AN ABSTRACT OF THE THESIS OF

Doo-Hun Eum for the degree of Master of Science in Electrical and Computer Engineering presented on September 17, 1987.

Title: COLAN III, A Control-Oriented LAN Using CSMA/CD Protocol

Redacted for privacy

Abstract approved: James H. Herzog

COLAN III is a control oriented LAN designed to achieve reasonable network performance and flexibility at low cost. These objectives are achieved by utilizing inexpensive microcontrollers (8031s) and hardware support. The CSMA/CD network protocol with single bus structure has been employed in COLAN III as an arbitration method to control the access to the bus.

This thesis discusses the design and implementation of COLAN III in two detailed parts: the system structure and the operating system. The COLAN III bus uses the EIA RS-485 standard, which has a higher data rate and a lower error rate than the conventional RS-232C standard. The COLAN III operating system was developed using two implemented nodes and later tested with three nodes. Every control-oriented task and electronic mailing activity worked with three nodes, meaning that the system should logically work with any number of nodes.

COLAN III, A Control-Oriented LAN Using CSMA/CD Protocol

by

Doo-Hun Eum

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed September 17, 1987

Commencement June 1988

APPROVED:

Redacted for privacy

Associate Professor of Computer Engineering in charge of major

Redacted for privacy

Head of Department of Electrical and Computer Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented September 17, 1987

Typed by B. McMechan for Doo-Hun Eum

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

COLAN III, A CONTROL-ORIENTED LAN USING CSMA/CD PROTOCOL

CHAPTER 1

INTRODUCTION

1.1 Background

A Local Area Network (LAN) is defined generally as a communication network interconnecting a variety of data communicating devices within a small area. This definition implies that a LAN has the following basic properties [STAL 84]:

1) A LAN is a communication network containing its own communication facilities;
2) The data communication devices which communicate over a transmission medium include computers, terminals, peripheral devices, sensors, and telephones; and
3) A LAN is geographically local.

Therefore, the opportunity to optimize a network for a particular user's application is a key feature of the LAN.

Analysis of LANs reveals that they provide greater reliability of data processing than do large remote networks. Because of the small distances between user systems in local area networks, the transmission rate of data units can be increased in a reasonably simple manner. Since transmission errors are reduced in the process, the algorithms to detect and correct these errors are of relatively simple formulation.

The main attributes of local networks may be characterized as follows:

1) They include inexpensive devices (modems, repeaters and transceivers);
2) They provide ease of physical connection of devices to the media;
3) They provide for high data transmission rates; and
4) There may be a high degree of interconnectivity between devices.

In terms of their functional purpose, LANs may be divided into two groups: (1) general-purpose networks providing for all types of data processing and (2) local specialized networks.

Of late, a new generation of microcontrollers has provided design engineers with high performance and cost effective methods of implementing control strategies for a wide variety of real-time systems [HERZ 87]. Complex control and monitoring systems often require multiple microprocessors, each optimized and located for best performance. Thus, the natural evolution of microcontrollers has been toward systems with multiple controller units which may be spatially distributed. A case in point is TASKMASTER, an experimental microcontroller node, designed and constructed at Oregon State University to demonstrate the feasibility of performing real-time, distributed control actions. In COLAN III, the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) LAN protocol has been adopted to allow for the integration of TASKMASTER nodes into high performance control and mailing systems, which may then be managed by conventional personal computers using conventional communications networks. This study, therefore, is a discussion of the modification of a simple daisy-chain structured control system (TASKMASTER SYSTEM) into a simple LAN system (COLAN III), using the CSMA/CD protocol.

In any network there exists a collection of machines intended for running user programs. These machines, the "hosts," are connected by a communication subnet carrying messages from host-to-host. The subnet consists of two basic components: (1) switching elements and (2) transmission lines. The switching elements are specialized computers used for network management, also known as Interface Message Processors (IMP) [TANE 81]. In COLAN III, microcontrollers serve as the switching elements. The transmission lines are often called communication channels or media. They may be coaxial, twisted wire, or fiber-optic cables. In COLAN III, twisted wire pairs are used.

In general terms, two types of subnets are commonly used, the first composed of point-to-point channels, such as a telephone network, and the second composed of broadcast channels. In the latter subnet, a single communication channel is shared by all IMPs. As a result, messages sent by an IMP are received by all of the other IMPs on the network. The message must contain an internal designation, specifying the IMP to which it is directed; all the non-designated IMPs merely ignore the message.

There are two basic types of LAN broadcast systems. The first is a bus network in which at any given instant only one IMP, designated as the bus master, is allowed to transmit; all of the other IMPs must refrain from transmission at that time. A bus network contains an arbitration mechanism, either centralized or distributed, which resolves conflicts when two or more IMPs wish to transmit simultaneously. In COLAN III, a bus structure with a distributed control scheme was adopted.

The second broadcast system is the ring network. In a ring, each bit propagates around the ring on its own and does not wait for the remaining bits of the message of which it is a part. Typically, each bit

circumnavigates the entire ring within a few bit times, often prior to completion of the entire message transmission.

1.2 COLAN III

Recently, a great deal of effort has been devoted to the study, design, standardization, and implementation of LANs. The organization of most LAN implementations have required specialized and somewhat expensive hardware support. Consequently, the design of a low-cost LAN for a control-oriented and electronic mailing system capable of reasonable levels of performance purposes is the main interest of this study. Moreover, this project is one of the first trials implementing a relatively simple LAN while using the new Electronic Industries Association (EIA) RS-485 bus with a baseband technique.

The concept of modifying the existing daisy-chain structured TASK-MASTER system into a local area network was initiated by Y. P. Zheng [ZHEN 86], who in 1986 developed COLAN, a control-oriented local area network. COLAN used a hybrid medium access method: token passing and CSMA/CD. Since COLAN is very complex, it was not fully implemented. COLAN III uses CSMA/CD only to control the bus access and its design and implementation are less complex than that of COLAN.

The COLAN III system consists of several nodes, connected by a single bus using the RS-485 standard. Each node consists of one bus interfacing circuit board, with a RS-485/RS-232C converter changing the RS-232C standard into the RS-485 standard, or vice versa; one microcontroller (an Intel 8031) based board; and one host, most often a personal computer.

COLAN III, intended to operate under lighter loads, employs a CSMA/CD media access protocol to control the access to the bus. The

CSMA/CD protocol presents advantages under lighter loads because there is no token passing overhead and no significant collisions are present. In COLAN III, a node wishing to transmit monitors the medium and obeys the following rules:

1) If the medium is idle, transmit; otherwise, continue to monitor the channel until it is clear that it is idle.
2) If collision is detected during transmission, immediately cease transmitting the packet while transmitting a few more jamming characters to assure that all nodes involved in the collision are aware that there has been a collision.
3) After transmitting the jamming characters, wait a random amount of time (dependent upon its own node address) to avoid further collisions, then again attempt to transmit, observing the above rules. If there has been no collision, wait for the acknowledgment from the destination node. If there is no acknowledgment, begin again, observing the above rules.

This chapter has served to introduce COLAN III, establishing its background as a control-oriented LAN and summarizing its objectives and most important features. Chapter 2 is a review of related LAN materials and systems and in Chapter 3 improvements to the original TASKMASTER SYSTEM and the design approach of COLAN III are introduced. Details of the hardware structure for COLAN III are discussed in chapter 4 and its software algorithms are explained in chapter 5. Chapter 6 is a brief description and discussion of high level software support required for user friendliness.

The final chapter of this study begins with a summary of the COLAN III system. The unsolved problems which remain are identified and suggestions will be offered for future research.

CHAPTER 2

REVIEW OF RELATED MATERIALS AND SYSTEMS

2.1 Introduction

A LAN system includes definitions of the network topology, the physical transmission medium, the switching technology, the network control mechanism, and access methods or communication protocols for transmission between users. These related activities are briefly discussed below.

2.2 Topology

The term "topology," in the context of a communications network, refers to the manner in which nodes, user devices, and transmission links are interconnected. The following four topologies, Bus, Star, Loop, and Mesh, have been used widely for LANs [ANDE 75, SOI 81, MALH 86]. For reason of its simplicity and low cost, COLAN III encompasses use of a bus topology.

2.2.1 Bus Topology

A bus network consists of a highway for transporting information between nodes which are connected to it (see Figure 2-1a). Each node is given a unique address, which other nodes append to messages dispatched to the target node. Since only one transmission path is available for the entire network, there must be a method in which its use may be shared. The most common techniques used in this topology are the Carrier Sense Multiple Access (CSMA) and the token bus.

2.2.2 Star Topology

The star network, shown in Figure 2-1b, consists of a central switching resource to which a number of nodes are connected, each by a functionally single, but bidirectional path. Messages are exchanged between network nodes using the central switching resource as an intermediary. Therefore, this topology eliminates the need for each network node to determine routing decisions, allowing a particularly simple structure for each of the nodes composing the network.

2.2.3 Loop Topology

The basic organization of this specialized network is simple. Several processors are linked together, as shown in Figure 2-1c, to form the equivalent of a ring. The appeal of the ring structure lies in the simplicity of its high channel utilization and an absence of message routing problems. Since all information passes through each processor, the only decision faced by each is whether or not to extract the information transmitted on the network.

2.2.4 Mesh Topology

This form, shown in Figure 2-1d, is most often found in public data communication networks and also in some modern private networks. Mesh topology is required when it is important to have redundant connections, possibly to handle exceptional traffic loads or to provide a high degree of security from line failures by providing alternative paths between nodes.

2.3 The Transmission Medium

The transmission medium is the physical path between the transmitter and the receiver in a communication network. The most common media are twisted pair copper cables, coaxial cables, and fiber-optic cables. Among the three, COLAN III adopts the twisted pair cable (EIA RS-485) as its

transmission medium because the principal motive underlying this project is to implement a low cost LAN with a reasonable performance.

2.3.1 Twisted Pair Cables

Twisted pair cables are suited to both analog and digital transmissions. Bandwidth and distance characteristics which can be supported by twisted pair cable, however, are much less than those provided by coaxial cable. Using a typical twisted pair cable, 1 to 2 Mbps can be achieved. It is best suited for transmission over relatively short distances between devices or repeaters on the network.

2.3.2 Coaxial Cables

Coaxial cables have existed since the early 1940s. They are used extensively in long distance telephone toll trunks and local area networks. The popularity of coaxial cable is due to its moderate cost, high bandwidth, and low bit-error rate. Coaxial cables are designed to provide for greater bandwidth and faster bit rates than twisted pair cables. Two modes of data transmission are possible using coaxial cable: baseband and broadband. In baseband signaling, the information is sent through the cable in essentially unmodulated form. Each data bit is represented by a discrete signal level on the cable. Broadband, on the other hand, uses a modulation technique to transmit multiplexed analog signals.

2.3.3 Fiber-Optic Cables

The most exciting recent development in the area of transmission media is fiber-optic cable. Optical fiber transmits a signal encoded beam of light by means of total internal reflection. This fiber-optic channel possesses several properties which make it the preferred transmission medium for some applications, including those requiring very high rates of transmission (up to 1 gigabits/s is possible) and a low error rate.

2.4 Digital Switching

Switching is a means of allocating resources (space, bandwidth, or time) to users or machines that require resources to communicate. In point-to-point operations the alternatives are multiplexing and switching, while the multipoint (e.g., broadcast, multidrop, or bus) operational alternatives are polling and contention. All of these alternatives involve the basic switching disciplines available for line switching: circuit, message, and packet.

COLAN III employs a type of message switching because the message itself contains the addressing information and the entire message is passed through the network without segmentation of the message into a number of packets.

2.4.1 Circuit Switching

Circuit switching is the oldest and the most widely used method. In this system a dedicated communication path is established between two stations through the nodes of the network.

2.4.2 Message Switching

In message switching a block of data, the message, finds its way through network resources that are not assigned exclusively for the message. Hence, there is no need to set up a physical path before transmitting a message. This switching method requires each switching node along the path to the destination to store a message coming through one of its input lines, and then forward this message to one (or all) of its outgoing lines.

