited. It also had low speed of operation, daisy chained structure and restricted location of the host.
3.2.2 Contributions of COLAN II:

COLAN II employed a bus interconnection structure and a token passing method to control the access to the bus. Major emphasis of the work has been done on the system structure and the operating system. The nodes in COLAN II communicated with one another using one of the three packets: command packets, message packets, and text/data packets. The software also supported node level application programs suitable for control purposes.

3.2.3 Contributions of COLAN III:

COLAN III is a control oriented LAN designed to achieve reasonable network performance and flexibility at low cost. COLAN III used Carrier Sense Multiple Access and Collision Detection (CSMA/CD) protocol to control the bus access. This was less complex in design compared to COLAN. The greatest improvement from TASKMASTER SYSTEM to COLAN III is that the number of possible system schedulers is increased from just a single scheduler to equal the number of nodes on the network bus. In other words, any node can be the system scheduler, making higher level decisions concerning control activities. The scheduler node transmits commands to skilled or specialized TASKMASTERS for coordinated action.

In addition to control-oriented activities, COLAN III also supports mailing activities, including file transfers and direct message transfers.
3.2.4 Contributions of COLAN IV:

COLAN IV was designed to operate under light loads. COLAN IV adopted CSMA/CD media access method. It was designed for dual purposes i.e. communications and control. As such, it is functionally flexible. It may either function as a pure communications network, or as a pure control system, or as a mix of both.

As a communications network, the services provided by COLAN IV were

1) Electronic mail
2) Resource sharing
3) Network bus monitoring

Electronic Mail was the major application of COLAN IV in its capacity as a communications network. Electronic mail provided personal mail boxes in all network nodes.

Resource Sharing was another service provided to all network users. Printer sharing has been implemented. To reduce cost, a single shared printer may be utilized to service the printing needs of the entire Local Area Network.

Network Bus Monitoring was another function provided. It’s a passive function that captures all the data traffic passing through the network bus. It’s useful for network management and maintenance.

COLAN IV was also a dual purpose LAN. It also served as a real-time control system.
3.3 Overview of COLAN V:

The physical configuration of COLAN V consists of several nodes linked into a local area network by a shared communication medium. Each node has two components, a PC and a Network Interface Unit (NIU). Nodes are connected to the shared bus (RS-485). Communications within a node, i.e. between the PC and the NIU, utilize the common asynchronous technique of RS-232-C. The standard serial port, COM 1, of the IBM PC provides a full-duplex, serial interface with the NIU at the data rate of 9600 baud. Communications among network nodes via their shared communication medium are carried out with a half-duplex, synchronous, serial, baseband transmission of up to 300 Kbits/sec.

Twisted-pair cable serves as the shared communication medium which interconnects all the nodes with a linear "BUS" network topology. The BUS topology provides a simple but effective way of realizing a decentralized control structure of a LAN. The RS-485 bus standard is adopted to achieve a high performance with a low cost twisted-pair cable. It supports data rates up to 10 Mbps with a 50 ft cable, or 100 Kbps with a 4000 ft cable.

Designed to operate under light loads, COLAN V adopts the CSMA/CD media access method. The CSMA/CD technique outperforms both TOKEN BUS and TOKEN RING techniques in low load environments [HAMM 86]. Furthermore, no complex software algorithm nor complicated hardware is needed to implement the CSMA/CD technique. This simplicity is a
favorable feature which makes the network highly reliable and maintainable.

COLAN V is designed for the dual purpose of communications and control. Improvements of COLAN V over previous COLANs has been discussed in the following sections. Details of COLAN V are discussed in the succeeding chapters.

3.3.1 Packet Switching:

So far, previous COLANs have utilized a message switching technique for file transfers. A node wishing to transfer a long file transmits the entire file in a single packet, i.e. a complete message. COLAN V adopts packet switching technique in which data transactions are carried out among the nodes by the breaking down the message into small packets of 50 bytes each. This was done to split up the internal data RAM into two parts i.e. 50 bytes reserved for receiving data from the network and another 50 bytes reserved for data received from the host (IBM PC). Other 92 bytes have been reserved for flags and as general purpose memory locations.