2.4.3 Packet Switching

Packet switching is similar to message switching. The principal external difference is that the length of the units of data is limited in a packet-switched network. Namely, in packet switching messages are

grouped in packets. Long messages may be segmented into a number of packets that can be passed individually through the network. The packets contain addressing information that enables the switching nodes to send each packet to its proper destination.

2.5 Network Control and Routing Mechanism

One of the principal requirements of a distributed computer network is network control and routing. The control mechanism is responsible for all activities associated with network communications. When a network resource is shared by multiple nodes, there must be some method whereby a particular unit can request and obtain the resource at a given time. The control schemes are classified as "centralized" or "decentralized."

COLAN III adopts decentralized network control because control operations are more flexible with respect to the application requirements of the network.

2.5.1 Centralized Control

In a centralized control system all network activities are governed, and the network states are traced, by a central controller.

2.5.2 Decentralized Control

In a decentralized control structure the distributed components of the decentralized algorithm interact and cooperate to control the entire network via communications among themselves.

2.6 Communication Protocols

Normally, communication functions are partitioned into a hierarchical set of layers. Communication is achieved by having the corresponding (or "peer") layers in two systems communicate by means of a set of rules or conventions known as a protocol. Each layer performs a related subset of the functions required to communicate with the other system. Each layer

relies on the next lower layer to perform the more primitive functions and provide services to the next layer. Each layer is defined so that changes in one layer do not require changes in other layers [STAL 84]. In May 1983, the International Standards Organization (ISO) approved specifications for the Open Systems Interconnection (OSI) shown in Figure 2-2. Table 2-1 summarizes the functions that must be performed within a system in order to establish communications [IEEE 83].

Due to such characteristics as simplicity, high speed, low transmission delay, the wide range of application areas, and the use of real time control operations, the LAN protocol design strategy is quite different from that of conventional long-haul networks. An important and different characteristic of LANs is that the data are transmitted in address frames. Most local area networks include not only the machine address, but also the process address in both the source and the destination address fields of all packets or messages.

The service layer for a typical LAN might include the following [FRAN 81]:

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 acknowledgment 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 protocols 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.

2.7 Local Area Network with Bus Topology

Most of the studies devoted to the design and implementation of LANs have been based on simple linear bus or ring topology [KANG 86]. The use of a bus topology as the communications subnetwork is a popular approach to construction of LANs. Ethernet and token bus are two typical examples of systems which utilize distributed bus control techniques. The principal motivations underlying COLAN III are relatively simple network control and the reduction of hardware costs by the use of a single transmission medium.

COLAN III adopts the CSMA/CD protocol with acknowledgment processing, because the CSMA/CD protocol is more efficient and simpler than the token bus protocol in systems under lighter loads, such as COLAN III.

2.7.1 CSMA/CD

The original Ethernet system was installed on an experimental basis in the Palo Alto Research Center of the Xerox Corporation in the early 1970s. Its purpose was to connect office workstations to expensive computing resources or other office machinery in order that they could be shared by every workstation. Ethernet uses a Carrier Sense Multiple Access (CSMA) technique, in which a single channel is shared by all users. Each user may transmit a packet at any time and the system is reliant upon the intended recipient hearing the transmission and reading it into its own store for subsequent processing. Other users of the network are instructed to ignore transmissions which are not addressed to them and to avoid attempts to transmit when the channel is in use. An automatic process acknowledging successful receipt of the packet was not built into this technique.

With a 1-persistent CSMA protocol, a station wishing to transmit monitors the medium and obeys the following rules [STAL 84]:

1) If the medium is idle, transmit;
2) If the medium is busy, continue to listen until the channel is sensed idle, then transmit immediately; and
3) If there is a collision (determined by a lack of acknowledgment), wait a random amount of time and repeat step 1.

This protocol is called "1-persistent" because the station transmits with a probability of 1 whenever it finds the channel idle.

When two frames collide, the medium remains unusable for the duration of transmission of both damaged frames. The amount of wasted

bandwidth can be considerable for long frames. If the station transmitting a frame can monitor the medium and stop the transmission as soon as it detects a collision, performance will be improved. This is the concept of CSMA/CD. With a CSMA/CD protocol, two rules can be added to the CSMA rules:

1) If a collision is detected during transmission, immediately cease transmitting the frame and transmit a brief jamming signal to assure that all stations are aware that there has been a collision; and
2) After transmitting the jamming signal, wait a random amount of time, then attempt to transmit again using CSMA.

Besides 1-persistence, there are also nonpersistence and p-persistence CSMA/CD protocols. But the most common choice is 1-persistence. It is used by both Ethernet and MITREnet and in the IEEE 802 standard.

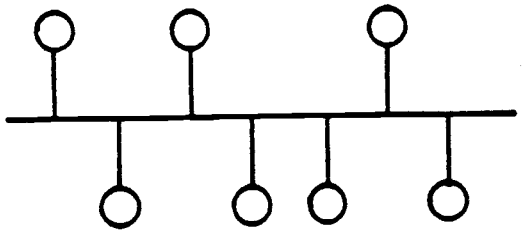
A higher level of protocol is required to return explicit acknowledgment packets, or requests for retransmission when an error in the packet is detected by the recipient.

2.7.2 Token Bus

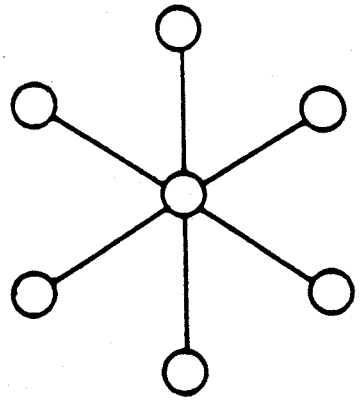
The token bus is another popular LAN using bus topology. In this system, although messages are simultaneously broadcast to all nodes in the network, as is the case with Ethernet, the stations are formed into a logical ring in which control is passed in sequence from station to station.

Each station is assigned a logical position in an ordered sequence, with the last member of the sequence followed by the first. The physical ordering of the stations on the bus is not important. Each station knows the identity of the stations preceding and following it. All stations on the bus need not necessarily be on the logical ring. Stations may receive a frame, but they may not transmit a frame.

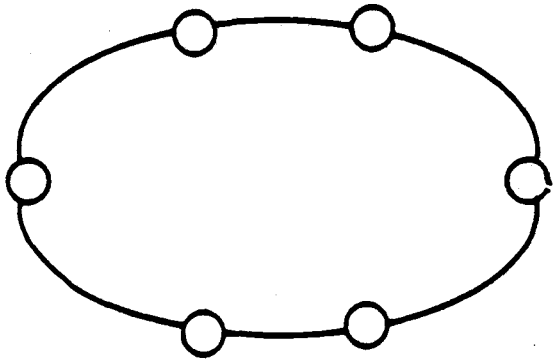
A control frame, known as the token, regulates the right of access. This token includes a destination address and stations receiving the token control the bus for a specified time. The control station may transmit frames to other stations, poll other stations, and get responses. When these actions have 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.



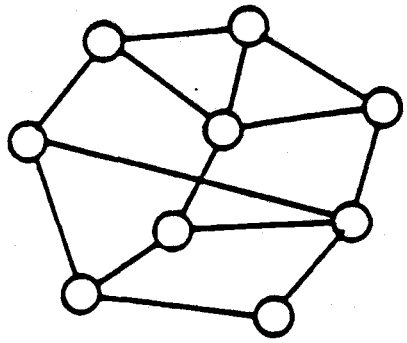
2-1a. Bus Topology



2-1b. Star Topology



2-1c. Loop Topology



2-1d. Mesh Topology

Figure 2-1. Local Area Network Topologies.

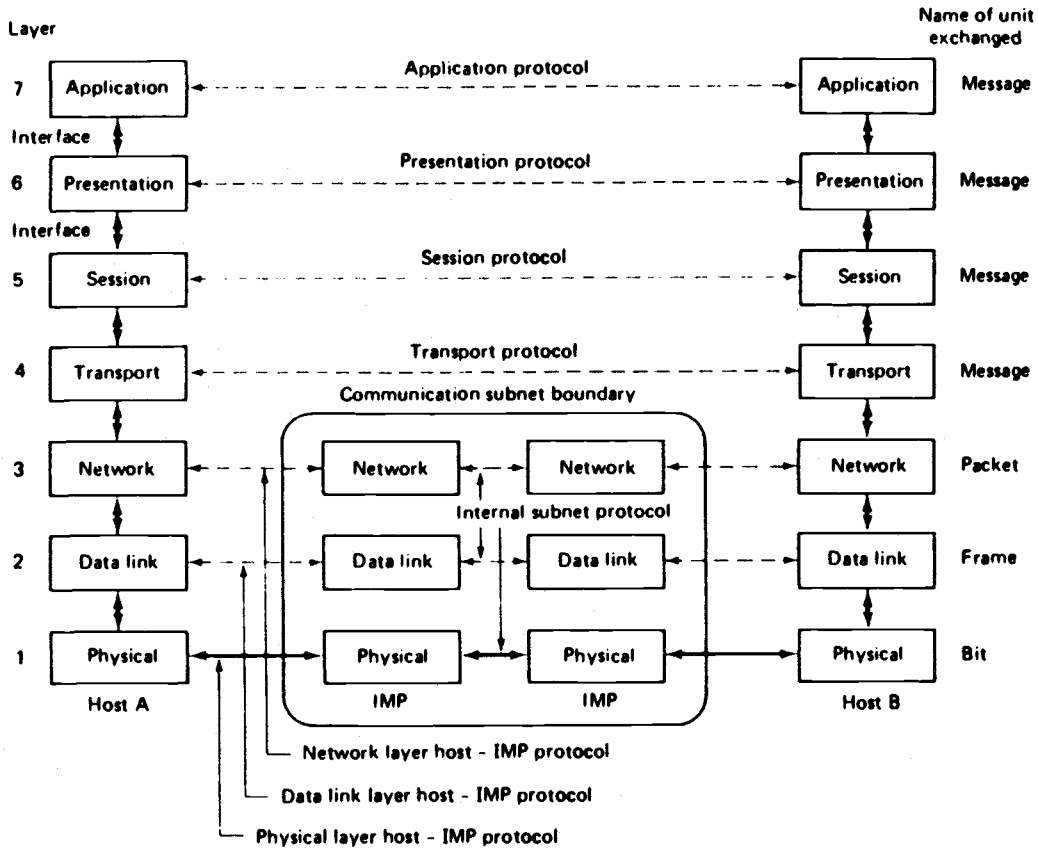


Figure 2-2. ISO Open System Interconnection Reference Model [TANE 81].

Table 2-1. Definitions of OSI Layers [STAL 84, IEEE 83].

Layer	Definition
1. Physical	Concerned with transmission of unstructured bit stream over physical link; involves such parameters as signal voltage swing and bit duration; deals with the mechanical, electrical, and procedural characteristics to establish, maintain, and deactivate the physical link (RS-232-C, RS-449, X.21).
2. Data link	Converts an unreliable transmission channel into a reliable one; sends blocks of data (frames) with checksum; uses error detection and frame acknowledgement (HDLC, SDLC, BiSync).
3. Network	Transmits packets of data through a network; packets may be independent (datagram) or traverse a preestablished network connection (virtual circuit); responsible for routing and congestion control (X.25, layer 3).
4. Transport	Provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control.
5. Session	Provides means of establishing, managing, and terminating connection (session) between two processes; may provide checkpoint and restart service, quarantine service.
6. Presentation	Performs generally useful transformations on data to provide a standardized application interface and to provide common communications services; examples: encryption, text compression, reformatting.
7. Application	Provides services to users of the OSI environment; examples: transaction server, file transfer protocol, network management.

CHAPTER 3

COLAN III SYSTEM

3.1 Overview of TASKMASTER SYSTEM

COLAN III is based on the TASKMASTER SYSTEM, a daisy-chain structured system. The TASKMASTER SYSTEM consists of one host computer and one or more microcontroller based boards, called the TASKMASTER(s). The host is a master and the TASKMASTER boards are intelligent slaves capable of performing specified control tasks. The configuration of TASKMASTER SYSTEM is shown in Figure 3-1.

The TASKMASTER SYSTEM is a control-oriented distributed system designed for a real-time control purpose. Each control action is accomplished through the use of system tasks, each of which has a unique task number which can be issued by the host computer. Tasks, in the form of a task library, reside in program memory on the TASKMASTER board. The library can be modified by users to perform specified tasks.

The host communicates with a TASKMASTER by issuing a command packet via its serial port. Every command issued by the host goes to all TASKMASTERs and, in turn, the host receives all communications originating at any TASKMASTER [HERZ 87]. A command from the host must specify the unique device number of each TASKMASTER board, the prefix control of the task, task number, the postfix control of the task, and space for up to 10 characters to specify task parameters, data, or options.

An interrupt is generated when a TASKMASTER unit receives a character on its communication port. The operating system of TASK-MASTER temporarily suspends any task which is currently running and serves the received character. This brief interruption of a task is a transparent activity. When a complete task specification packet has been received, the operating system examines the address field. If the address is correct, the task specification is placed on the task queue.

The format of a command packet is:

{AA P NN S DD DD DD DD DD /}

Individual elements of the command packet are interpreted as follows:

"{" and "}" delineate the start and end of a command packet.

"AA" is a sequence of two ASCII character indicating the device address.

"P" is the prefix control character informing an addressed TASKMASTER of the method to be used in responding to the directive specified by the task number to follow. Three types of pre-execution controls are permissible:

: the task is placed on the task queue and runs at completion of all other tasks on the queue;

? the task is placed on the task queue and will not start until the host issues a special "synchronizing" packet allowing the host to have complete control over the starting time of these tasks; or

! the task runs immediately and temporarily suspends any existing task which is currently running.

"NN" is a sequence of two numeric ASCII characters which specifies the task number to be run.

"S" is a postfix character indicating the action associated with task termination. There are two possibilities:

. the task is discarded after termination; or

+ the task is requeued after termination, allowing some or all tasks to be continuously looped for repetitive action.

"DD" is an optional data field. This data field may contain up to five pairs of hexadecimal characters, which allow the host to pass data, arguments, or encoded instructions to the TASKMASTER.

"/" is an optional echo request field used for error control purposes. The prefix and postfix characters allow a variety of special task sequencing to be accomplished. Adress "00" is treated as a universal address to allow commands to be simultaneously addressed to all nodes. All blanks within packets are ignored.

The operating system of the TASKMASTER consists of two major parts, the main routine and interrupt service routine.

3.1.1 Main Routine

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

3.1.2 Interrupt Service Routine

This part of the operating system is for interpretation of the command packet: instructions to receive a character from the host or process the command packet upon receiving a complete packet and, if a queued or synchronized task, put the modified packet on the task queue or execute it if it is an immediate task.

The flowcharts for the algorithms of both routines are shown in Figure 3-2.

3.2 Overview of COLAN III

COLAN III adopts a bus structure, which is shown in Figure 3-3. The system consists of several nodes, which are connected by a single bus using the RS-485. Each node consists of one bus interfacing circuit board, one microcontroller (8031) based board, and one host, typically a personal computer. The CSMA/CD arbitration method has been adopted in COLAN III to control access to the bus. While the TASKMASTER SYSTEM supports only real-time control tasks, there are two possible views of COLAN III: the control-oriented view and the mail oriented view. Each is dependent on the system application: real-time control or electronic mailing. Basically, the two views of the system are the same, except for the format of transmitted information. Transmitted information in the control orientation are remote commands, while those in the mail orientation are either messages entered on the screen console of a host computer or the contents of disk files.

3.2.1 Control-Oriented View

The operating system of COLAN III incorporates the use of multiple TASKMASTERs (microcontroller based boards) in a single unified real-time system utilizing multiple hosts, rather than a single host as in the TASKMASTER SYSTEM. The complete COLAN III system in the control-oriented view requires three components:

1) The system scheduler (one host computer and its TASKMASTER) is the source of all scheduling and coordination activities;

2) TASKMASTERs (with the exception of the one associated with the system scheduler) are the "action elements" which carries out the operation, using its specialized application interface; and

3) The communication system (RS-485 bus with CSMA/CD protocol) provides the transmission media for exchange of information between the scheduler and the TASKMASTERs.

Complex control and monitoring systems often require multiple microprocessors, each optimized and located for maximum performance [HERZ 87]. In order to achieve overall control and synchronization, a system of multiple TASKMASTERs and multiple schedulers are used in COLAN III. The main design issue of COLAN III was directed at the use of multiple TASKMASTER units, which may be spatially distributed and may be scheduled by any node on the bus. COLAN III utilizes a TASKMASTER at each of several distributed points of control. Each TASKMASTER will have specialized interfaces and support hardware to perform a highly individualized activity. COLAN III allows these TASKMASTERs to be easily integrated into a high performance system which may be managed by any scheduler on the network, using the CSMA/CD communications protocol.

The greatest improvement from the 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.

As with any system using a shared communication link, there must be a method for assuring that only one device at a time attempts to send information. In the daisy-chained TASKMASTER SYSTEM, a conventional

computer acts as a central authority to regulate the use of the communication channel and there is no arbitration method to access the shared channel. COLAN III improves this system by the adoption of the CSMA/CD communications protocol as a arbitration method to access the shared channel. This method allows for fair access to the bus, higher performance, greater simplicity, and ease of application programming. In COLAN III, control of access to the bus is distributed fairly: all nodes have an equal chance to use the network to communicate.

3.2.2 Mail-Oriented View

In addition to control-oriented activities, COLAN III also supports electronic mailing activities, including file transfers and direct message transfers. In this orientation, each node may be conceived as a person standing in a room at a party having a polite conversation. If one person wishes to speak, and if no one else is speaking, the person simply begins to speak. If someone else is talking, the person wishing to speak waits for them to finish. However, it is possible that two people, noting that no one else is talking, both begin to speak at the same time. They would then realize that they are interfering with one another and both would stop speaking and wait a random period of time before starting again. Eventually, one person would again begin to speak before the other could take the opportunity, and thus gain the floor.

Further details of mailing activities will be discussed in subsequent chapters.

3.2.3 COLAN III CSMA/CD Protocol

COLAN III uses a CSMA/CD media access protocol to control access to the shared bus. In COLAN III, a node wishing to transmit control-oriented commands or electronic mail monitors the medium and obeys the following rules:

1) If the medium is idle, transmit;
2) If the medium is busy, continue to listen until the channel is sensed idle;
3) If there no collisions occur during transmission, go to step 5; if a collision is detected, immediately cease transmitting the packet and transmit three more jamming characters to assure that all nodes involved in the collision are aware that there has been a collision;
4) Wait a random amount of time, dependent upon its own node address, then repeat step 1;
5) Wait for the acknowledgment from the destination node, receipt of which indicates the completion of transmission; or
6) If there is no acknowledgment, repeat step 1.

3.2.4 COLAN III and IEEE 802 CSMA/CD

COLAN III was designed and implemented based on Ethernet. Since the IEEE 802 CSMA/CD standard is very close to that of Ethernet, COLAN III is closely related to the IEEE 802 CSMA/CD. Indeed, the differences between the two are in the frame structure. The principle of the CSMA/CD method used in COLAN III is the same as described in the IEEE 802.3 standard [IEEE 82a].

COLAN III implements the physical layer and medium access control layer as described in the IEEE 802.3 standard. The logical link layer, which is above the medium access control layer, was partially implemented and will be completed in future development of COLAN III.

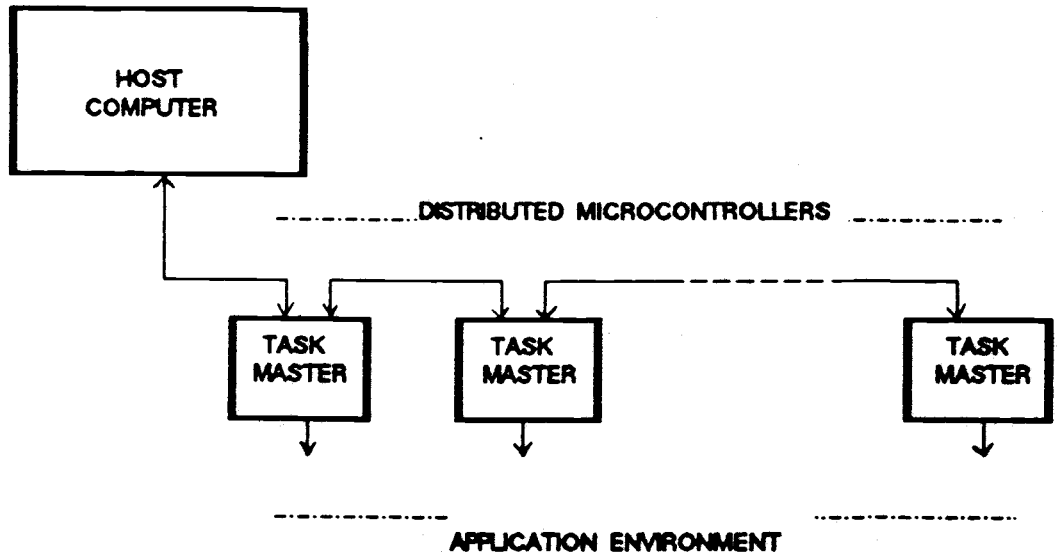


Figure 3-1. TASKMASTER SYSTEM Configuration [HERZ 87].

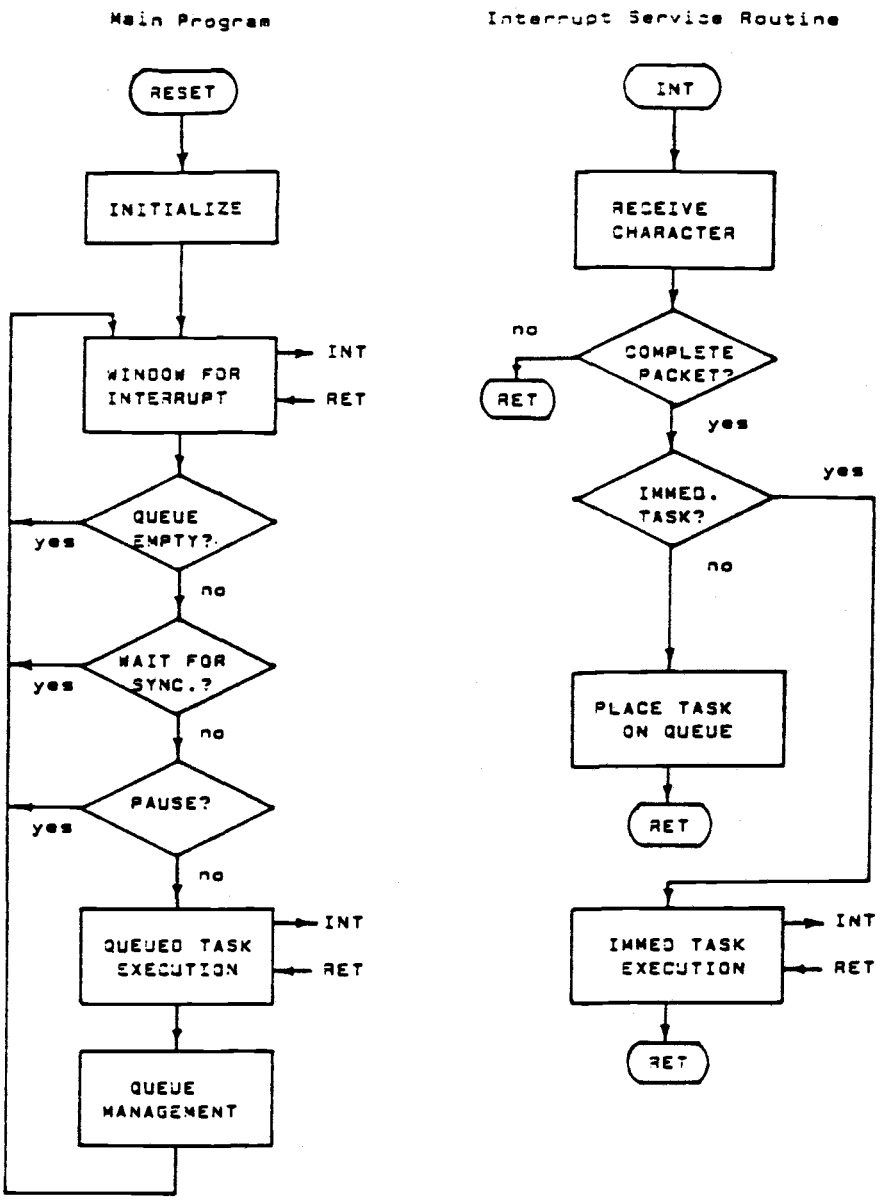


Figure 3-2. TASKMASTER Operating System Flowchart [HERZ 87].