3.3.2 Error Detection Technique:

A simple checksum technique for error detection was adopted by previous COLANs. This leads to less reliable data transmission among the nodes. COLAN V utilizes Cyclic Redundancy Code (CRC) technique for data error detection. This CRC is based on [ X^16+X^15+X^2+1 ] polynomial. Intel 8344's Serial Interface Unit (SIU) manipulates this CRC in
hardware. It automatically calculates the CRC for the transmitting data and inserts it into the transmitting frame [INTE 86]. In the receiving end the SIU again checks for the correct CRC. If the CRC does not match, the SIU does not interrupt the CPU and hence the frame received will be ignored by the receiver.

3.3.3 Concurrency of Operation:

In previous COLANs a single processor was required to service both task management and communication processes. In COLAN V the serial interface unit handles all the communication overhead by itself. The CPU is interrupted when a correct frame is received and transmitted. The SIU operates concurrently with the core CPU of 8344 which is an 8051. Therefore the CPU time could be utilized in better fashion and efficiently in non-communication jobs.

3.3.4 Speed of Transmission:

COLAN V has been implemented utilizing the flexible mode [Chapter 5] of operation of the 8044 [INTE 86]. All nodes are peers and have equal opportunity to capture the bus for communications. Using this flexible mode, speeds of data transfer up to 300K bits/sec have been achieved. Previous COLANs had a maximum data rate of 1200 baud over the bus.
Figure 3.1 TASKMASTER SYSTEM Configuration

HERZ 87
Figure 3.2 TASKMASTER Operating System Flowchart
HERZ 87
Figure 3.3 Serial Interface Timing-Clocked Mode
Figure 3.4 Serial Interface Timing—Self Clocked Mode
CHAPTER 4

HARDWARE OF COLAN V SYSTEM

4.1 Introduction:

This chapter briefly describes the hardware involved in the design of the COLAN V system.

4.2 Intel's 8044 Architecture:

The 8044 is based on the 8051 core. The 8044 replaces the 8051's serial port with an intelligent HDLC/SDLC controller called the Serial Interface or SIU. Thus the differences between the two result from the 8044's increased on-chip RAM (192 bytes) and additional special function registers necessary to control the SIU (Figure 1.0). Aside from the increased memory, the SIU itself, and differences in 5 pins (for the serial port), the 8044 and 8051 are compatible [INTE 86].

4.3 Serial Interface Unit (SIU):

This section describes the circuit used for connection with SIU and describes in brief about the SIU itself.

As shown in the Figure 4.2 the RTS (request to send) and CTS (clear to send) have been shorted. This was done because a four wire configuration for the bus was not used. Instead a RS-485 differential bus was used for transmission/reception over the network. Pin P3.0 and pin P3.1 were used in nonloop mode. P3.0 was used as I/O for data direction control and P3.1 was used for actual data input and output. Note that P3.0 the I/O pin has been used
to drive the 75174 & 75175 (bus drivers/receivers) which buffer the outgoing and incoming data on P3.1.

The SIU divides functionally into two sections a bit processor (BIP) and a byte processor (BYP). They share some common timing and control logic. As shown in Figure 4.3, the BIP operates between the serial port pins and the SIU bus, and performs all functions necessary to transmit/receive a byte of data to/from the serial data stream. These operations include shifting, NRZI encoding, zero insertion/deletion, and FCS generation/checking. The BYP manipulates bytes of data to perform message formatting, and other transmitting and receiving functions. It operates between the SIU bus (SIB) and the 8344's internal bus (IB). The interface between the SIU and the CPU involves an interrupt and some locations in on-chip RAM space which are managed by the BYP.