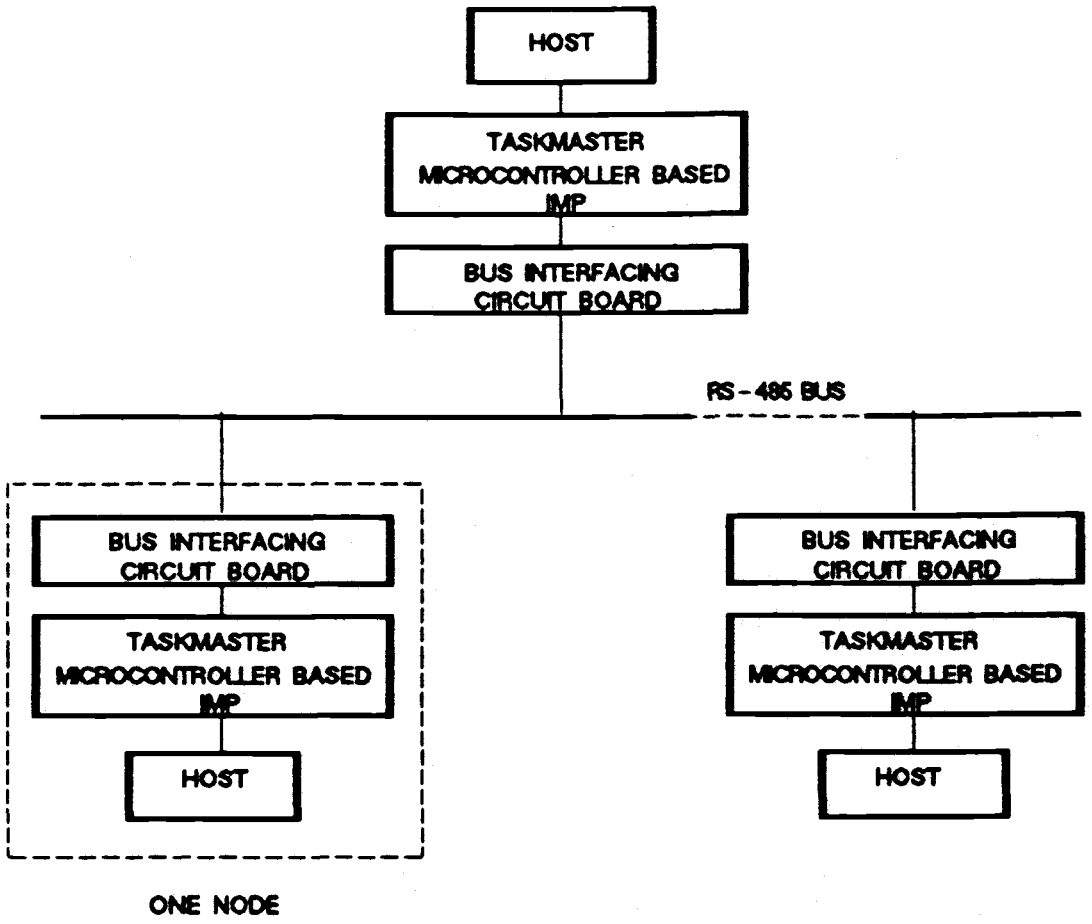


Figure 3-3. COLAN III System Conceptual Configuration.

CHAPTER 4

HARDWARE FOR COLAN III

COLAN III consists of a RS-485 bus and network nodes consisting of bus interfacing circuit boards, SIBEC II single board computers (TASKMASTERs) using an Intel 8031 microcontroller, and IBM PCs (hosts). The system structure is shown in Figure 4-1. Packets issued by an IBM PC are transmitted to its SIBEC II through a RS-232C serial linkage. Two kinds of packet may be issued by a host: (1) the local host command packet, which is equivalent to that used in the TASKMASTER SYSTEM; and (2) the network packet. The network packet is directed through the RS-485 network bus via the bus interfacing circuit board to the intended destination node.

In the following sections, the components of the COLAN III system are discussed in detail.

4.1 System Bus

COLAN III 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 board, which has an RS-485/RS-232C converter.

There are two basic methods for electronic communications between the various components of a communications network: single-ended trans-

missions 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 transmissions uses two signal lines and the signal level is determined by the voltage difference between the two lines. Since unwanted signals 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 III employs the RS-485 as a network bus.

4.2 Bus Interfacing Circuit Board

The bus interfacing circuit board converts RS-232C standard signals to the RS-485 standard, and vice versa. It also continually monitors the status of the network bus, reporting this status to the SIBEC II when needed. The bus interfacing circuit board 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 diode circuit for collision detection, and a retriggerable one-shot circuit (74LS122) which transmits the bus status signal to an 8031 microcontroller on the SIBEC II board.

If there are no data on the network bus, the one-shot sends a low signal to the SIBEC II board, indicating that the 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, one byte in length, indicating that the network bus is busy. Once triggered, the basic pulse width (one byte long for 1200 bps) may be extended if another one-to-zero signal transition is detected. Figure 4-2 illustrates pulse control by retriggering. When a character stream is in transmission on the network bus, the output of 74LS122 remains high until the transmission is completed.

Detection of one-to-zero transition has been adopted for the 74LS122 because there is always a transition at the beginning of a one-byte transmission, due to the zero start bit. Therefore, the retriggerable one-shot (74LS122) is an appropriate device to monitor the bus status of the CSMA/CD protocol. The main reasons for using a retriggerable device are to reserve a time slot during which a receiver may process checksum and acknowledgment following completion of the sender's transmission and to assure continuity of the Bus Busy signal. Since the network bus cannot be accessed by other nodes during reserved time, which must be longer than 1 byte period, the process of sending a packet and receiving acknowledgment from the receiver is an atomic operation. Further details of this process are discussed in Chapter 5.

The port pin P1.2 of the 8031 microcontroller is used to control the 74174 driver on the bus interfacing circuit board. If there is no data to send, the pin turns off the 74174 bus driver, making the output tri-state. When the network bus is free and there is something to send through the network, the pin turns on the driver by emitting a high signal. In this way, a node will not interfere with the network bus when there is no data to be

transmitted. An inverter (7406) has been used to toggle the output voltage level of P1.2, because all the port latches in the 8031 microcontroller have 1s written to them by the reset function or upon power-on.

Two diodes and two resisters are added to the driver side of the board to affect the voltage difference on the receiver side when data collision has occurred on the network bus. This collision detection circuit (see Figure 4-3 for the diagram) may have either of three cases in accordance with its inputs (see Table 4-1), and each case is discussed below:

1) Inputs A and B are both "1" to the RS-485 bus: All diodes are conducting and both outputs of receivers detect the "1" state.
2) Inputs A and B are both "0" to the RS-485 bus: All diodes are not conducting and both outputs of receivers detect the "0" state.
3) Input A is "1" and input B is "0", or vice versa: When the input of A is "1" state and that of B is "0", diodes D1 and D2 are conducting, but diodes D3 and D4 are not conducting. The dominant state of the bus is "1" and thus both receivers A and B recognize a "1" state. This means that there is a collision from the point of view of B. When a collision has occurred, the side which is not dominant can detect the collision by comparing the transmitted data and the received data. By transmitting three more jamming characters after detection of the collision, the other node involved in the collision will be made aware that there has been a collision.

The resisters used in the diode circuit are 13K. When a great many nodes are used on the network bus, it is good idea to increase the resistance value because the total value of resisters in parallel will diminish as the number of nodes is increased.

4.3 SIBEC II

SIBEC II is a single board microcontroller of the MCS-51 family made by Binary Technology, Inc.. The 8031 microcontroller on-chip serial communication port is used to interface with the bus interfacing circuit board. An Intel 82530 serial communication controller (SCC) is placed on the board to connect with an IBM PC serial port. With the SCC interrupt request line connected to the 8031 microcontroller $\overline{\text{INT1}}$, the 8031 external interrupt 1 is used to indicate the presence of an input from the host, the IBM PC. When a character from the network is received, a serial port interrupt occurs and the interrupt service routine for an incoming network packet 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 packets is initiated.

Since the 8031 microcontroller does not have 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 ANDing the program memory strobe $\overline{\text{PSEN}}$ and the data memory strobe $\overline{\text{RD}}$, 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 III system, an 8K ROM (2764), residing at the addresses 0000H to 1FFFH, is used for program memory and two 8K RAMs (6264) are used 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 is reserved for the COLAN III operating system.

Five interrupts are supported by the 8031 microcontroller and in COLAN III, four of the interrupt sources are used by the operating system. The serial port interrupt and external interrupt 1 are used for the network packets and host packets. The Timer 1 supports the on-chip serial port's baud rate and cannot be used for interrupt purposes, while the Timer 0 interrupt is used to support the real-time clock and time-out functions necessary for acknowledgment processing. The only unused interrupt is external interrupt 0, which is left for future modifications and development.

Characters transmitted from the host are sent to COLAN III in standard ASCII format at the data rate of 1200 bps. This baud rate is not the one intended for COLAN III and will be changed to 9600 bps in a future revision. Serial communications between two nodes are also at 1200 bps, which can be increased to 9600 bps.

The 8031 serial port is full duplex and can transmit and receive simultaneously [INTE 83]. It is also receive-buffered, meaning it will commence reception of a second byte before a previously received byte has been read from the receive register. This feature is used to detect collisions while transmitting a packet by saving every character before sending, receiving it back through the network bus, and comparing both characters. The serial port can operate in 4 modes and COLAN III uses mode 3 which has a special provision for multiprocessor communications. When the sender wishes to transmit a packet to a destination node, it first sends out a destination address byte to identify the target node. A destination address byte differs from other data bytes in that its 9th bit is 1, while the 9th bit of other data bytes are 0. On the 8031 microcontroller, SM2=1 prevents the interruption of nodes by a data byte with the 9th bit at 0. A destination address byte with a 9th bit at 1, however, will interrupt all nodes. This allows each node to examine the received destination address byte to see if

it is the target node. The addressed node clears its SM2 bit and prepares to receive the data bytes which are to follow. The nodes that were not addressed leave their SM2 bits set and ignore the following data bytes.

Other features of the SIBEC II communication node include a green LED, added to indicate the existence of mails on the queue and to blink for a random period when a data collision has been detected. An Intel 8255 I/O chip on the SIBEC II board is used as an interface to the 6 bits of an internal switch bank (commonly called a DIP switch). The 6-bit switch, used to set the node address, allows an address range of 0 to 63 in decimal or 00 to 3F in hexadecimal.

4.4 Host

A host running a high level language program, such as a BASIC or PASCAL program, serves as a user in COLAN III. 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 sending mail and receiving mail. Control and operation of the system is accomplished by sending a command from the host keyboard or through the host application program, which initiates and controls the running of specific tasks.

In general terms, any computer using a conventional asynchronous serial protocol can be used as a host. In this study, IBM PCs were used as hosts.