The maximum possible data rate for the serial port is limited to 1/2 the internal clock rate. This limit is imposed by both the maximum rate of DMA to the on-chip RAM, and by the requirements of synchronizing to an external clock. The internal clock rate for an 8044 running on a 12 MHz crystal is 6 MHz. Thus the maximum 8044 serial data rate is 3 MHz. This data rate drops down to 2.4 MHz when time is allowed for external clock synchronization.

4.3.1 The Bit Processor:

In the asynchronous (self clocked) modes the clock is extracted from the data stream using the on-chip digital
phase-locked-loop (DPLL). The DPLL requires a clock input at 16 times the data rate. This 16 X clock may originate from SCLK, Timer 1 Overflow, or PH2 (one half the oscillator frequency). The extra divide-by-two described above allows these sources to be treated alternatively as 32 X clocks.

The DPLL is a free-running four-bit counter running off the 16 X clock. When a transition is detected in the receive data stream, a count is dropped (by suppressing the carry-in) if the current count value is greater than 8. A count is added (by injecting a carry into the second stage rather than the first) if the count is less than 8. No adjustment is made if the transition occurs at the count of 8. In this manner the counter locks in on the point at which transitions in the data stream occur at the count of 8, and a clock pulse is generated when the count over-flows to 0.

In order to perform NRZI decoding, the NRZI decoder compares each bit of input data to the previous bit. There are no clock delays in going through the NRZI decoder.

The zero insert/delete circuitry (ZID) performs zero insertion/deletion, and also detects flags.

As an example of the operation of the bit processor, the following sequence occurs in relation to the receive data:

1) RXD is sampled by SCLK, and then synchronized to the internal processor clock (IPC).
2) If the NRZI mode is selected, the incoming data is NRZI decoded.

3) When receiving other than the flag pattern, the ZID deletes the '0' after 5 consecutive '1's (during transmission this zero is inserted). The ZID locates the byte boundary for the rest of the circuitry. The ZID deletes the '0's by preventing the SR (shift register) from receiving a clocking pulse.

4) The FCS, which is a function of the data between the flags, not including the flags, is initialized and started at the detection of the byte boundary at the end of the opening flag. The FCS is computed at each bit boundary until the closing flag is detected. Note that the received FCS has gone through the ZID during transmission.

4.3.2 The Byte Processor:

Figure 4.4 is a block diagram of the byte processor (BYP). The BYP contains the registers and controllers necessary to perform the data manipulation associated with SDLC communications. The BYP registers may be read or written by the CPU over the 8044's internal bus (IB), using standard 8044 hardware register operations. The 8044 register select PLA controls these operations. Three of the BYP registers connect to the IB through the SIB, a sub-bus which also connects to the CPU interrupt control registers.

Simultaneous access of a register by both the IB and the SIB is prevented by timing. In particular, RAM access
is restricted to alternate internal processor cycles for the CPU and the SIU, in such a way that collisions do not occur.

As an example of the operations of the byte processor, the following sequence occurs in relation to the receive data:

1) Assuming that there is an address field in the frame, the BYP takes the station address from the register file into temporary storage. After the opening flag, the next field (the address field) is compared to the station address in the temporary storage. If a match occurs, the operation continues.

2) Assuming that there is a control field in the frame, the BYP takes the next byte and loads it into the RCB register. The RCB register has the logic to update the NSNR register (increment receive count, set SES and SER flags, etc.).

3) Assuming that there is an information field, the next byte is dumped into RAM at the RBS location. The DMA CNT (RBL at the opening flag) is loaded from the DMA CNT register into the RB register and decremented. The RFL is then loaded into the RB register, incremented, and stored back into the register file.

4) This process continues until the DMA CNT reaches zero, or until a closing flag is received. Upon either event, the BYP updates the status, and, if the CRC is good, the NSNR register.
4.4 SIBEC II Board:

Sibec II is a MCS-51 family single board microcontroller made by Binary Technology, Inc. This board was designed for the Intel 8031 microcontroller and corresponding peripherals such as on-board I/O, RAM, ROM and bus drivers. This board's design has been slightly altered to suit the requirements of Intel 8344 microcontroller.

The 8344's on-chip serial interface unit is used to interface with the bus interfacing circuit. An Intel 82530 serial communication controller (SCC) is placed on the board to connect with an IBM PC serial port (Figure 4.1). With the SCC interrupt request line connected to the 8344 microcontroller INT1, the 8344 external interrupt 1 is used to indicate the presence of an input from the host, the IBM PC.

When a frame from the network is received, Intel 8344's serial port interrupt occurs and the interrupt service routine for an incoming network frame is initiated. When a character from the host is received, an external interrupt 1 occurs and an interrupt service routine for incoming local host packets and outgoing network frames is initiated.

There is an on-board DIP switch which is interfaced with the CPU through INTEL's 8255 (a Programmable Peripheral Interface) (Figure 4.5). This is useful for manually setting up the station addresses of the individual nodes. Station address could be set up through software also as explained in chapter 5.
Since in the 8344 microcontroller there is no read-only memory (ROM), use of external program memory is required. The SIBEC II is designed to support a maximum of 48K of external memory, divided into five blocks. Since the read strobe for these memory blocks is obtained by ORing the program memory strobe PSEN and the data memory strobe RD, all external memory blocks can support ROM as well as random access memory (RAM). It is up to the user to decide which blocks are used for program memory and which blocks for data memory.

In the COLAN V system, an 8K ROM (2764), residing at the addresses 0000H to 1FFFH, is used for program memory and two 8K RAMs (6264) are reserved for use as an external data memory for storage of incoming and outgoing network packets at the addresses 5000H to 7FFFH. All of the microcontroller’s internal data memory which is 192 bytes is reserved for the COLAN V operating system.

Five interrupts are supported by the 8344 microcontroller. In COLAN V two interrupts have been used by the operating system. External interrupt 1 has been used for communication between the host (IBM PC) and the node. Serial interrupt generated by the serial interface unit (SIU) is used for data transfer on the RS-485 bus as explained in sections 4.4 and 4.5.

Characters transmitted from the host are sent to the node in standard ASCII format at the data rate of 9600 baud.
4.5 System Bus:

COLAN V employs an RS-485 standard bus for the network bus, carrying signals to all of the nodes on the bus. The conventional RS-232C standard is used for serial communications between a single board computer and a host because most personal computers or other small computers support this serial interface. Conversions between these two standards are performed by the bus interfacing circuit, which has an RS-485/RS-232C converter and has been installed in the prototype area of the SIBEC II board along with INTEL's 82530 a serial communication controller (SCC).

There are two basic methods for electronic communications between the various components of a communications network: single-ended transmissions and differential transmissions. Single-ended transmissions use only one signal line and a ground level reference, and may be used only for short distances at low rates of data transmission. Long distance transmissions make it difficult to distinguish valid signals from garbled signals due to induced noise and ground shifts. Differential transmission uses two signal lines and the signal level is determined by the voltage difference between the two lines. Since induced noise appear as common-mode levels, they are rejected by the differential line receiver. This method can be used over longer distances at higher rates of data transmission.

The Electronic Industries Association (EIA) has developed several standards for both transmissions. The RS-
232C and newer RS-423 standards are for single-ended transmissions. The maximum RS-232C cable length is only 50 feet and the maximum data rate is 20 Kbps. The RS-422 and the RS-485 are used for differential transmissions. The maximum RS-485 cable length is 4000 feet and the maximum data rate is 10 Mbps. For this reason, COLAN V employs the RS-485 as a network bus.