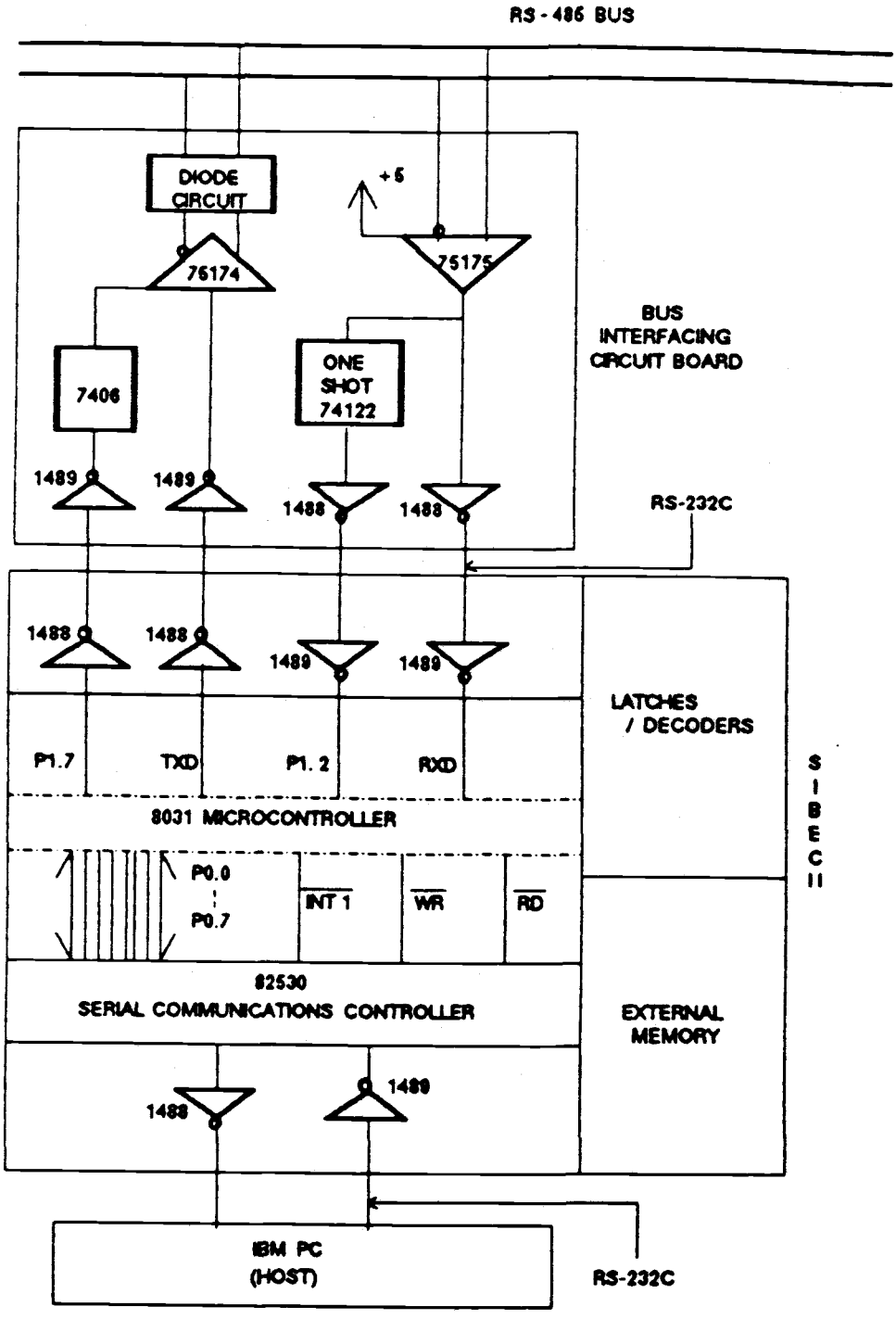


Figure 4-1. COLAN III System Structure (One Node).

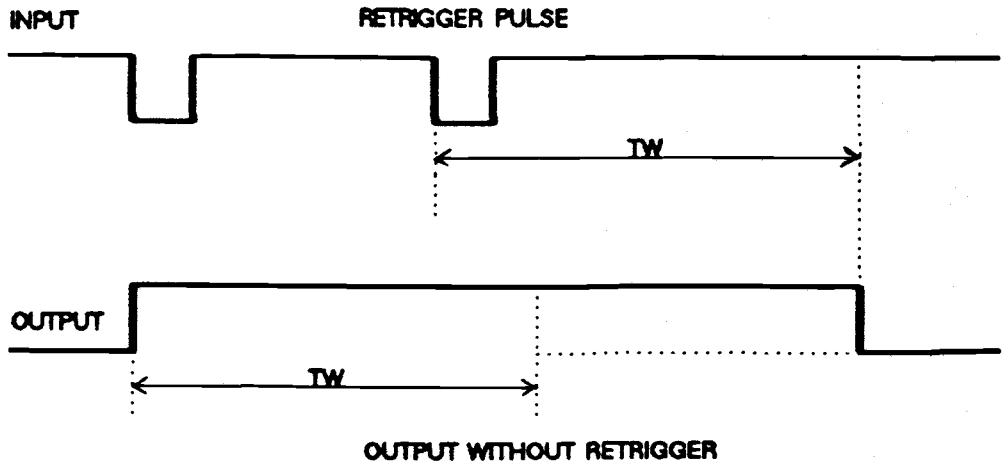


Figure 4-2. Input/Output Pulses of Retriggerable One-Shot.

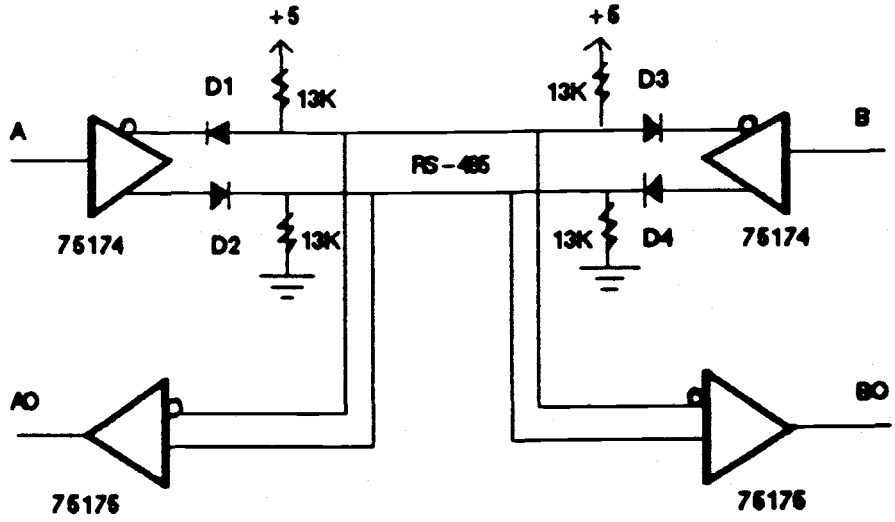


Figure 4-3. Diode Circuit for Collision Detection.

Table 4-1. Collision Detection Chart.

INPUT		OUTPUT	
A	B	AO	BO
0	0	0	0
0	1	1*	1
1	0	1	1*
1	1	1	1

*: COLLISION HAS BEEN DETECTED

CHAPTER 5

LOW LEVEL SOFTWARE FOR COLAN III

In this chapter, the low-level software design for the COLAN III system is discussed. The first section describes the general nature of the operating system and the remaining sections are devoted to specific aspects of the operating system. Since the COLAN III system is based on the original TASKMASTER SYSTEM, the local host command packet processing part of the latter system was retained.

5.1 General Description

The operating system is responsible for the proper overall operation of the system, including bus monitoring, collision detection and backing-off, broadcasting, checksum and acknowledgment processing, local host command packet processing, incoming network packet processing, outgoing network packet processing, queue management, and the task library.

In order to understand the operating system, it is useful to define certain of its aspects, such as format definitions, memory assignments, and flag definitions.

5.1.1 Packet Format

There are two types of packets in COLAN III: the local host command packet, which is identical to that used in the original TASKMASTER SYSTEM (discussed in chapter 3.1), and the flexible network packet, which will be discussed in this section.

Local host command packets are designed for users to initiate tasks associated with the local TASKMASTER. The network packets are designed to allow users to transfer and receive mail and remote commands among the nodes. This packet has a format similar to that of Ethernet and is expressed in the following general form:

[DA SA {command} OD]

where

"[" is the Starting Delimiter of Network Packet;

"]" is the Ending Delimiter of Network Packet;

"DA" is the Destination Address Field (2 ASCII characters);

"SA" is the Source Address Field (2 ASCII characters);

"{command}" is the Optional Command Field; and

"OD" is the Optional Data Field (maximum length of 4K).

After the complete reception of this packet from the host, the transmitting TASKMASTER node obtains the checksum by exclusive-ORing all of the characters inside the starting and ending delimiters, except for the destination address, attaching it at the end of the packet prior to the ending delimiter. The transmitting node also converts the packet into the appropriate sending format prior to transmission. The converted format is as follows:

DA [SA {command} OD CK]

where CK is the checksum of exclusive-ORing all of the bytes inside the delimiters.

The destination address is a one-byte hex number converted from two ASCII numbers which is directed to interrupt every node on the network bus to identify the target node. If the destination address is 00H (broadcasting), every node receives and serves the rest of the packet. The

source address is two ASCII characters, which informs the intended destination node of the source of the packet.

The optional command, when included, informs the destination node what to do with the data which follows. The data may be directed to the screen of the destination host computer (TASK26), or to the printer attached to the destination TASKMASTER node (TASK27), as specified by the included optional command. The data may be either a short message typed directly on the screen or the contents of a file. The optional command field has the same format as that of local host command, except that it does not include the device number field. There is no need to specify the device number in the optional command because it has already been included in the network packet in the form of the destination address. Spaces inside the command packet are ignored. The format of the optional command is as follows:

{P TT Q DD}

where

"P" is the Pre-Execution Control Character;

"TT" is the Task Number;

"Q" is the Post-Execution Control Character; and

"DD" is the Command Data Field (optional).

The network packet may take any of three possible forms. The first is a general format, including both an optional command field and an optional data field as described above. The second is a format which does not include the optional data field and the last is a format which does not include the optional command field. The functions of general format and its two variations may be described as follows.

5.1.1.1 Both Data and Command Field Included, General Format. In this format, the user of the transmitting node specifies the action to be

performed with the data he is sending. In the present version of COLAN III, this action may be either printing the data on the screen of the receiving host (TASK26) or directing the data to the printer of the receiving TASKMASTER (TASK27). Future tasks associated with mailing activities may be added easily with this format, since simply adding the task to the task library and specifying its task number in the optional command field will cause the implementation of the desired action at the intended destination node with the data which follows the task designation.

5.1.1.2 Optional Data Field Not Included, Remote Command Format. The remote command packet has the following format: [DA SA {command}]. To send a command to another node for a real-time control purpose, a user needs to specify the starting delimiter, destination address, source address, command to be executed at that node, and the ending delimiter. The command is executed at the intended destination node. Remote command formats are associated with control-oriented tasks (TASK0 to TASK25), while the general command formats are associated with mail-oriented tasks (TASK26 and TASK27).

5.1.1.3 Optional Command Field Not Included. In this format, the user of the transmitting node does not specify the action to be performed and merely transmits data which is to be saved on the incoming mail queue of the receiving node. When the receiving node accepts this format, the green LED on the destination TASKMASTER board is lit, indicating the arrival of mail. The action to be performed with this data depends upon the user's choice of receiving node when issuing the local host command (TASK26 or TASK27). The LED is turned off after the mail is serviced by the user's local host command. The format of this packet is: [DA SA OD]. A user must enter the starting delimiter, destination address, source address, the data, and the ending delimiter.

5.1.2 COLAN III System Data Memory Assignment

All 128 bytes of internal data memory are reserved for the COLAN III operating system. External data memory is used for incoming and outgoing packet queues.