4.6 Bus Interfacing Circuit:

The bus interfacing circuit converts RS-232C standard signals to the RS-485 standard, and vice versa. This has been installed in the prototype expansion area of the SIBEC II board along with the serial communication controller (SCC) circuit. It also continually monitors the status of the network bus, reporting this status to the SIBEC II when needed. The bus interfacing circuit includes two bus drivers (an MC1488 for the RS-232C and an SN75174 for the RS-485), two receivers (an MC1489 for the RS-232C and an SN75175 for the RS-485).

A retriggerable one-shot circuit (74LS122) transmits the bus status signal to an 8344 microcontroller I/O pin on the SIBEC II board. If there is no data on the network bus, the one-shot sends a low signal to the SIBEC II board, indicating that network bus is free for use. When there is a one-to-zero signal transition coming from the network bus, the one-shot 74LS122 sends a high signal to the SIBEC II board, a frame length, indicating that the network bus is busy. This allows the network to be allotted to a particular
transmitting station till this station receives an acknowledgement of its transmission or times out if there is a collision.

4.7 Parallel Printer Port:

Provision for a Centronics parallel printer port has been provided (Figure 4.5). This allows data transmitted to the node to be redirected to the printer.

4.8 Host:

A host running a high level language program, such as a C or PASCAL program, serves as a user interface in COLAN V. A small C program has been implemented which interfaces the IBM PC/AT keyboard, screen and the serial communications port. PROCOMM version 2.3 a communications program has been extensively used in this project. Human operators can control the system through the host to perform specific tasks, such as production line monitoring, greenhouse temperature and humidity control, experimental data collection, and so on. The control and operation of the system is accomplished by sending a command from the host keyboard or host application program, which initiates and controls the running of specific tasks.
Figure 4.1 NIU/PC Serial Interface and NIU's Local Serial Port.
Figure 4.2 Carrier Sense Circuit,
RS-485 Interface Circuit,
8344 SIU Interface Circuit.
Figure 4.3 Bit Processor
Figure 4.4 Byte Processor
Figure 4.5
Centronics Printer Port

Figure 4.6 Node Address Identification Circuit
CHAPTER 5
SOFTWARE FOR COLAN V SYSTEM

In this chapter, the low-level software design for COLAN V system is discussed. The first section describes various modes of operation of the INTEL 8344's SIU. Remaining sections are devoted to the specific aspects of the software involved in this project. This software has been developed with an idea to retain the characteristics of the original TASKMASTER system.

5.1 Modes Of Operation Of INTEL 8344's SIU:

In a given Information frame the address generation, error detection, detection of starting and ending flags are all done by the hardware of the SIU in the INTEL 8344.

The Serial Interface Unit (SIU) can operate in either of two response modes:

1) AUTO mode
2) FLEXIBLE (NON-AUTO) mode

In the AUTO mode, the SIU performs in hardware a subset of the SDLC protocol called the normal response mode. The AUTO mode enables the SIU to recognize and respond to certain kinds of SDLC frames without intervention from the 8044's CPU. AUTO mode provides a faster turnaround time and a simplified software interface, whereas NON-AUTO mode provides a greater flexibility with regard to the kind of operation permitted.
In AUTO mode, the 8344 can act only as a normal response mode secondary station. That is, it can transmit only when instructed to do so by the primary station. All such AUTO mode responses adhere strictly to IBM's Synchronous Data Link Control (SDLC) definitions (Section 5.2).

In the FLEXIBLE mode, reception or transmission of each frame by the SIU is performed under the control of the CPU. In this mode the 8044 can be either a primary station or a secondary station.

In both AUTO and FLEXIBLE modes, short frames, aborted frames, or frames which have had CRC violations are ignored by the SIU. The basic format of an SDLC frame is as shown in Figure 5.1. Format variations consist of omitting one or more of the fields in the SDLC frame. For example, a supervisory frame is formed by omitting the information field. Supervisory frames are used to confirm received frames, indicate ready or busy condition, and to report errors.