5.1.2.1 Internal Data Memory. The internal data memory map is shown in Figure 5-1. The 128 bytes are assigned as follows:
Register bank 0 is dedicated to queued tasks, register bank 1 for immediate tasks, register bank 2 for timer operations, and register bank 3 for the operating system. The memory from 20H to 2FH is bit addressable and is reserved for flags, dedicated registers of the operating systems, and queue pointers for external data memory. The system stack resides at address 30H to 3FH. The host command packet buffer is from 40H to 4FH and the network command packet buffer is from 50H to 5DH. Location 5EH and 5FH are reserved for transmit and receive checksum, respectively. The memory location from 60H to 7FH is reserved for the task queue.

5.1.2.2 External Data Memory. External data memory from 5000H to 7FFFH is used for storing outgoing and incoming network packets. The external data memory map is shown in Figure 5-2. This block of memory is divided into three queues. The memory location from 4000H to 4FFF is reserved for additional network packet queues in future development of COLAN III.

1) Outgoing Packet Queue (Queue1). The outgoing packet queue is from 6000H through 6FFFH (4K) and is used to buffer the outgoing network packets. Buffering the outgoing network packet is necessary for CSMA/CD protocol because in case of a collision, the packet must be sent again from the beginning. Only the tail pointer of this queue is necessary for this purpose. The pointer registers associated with this queue are:

Q1TLHI: Tail of Outgoing Packet Queue1, Higher Byte;

Q1TLLO: Tail of Outgoing Packet Queue1, Lower Byte.

b) Incoming Packet Queues (Queue2 and Queue3). Because two queues are assigned for incoming network packets, the operating system can handle two network packets simultaneously. Namely, the operating system can direct the first received mail to the screen of the host computer or to printer, while receiving and saving the second incoming packet. Once the mail on one packet queue is served, the queue is immediately available for the next incoming network packet. The operating system uses these two incoming packet queues alternately. Queue2 is from 7000H to 7FFFH (4K) and Queue3 is from 5000H to 5FFFH (4K). The pointer registers associated with these two queues are:

Q2TLHI: Tail of Incoming Packet Queue2, Higher Byte;

Q2TLLO: Tail of Incoming Packet Queue2, Lower Byte;

Q3TLHI: Tail of Incoming Packet Queue3, Higher Byte;

Q3TLLO: Tail of Incoming Packet Queue3, Lower Byte.

5.1.3 Status Register and Dedicated Register

Discussion in this section includes the registers and flags used to control various algorithms in the COLAN III operating system.

1) NSWD. NSWD is a status register for network operation. Each bit of NSWD has the following meaning:

NSWD.7 = 1: save optional command in network command packet buffer;

NSWD.6 = 1: complete incoming network packet has been received;

NSWD.5 = 1: there is packet to send through bus;

NSWD.4 = 1: ready to accept incoming data (it is my address and queue is available);

NSWD.3 = 1: incoming network packet in progress;

NSWD.2 = 0: set TB8 for sending destination address;

NSWD.1 = 1: outgoing network packet in progress; and

NSWD.0 = 1: no acknowledgment for broadcasting.

2) SENSR. SENSR is a sensitize register and the following two bits have been added to the original version of TASKMASTER.

SENSR.1 = 1: incoming packet queue2 sensitized; and

SENSR.0 = 1: incoming packet queue3 sensitized.

3) TIMER. The original TIMER register was reserved for flags associated only with time-related activities, but in COLAN III this register includes additional flags associated with queue management. In COLAN III, the intention was to maintain all resources for the operating system inside the internal data memory.

TIMER.3 = 1: timeout for acknowledgment from receiver is active;

TIMER.2 = 1: switch packet queue for incoming packet;

TIMER.1 = 1: queue2, optional command not included; and

TIMER.0 = 1: queue3, optional command not included.

4) UNITADD. Upon running, the operating system reads its own node address from the address DIP switch and maintains it in this register. A unique node address in this register is used to give the random period required for CSMA/CD protocol implementation.

5.1.4 Operating System Configuration

The operating system consists of four functional parts: the main routine, the serial port interrupt service routine, the external interrupt 1

service routine, and the timer 0 interrupt service routine. Figure 5-3 is a conceptual configuration of the operating system.

From the point of view of the CSMA/CD protocol, bus monitoring, collision detection while transmitting, waiting for the random period in case of collision, and acknowledgment processing of the transmitter side have been implemented in the main routine. Checksum and acknowledgment processing of the receiver side is implemented in the serial port interrupt service routine. Finally, transmit checksum processing for an outgoing network packet is implemented in the external interrupt 1 service routine.

Among the three interrupts used, the timer 0 interrupt, used for a real-time clock and a timeout function for acknowledgment, has a higher priority than others.

5.2 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 5-4.

Within the infinite loop, the main routine opens a window for the serial port interrupt and the external interrupt 1. Then it checks for mail on the incoming packet queues waiting for user instruction. If mail packets are present, the main routine turns on the green LED, indicating to the user the existence of mail on the queue. Otherwise, it turns off the green LED. Next, it sends an outgoing network packet to the network bus, if there is a packet to send and the bus is free, and executes the queued tasks.

Figure 5-5 is the flowchart for the "sending outgoing network packet to network bus" part of the main routine. This is the most important part for implementation of the CSMA/CD protocol, first checking the status of

the network bus by reading the output of the retriggerable one-shot. If the network bus is free, the operating system starts to send the outgoing network packet through the bus, obeying the CSMA/CD protocol. If the network bus is busy, the operating system continues to monitor the network status until it is senses that it is free.

Prior to transmission, the serial port interrupt is disabled to avoid unnecessary interrupts caused by sending or receiving every character. Instead, the polling of the TI and RI flags is used to manage the serial port. The flag SM2 is cleared to receive back every character it sends to detect software collision, whether or not the 9th bit of each character is programmed to be set. Every character to be sent is first saved in the temporary register to compare with the one which is echoed back to the receive buffer through the network bus. If the two characters are different, a collision during transmission on the network bus is indicated. When this occurs, three more jamming characters are sent to assure that the other node involved in the collision is aware of the situation, and the backing-off process is initiated to reinitiate the transmission process. Before retransmitting the whole packet, a random delay (dependent upon its own unit address) is given to avoid further collision. In COLAN III, blinking the green LED a number of times equal to its unique node address has been adopted for the sake of simplicity. If the two characters are the same, then the next character is processed. By means of this software detection algorithm, collision detection while transmitting in CSMA/CD protocol is implemented.

Only the destination address byte is sent with the 9th bit set to 1, because every node on the bus is interrupted to compare the target address byte with their own node address. The remainder of the characters are sent with the 9th bit cleared to 0 in order that only the intended destination

node is interrupted with the characters which follow the destination address byte and other nodes may continue their work without unnecessary interrupts.

After completion of the transmission, the transmitting node awaits acknowledgment from the receiving node for a random period, dependent upon its own address. If there is no acknowledgment from the receiver, the transmitter then begins to retransmit the entire packet after a timeout. If after three trials there is still no acknowledgment, the operating system assumes that the receiver is not ready to accept the packet and reports this situation to the host. If the acknowledgment is received from the receiver, successful transmission is indicated. The operating system reports this status to the host and turns to the next job, which is executing the queued tasks. For broadcasting, for the sake of simplicity no acknowledgment processing has been adopted.

The algorithm designed for collision detection while transmitting is intended to efficiently detect possible collisions between packets put on the network bus at the same time. Acknowledgment and checksum processing is included to correct transmission errors caused by possible noise interference during transmission and for backing up the collision detection algorithm.

Another important design issue involved consideration of the implications of low speed transmissions. Since transmitting characters takes a relatively long time, this action may block the calling of an interrupt. If it is assumed that a packet is input from the host, an interrupt should occur in this situation. If it is further assumed that a transmitting action occurs simultaneously and this action is not taken under the condition that interrupts are enabled, the interrupt for the packet from the host will not occur until the transmission is completed. The consequence is the loss of some input

characters. This has been an issue of concern throughout the design of the COLAN III operating system. For this study, interrupts were enabled whenever there was a transmitting action.

5.3 Serial Port Interrupt Service Routine

The serial port interrupt service routine, a flowchart of which is shown in Figure 5-5, is responsible for collecting, processing, and storing incoming network packets. This service routine may be divided functionally into three parts: main incoming packet processing, checksum and acknowledgment processing, and optional command service.

5.3.1 Main Body

Upon receiving the first character designated to be the destination address, this routine checks if the received address is either a universal address or its own node address. If it is identified as one of these, the availability of the queue is checked next. If no queue is available, the rest of the packet will be ignored. If either one or both of two queues is available, the flag NSWD.1 will be set, which indicates the operating system is ready to accept and serve the rest of the packet. The data following the target address is directed to one of available queues. The optional command packet, if included in the network packet, is saved in both the incoming packet queue and the network command packet buffer for interpretation and service. Spaces in the optional command packet are saved in the incoming network packet queue for checksum, but not in the network command packet buffer.

This routine continues to monitor the character "[" in order that it can at any time reset the queue pointer (backing-off) in the case of a collision. When the input character from the network bus is "]", this will indicate the end of the transmission. Then, checksum and acknowledgment

processing begins with the received data in the incoming packet queue. If the checksum is proved to be good, the next step is to serve the optional command, if included. If the checksum is not good, the backing-off processing will be initiated and the operating system goes into the ready state to accept the entire packet again.

5.3.2 Checksum and Acknowledgment Process

This routine conducts exclusive-ORing of all the characters, including the transmit checksum in the queue which determines if the received packet has been damaged during transmission. If exclusive-ORing of all bytes in the packet is zero, no damage during transmission is indicated. After waiting a few milliseconds for the stop bit of the last input character to be processed in the serial port, this routine sends the acknowledgment character back to the transmitter and rids itself of the unnecessary transmit checksum in the packet. If the checksum is proved to be no good (exclusive-ORing is not zero), it returns to the main routine without giving an acknowledgment to the transmitter. The transmitter will eventually be timed-out for lack of an acknowledgment and will again transmit the entire packet.

While checksum and acknowledgment processing proceed on the receiver side, there is no chance for other nodes to access the network bus. This is because the binary pattern of the last ending delimiter "]" has a one-to-zero transition at the second bit position and the retriggerable one-shots of the other nodes on the bus will send another fresh one-byte-long pulse to their TASKMASTER boards, indicating that the bus is still busy. This one-byte-long period is sufficient for a fast microcontroller on the receiver side to check the checksum and initiate the dispatch of an acknowledgment back to the transmitter. Once the acknowledgment character is put on the bus, the character pattern will itself make the one-shots on

the network generate additional pulses. Consequently, the process of sending a packet and receiving a return acknowledgment in response is an atomic operation and there is no chance for the acknowledgment character to collide with other packets. Moreover, only the transmitter will be interrupted for the acknowledgment character with the 9th bit 0 from the receiver since the transmitter sets the SM2 flag to 0 for the acknowledgment character after the completion of transmission, while the other nodes set this flag at 1.

5.3.3 Optional Command Service Routine

This service routine is similar to that of the TASKMASTER local host command service routine. But serving commands are based on the FCFS ("first come, first served") rule. If either mail-related tasks or any immediate task is already running while the current task is immediate, rather than ignored it the current task will be placed at the end of the task queue.

5.4 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 that used in the original TASKMASTER. In this section, outgoing network packet processing is discussed, the flowchart for which is shown in Figure 5-6.

When a input packet from the host begins with a "[", it is considered as an outgoing network packet. These packets are processed and stored in the outgoing packet queue (Queue1). If an input character from the host is

"]", it is indicated that the entire packet has been received from the host and conversion of this packet into the proper sending format is begun.

The first conversion step is to change the destination address of the two ASCII characters into a one-byte hex address, switching it with the starting delimiter "[". Then, this routine checks if the destination address is a universal address, designated as 00H. If it is a universal address, corresponding flag NSWD.0 is set.

The second conversion step is to calculate the checksum and attach it at the end of packet prior to the ending delimiter "]". The checksum is one-byte long, and resulting from exclusive-ORing all of the characters in the packet except the destination address. Because the exclusive-ORing of characters could be any character, the 8th bit of checksum is set to 1 following exclusive-ORing in order to avoid a possible confusion with the ending delimiter "]" or the starting delimiter "[". This 8th bit of transmit checksum is set to the original 0 at the receiver node before the acknowledgment and checksum processing begins.

5.5 Some Tasks in COLAN III

There are two tasks in COLAN III related to mailing activities and queue management.

5.5.1 TASK26

TASK26 is used to display the contents of the incoming mail queue on the screen of host computer with source identification. This task may be executed by the optional command packet included in the incoming network packet or by the local host command packet. This task manages two queues on an FCFS basis. Namely, the mail which first arrives is displayed first and the next arrived is displayed next. After displaying the

mail, the queue is immediately made available for next incoming network packet.

5.5.2 TASK27

TASK27 is identical to TASK26, except that it directs the contents of the incoming mail queue to the Centronics parallel port for a printer operation.

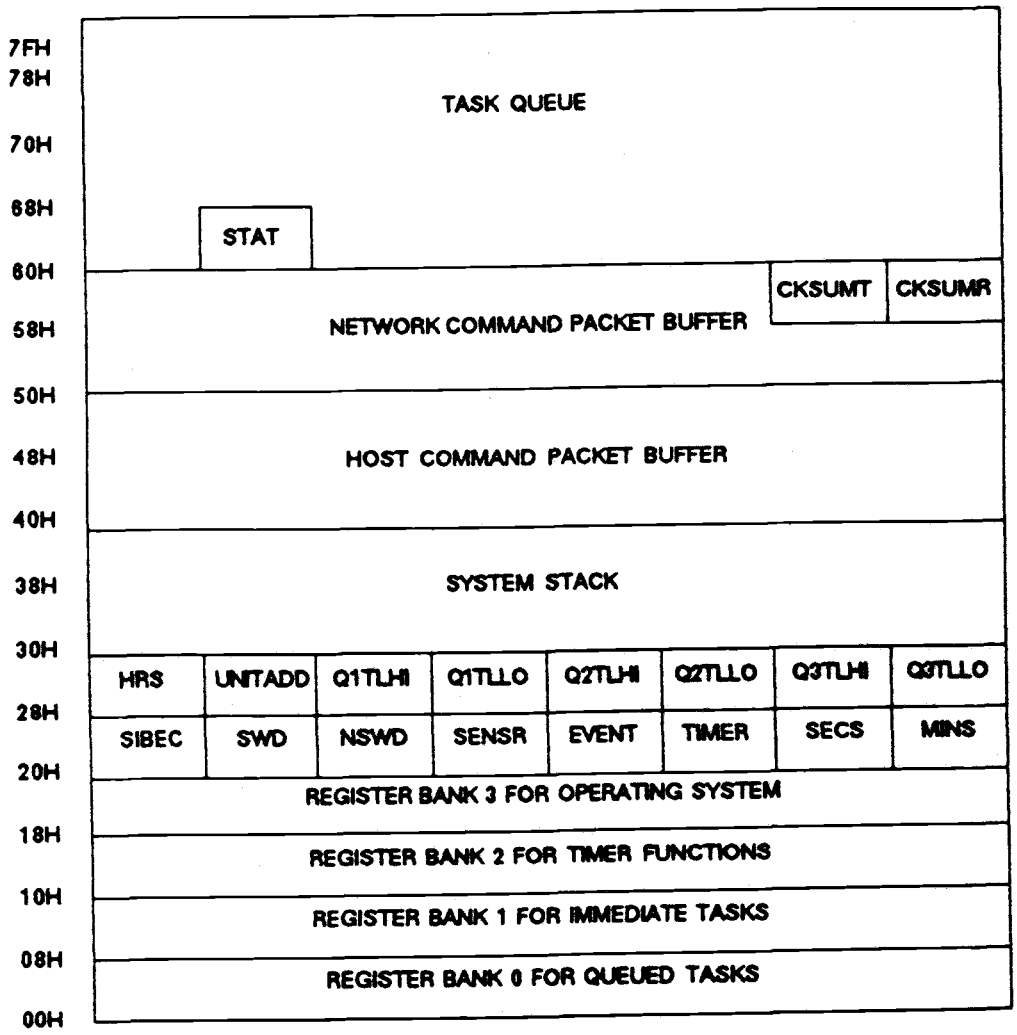


Figure 5-1. Internal Memory Map.

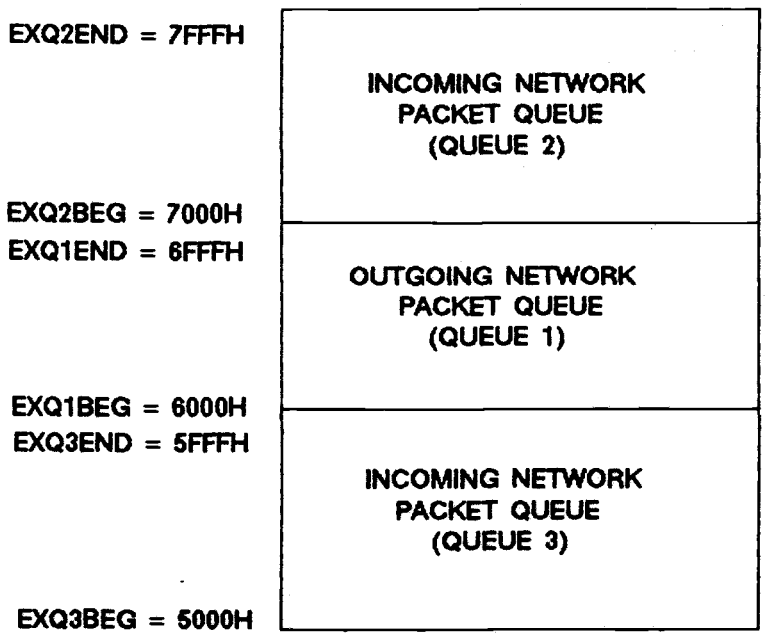


Figure 5-2. External Memory Map.

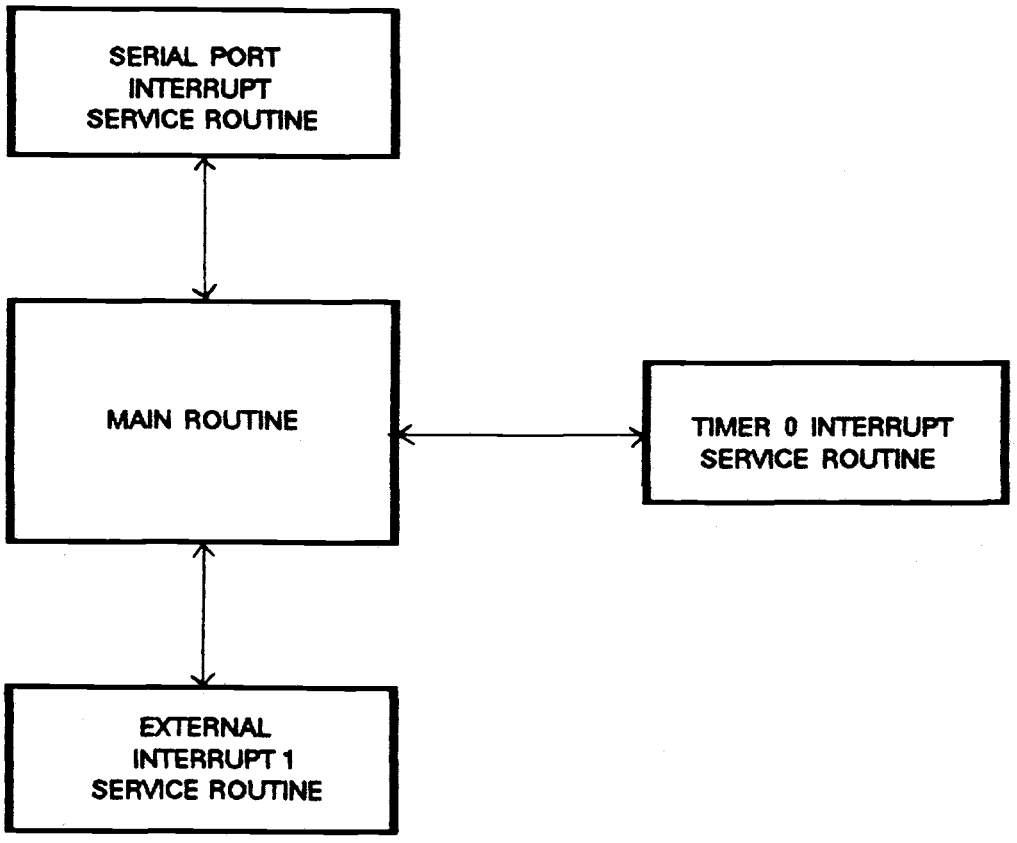


Figure 5-3. Operating System Conceptual Configuration.

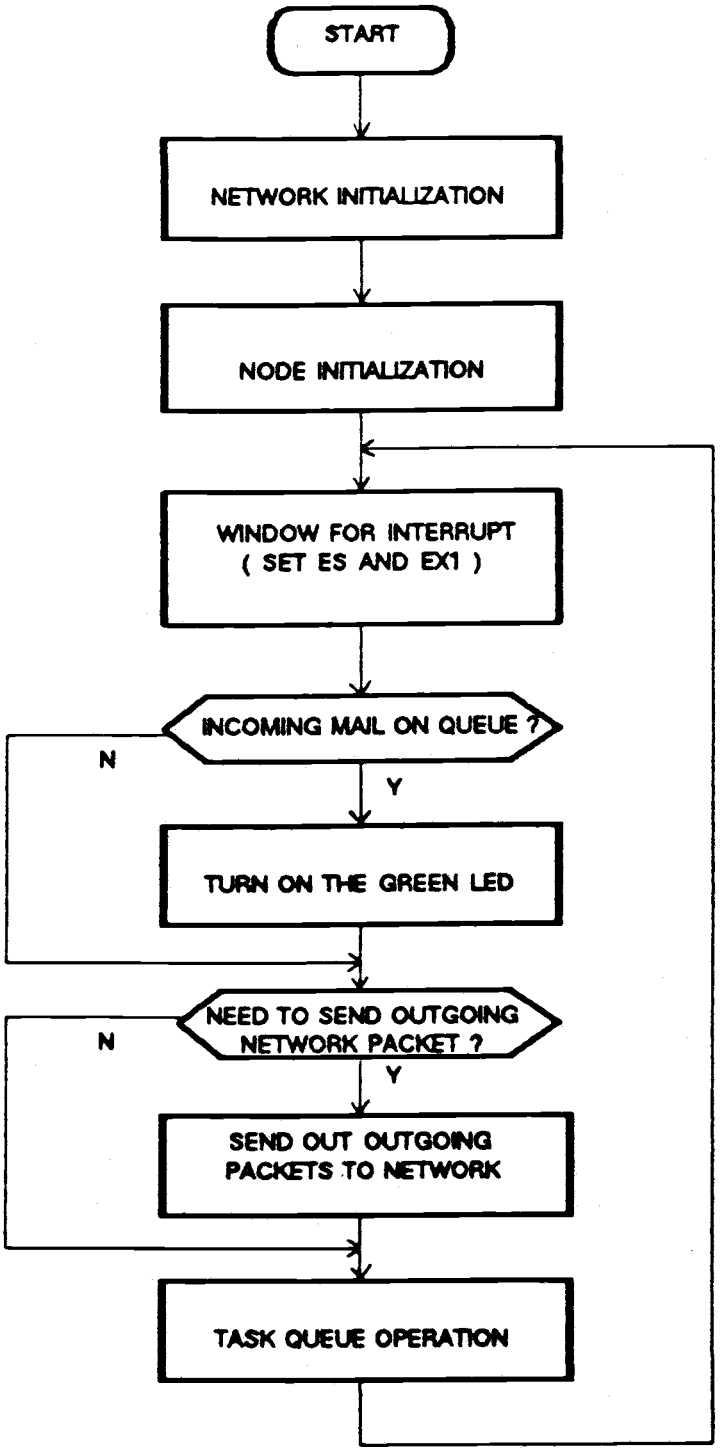


Figure 5-4. Main Routine Flowchart.