5.1.1 Flexible Mode:

In the FLEXIBLE (or non-auto) mode, all reception and transmission is under the control of the CPU. The full SDLC and HDLC protocols can be implemented, as well as any bit-synchronous variants of these protocols.

FLEXIBLE mode provides more flexibility than AUTO mode, but it requires more CPU overhead, and much longer recognition and response times. This is especially true
when the CPU is servicing an interrupt that has higher priority than the interrupts from the SIU.

In FLEXIBLE mode, when the SIU receives a frame, it interrupts the CPU. The CPU then reads the control byte from the Receive Control Byte (RCB) register. If the received frame is an information frame, the CPU also reads the information from the receive buffer, according to the values in the Receive Buffer Start (RBS) address register and the Received Field Length (RFL) register.

In FLEXIBLE mode, the 8044 can initiate transmissions without being polled, and thus it can act as the primary station. To initiate transmission or to generate a response, the CPU sets up and enables the SIU. The SIU then formats and transmits the desired frame. Upon completion of the transmission, without waiting for a positive acknowledgement from the receiving station, the SIU interrupts the CPU. Flexible mode of operation was implemented in COLAN V. Auto mode is more suitable in master-slave communication, as it involves primary and secondary stations. Flexible mode allows communicating devices to be peers on the network. This feature ideally suits the CSMA/CD bus access technique.

5.2 Standard SDLC Format:

The standard Synchronous Data Link Control (SDLC) format consists of an opening flag, an 8-bit address field, and 8-bit control field, an n-byte information field, a 16-bit Frame Check Sequence (FCS), and a closing flag. The FCS
is based on the CCITT-CRC polynomial \((X^{16} + X^{12} + X^{5} + 1)\). The address and control fields may not be extended. Within the 8344, the address and control fields is held in the Station Address (STAD) register, and the control field is held in the Control Byte (TCB) register. The standard SDLC format may be used in either AUTO mode or FLEXIBLE mode. Figure 5.1 gives the standard SDLC format.

5.3 HDLC Format:

In addition to its support of SDLC communications, the 8344 also supports some of the capabilities of High Level Data Link Control (HDLC). The following remarks indicate the principal differences between SDLC and HDLC.

- HDLC permits any number of bits in the information field, whereas SDLC requires a byte structure (multiple of 8 bits). The 8344 itself operates on byte boundaries, and thus it restricts fields to multiples of 8 bits.
- HDLC provides functional extensions to SDLC: an unlimited address field is allowed, and extended frame number sequencing.
- HDLC does not support operation in loop configurations.

5.4 Bit Stuffing

Synchronous communication involves bit oriented protocols and this means that transmission data may be arbitrary in content. It also allows much higher data rates compared to asynchronous communications over large distances. COLAN V used synchronous type of communication.
Arbitrary data transmission requires that the data field cannot contain characters which are defined to assist the transmission protocol. This property is called Data transparency. This code transparency is made possible by zero bit insertion or bit stuffing. Each frame begins and ends with a special bit pattern, namely 01111110. Whenever the transmitting hardware encounters five consecutive ones in the data, it automatically stuffs a 0 bit into the outgoing bit stream.

Whenever the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.

If the user's data contained a flag pattern 01111110, it would be transmitted as 011111010 but stored in the receiver's memory as 01111110.

5.5 Main Routine:

The main routine of the operating system runs in the fashion of an infinite loop. Before entering the loop, it initializes the network and the node. Upon entering the loop, it continues looping until a reset occurs. The flowchart for the main routine is shown in Figure 5.3.

Within the infinite loop, the main routine opens a window for the serial port interrupt and the external interrupt 1.
5.6 External Interrupt 1 Service Routine:

The external interrupt 1 service routine is responsible for collecting, processing, and storing inputs from the host. It consists of two major parts: the "local host command packet service routine" and the "outgoing network packet service routine." The local host command service routine is identical to the routine used in the original TASKMASTER. In this section, outgoing network packet processing is discussed, the flowchart for which is shown in Figure 5.2.

When a input packet from the host begins with any character other than "{", it is considered as an outgoing network packet.

5.7 Serial Interrupt Routine:

Serial Interrupt routine is responsible for clearing the Serial Interface Unit as soon as a transmission or a reception takes place.

5.8 Implementation of Immediate Task:

TASKMASTER has a task library. One of these tasks has been implemented in COLAN V. This task is an immediate task. When a command is issued from a local node in a command format an immediate task in the remote node is executed. The task chosen in the remote node is to blink an LED. Remote node's address is generated in the main loop of COLAN V as specified in station address register(STAD). This command is differentiated from data format by the local node when it receives "{" from the host. Once "{" is
received the node waits for the task number. In this case task 15 has been interpreted as a task for blinking an LED in the remote node. This command would be executed only if the node sees a "15" and a closing ")" from the host indicating completion of command. The structure of this command is shown below:

(15)

5.9 Implementation of Mail Transfer:

A simple mail transfer across the hosts has been implemented in COLAN V. The data frame length for transmission has been fixed at 50 bytes. Thus whenever the CPU receives 50 bytes of data from the host it immediately transfers these 50 bytes across the network to the destination node as specified in the station address (STAD) register. If a file of more than 50 bytes in length is to be transferred an end of file (EOF) mark has to be inserted which when CPU comes across would transfer the file upto the mark. The EOF mark is ".". Thereby files of various lengths could be perfectly transferred across the hosts.
<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opening Flag</td>
<td>8 bits</td>
</tr>
<tr>
<td>Address Field</td>
<td>8 bits</td>
</tr>
<tr>
<td>Control Field</td>
<td>8 bits</td>
</tr>
<tr>
<td>Inform Field</td>
<td>&gt;0 bits</td>
</tr>
<tr>
<td>Frame Check Sequence</td>
<td>16 bits</td>
</tr>
<tr>
<td>Closing Flag</td>
<td>8 bits</td>
</tr>
</tbody>
</table>

Figure 5.1

Standard SDLC Frame Format
Figure 5.2

External Interrupt Routine
Figure 5.2.1

External Interrupt Routine Continued
Move data into XMIT buffer

Is it end of file mark?

Increment data buffer pointer

Fill blanks in leftout mem bytes of buffer

Set TBF
set RTS
XMIT to Bus

Clear NSWD .1 CMD flag

Figure 5.2.2

External Interrupt Routine Continued
RESET

Clear data and screen RCV flags

Blink LED to check if system is working

Set serial Int higher priority declare stack

Set transmit buff start Addr register r

Figure 5.3

Main Loop
Figure 5.3.1

Main Loop Continued
Figure 5.3.2
Main Loop Continued
Figure 5.3.3
Main Loop Continued

Move data to host screen

Check control byte of 82530 (SCC)

Is RCV flag of 82530 cleared?

Has the data buffer of 8344 fully been read?
Figure 5.4
Serial Interrupt Routine
CHAPTER 6

CONCLUSIONS AND SUGGESTIONS

The previous chapters of this study have considered various features of the system design and implementation of COLAN V, a control-oriented local area network. The design goal of COLAN V was to implement a relatively simple and reasonably performing LAN based upon the CSMA/CD bus access method, at a low cost.

6.1 Features of COLAN V:

COLAN V retains some characteristics of TASKMASTER SYSTEM. It is an extremely simple system. Software is very simple. This is because, unlike previous COLANs, communications overhead is dealt with in hardware by serial interface unit (SIU) of an INTEL 8344 on a concurrent basis. On-chip error detection techniques (CRC) and synchronous mode of communications have greatly improved the speed and performance of the system. It can implement simple tasks. COLAN V accommodates multiple users, using CSMA/CD method to share the bus. By adding mailing features to the TASKMASTER SYSTEM, network users are able to exchange information sources. Moreover, COLAN V makes it possible for users to share expensive application resources.
6.2 Recommendations for Future Development:

The first version of COLAN V operating system, using the CSMA/CD protocol, has been successfully implemented. COLAN V can presently execute a simple mail transfer across the nodes using packet switching and can execute a remote task. Following suggestions are provided as a guide to future improvements.

6.2.1 Increasing Data Rates:

COLAN V currently operates up to 300 Kbps. By utilizing an external clock, data rates up to 1 Mbps or more can be achieved. RS-485 will support a data rates of up to 10 Mbps.

6.2.2 Long Frame Formats:

COLAN V has been implemented utilizing a information frame length of 50 bytes for data transmission. This frame length could be increased up to theoretical limits of 64 K [INTE 86, APP NOTE 283].

6.2.3 Completion of Task Library:

Since COLAN V is a control-oriented network, it will be important to complete the task library by adding more complex control tasks. More advanced mailing and queue management tasks, such as a mail directory list and asking for permission prior to transfers, are necessary features of a mature system and should be added.
APP283 Parviz Khodadadi, "Flexibility in Frame Size with the 8044", Intel Application note, August 1986.

DIGI 84 Young SOHN & Charles Gopen, "Networking with 8044", Digital design, May 1984, pp. 136-137.


THYE 88 Yong Thye. "COLAN IV, A Local Area Network for Communications and Control". Unpublished master’s

APPENDIX A

Expansion of Acronyms used in this thesis:

ALU  Arithmetic Logic Unit
ASCII American Standard Code for Information Interchange
BIP  Bit Processor
BYP  Byte Processor
COLAN Control Oriented Local Area Network
CSMA Carrier Sense Multiple Access
CTS  Clear to Send
CD   Collision Detection
CPU  Central Processing Unit
CCITT Consultative Committee for International Telephony and Telegraphy.
CRC  Cyclic Redundancy Code
DN   Device Number
DD   Data Field
DMA  Direct Memory Access
DPLL Digital Phase Locked Loop
EIA  Electronics Industries Association
FCS  Frame Check Sequence
HDLC High Level Data Link Control
IB   Internal Bus
ISO  International Standards Organization
Kbits/s Kilo bits per second
LAN  Local Area Network
Mbps Mega bits per second
NSNR Send/Receive count Register
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRZI</td>
<td>Non-Return to Zero Inverted</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnect</td>
</tr>
<tr>
<td>PLA</td>
<td>Programmable Logic Array</td>
</tr>
<tr>
<td>PPI</td>
<td>Programmable Peripheral Interface</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>ROM</td>
<td>Read Only Memory</td>
</tr>
<tr>
<td>RXD</td>
<td>Receive Data</td>
</tr>
<tr>
<td>RFL</td>
<td>Receive Field Length</td>
</tr>
<tr>
<td>RBS</td>
<td>Receive Buffer Start</td>
</tr>
<tr>
<td>RBE</td>
<td>Receive Buffer Empty</td>
</tr>
<tr>
<td>RCB</td>
<td>Receive Control Byte</td>
</tr>
<tr>
<td>RTS</td>
<td>Request to Send</td>
</tr>
<tr>
<td>SIB</td>
<td>Serial Interface Bus</td>
</tr>
<tr>
<td>SCLK</td>
<td>Serial Clock</td>
</tr>
<tr>
<td>SDLC</td>
<td>Synchronous Data Link Control</td>
</tr>
<tr>
<td>SIBEC II</td>
<td>Single Board Microcontroller</td>
</tr>
<tr>
<td>SCC</td>
<td>Serial Communication Controller</td>
</tr>
<tr>
<td>SIU</td>
<td>Serial Interface Unit</td>
</tr>
<tr>
<td>TXD</td>
<td>Transmit Data</td>
</tr>
<tr>
<td>ZID</td>
<td>Zero Insertion and Deletion</td>
</tr>
</tbody>
</table>