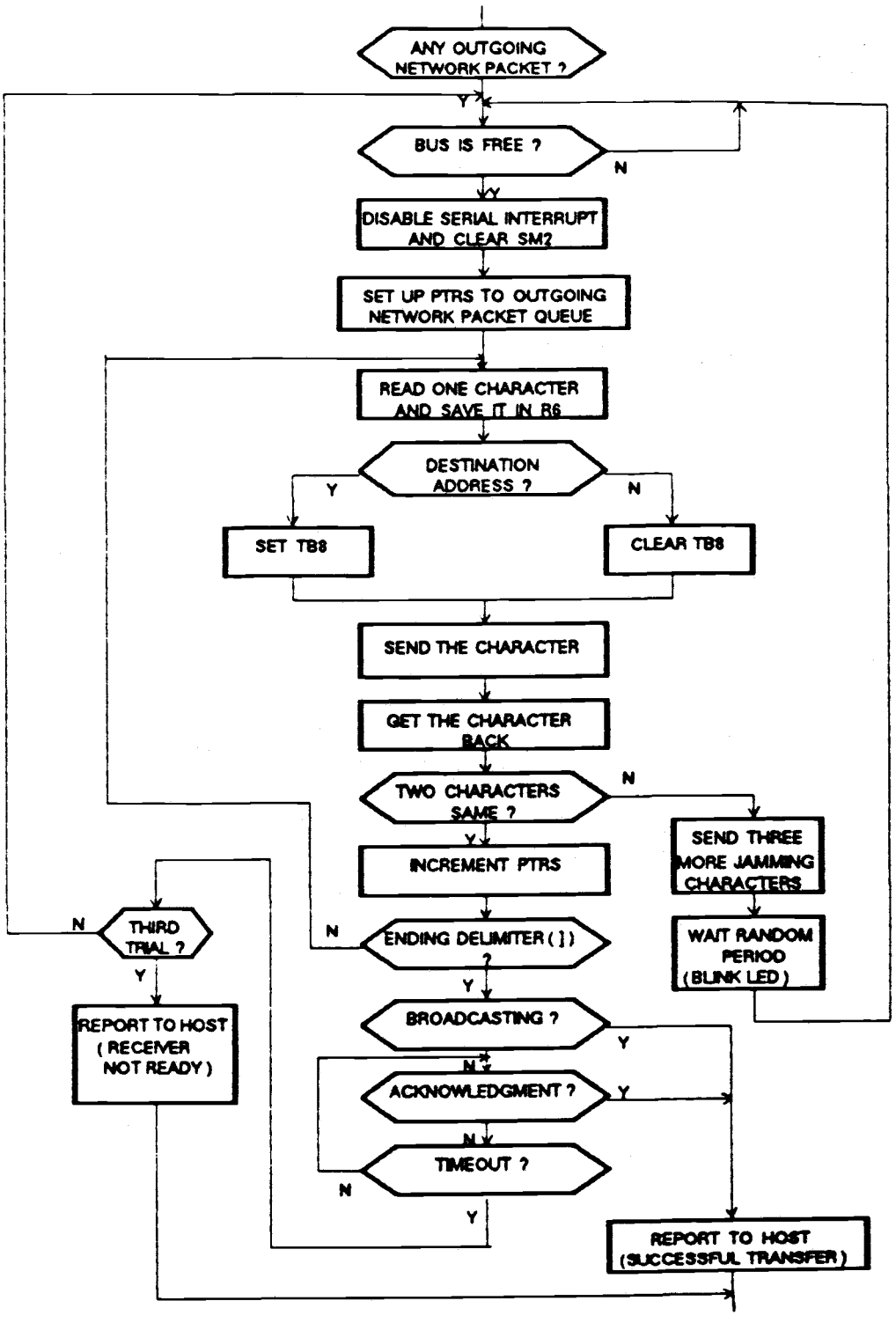


Figure 5-5. Sending Outgoing Network Packets Flowchart.

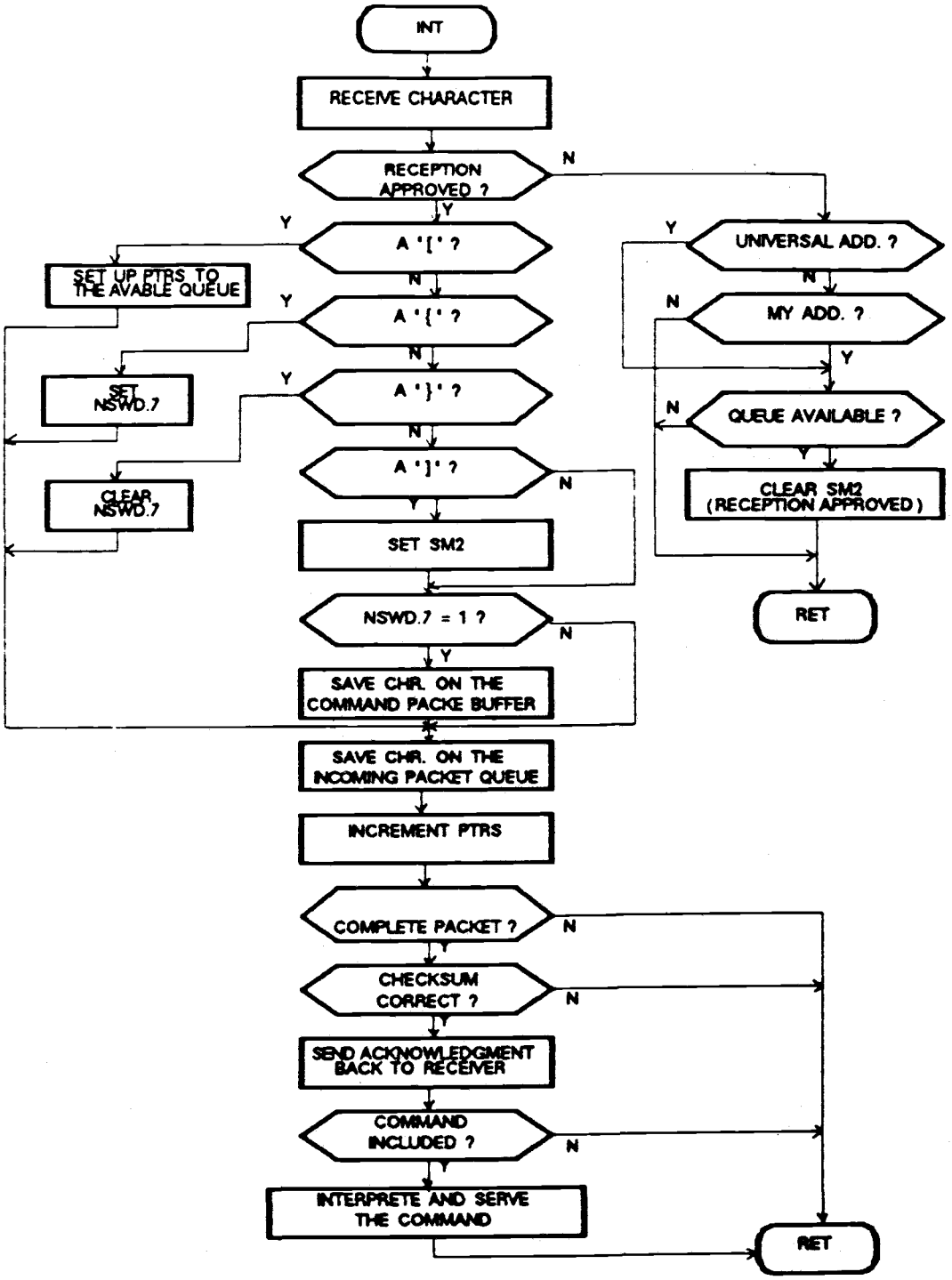


Figure 5-6. Serial Interrupt Service Routine Flowchart.

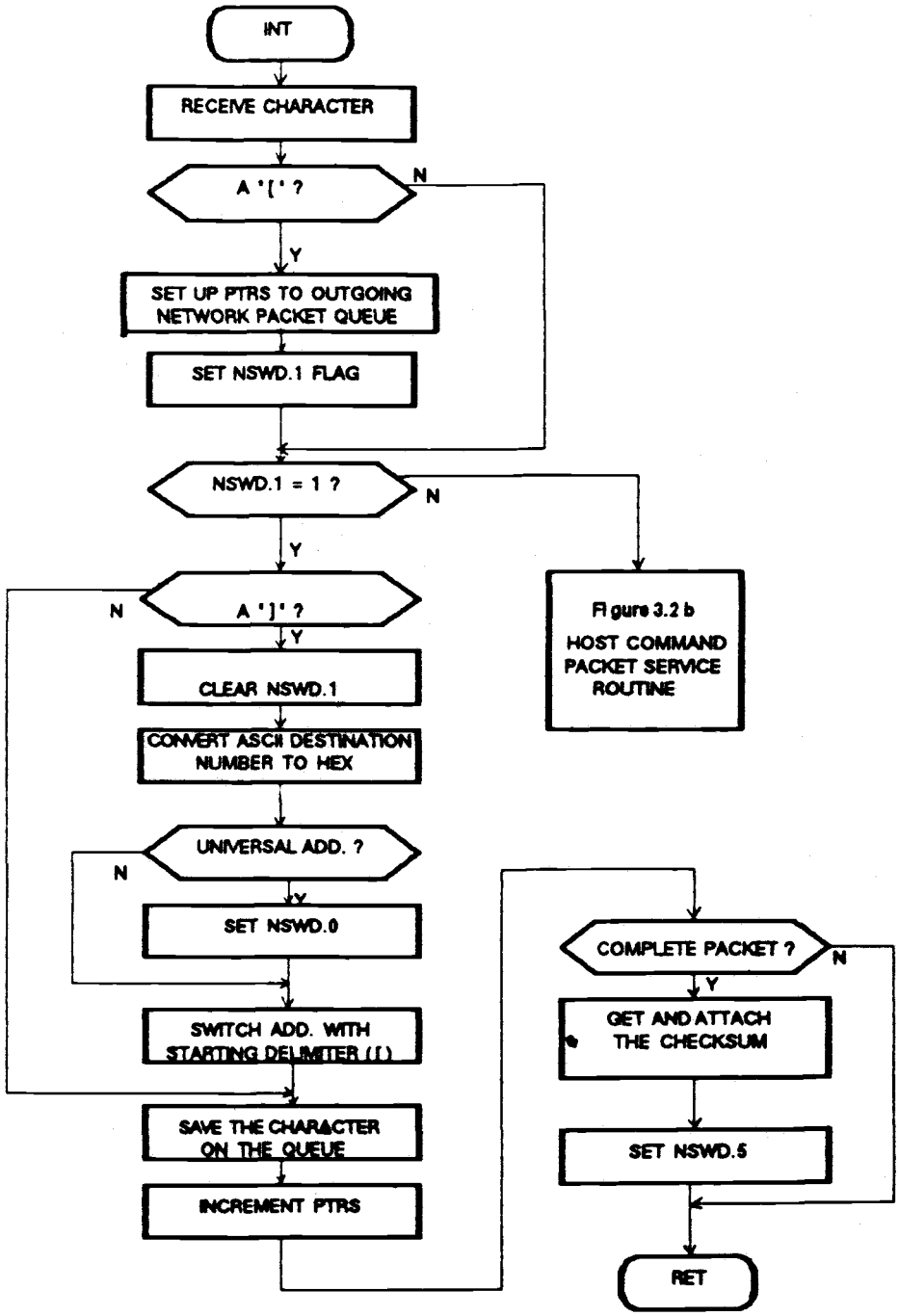


Figure 5-7. External Interrupt 1 Service Routine Flowchart.

CHAPTER 6

HIGH LEVEL SOFTWARE SUPPORT FOR COLAN III

COLAN III includes high level software written in TURBO PASCAL, allowing users to interact with COLAN III at various levels of familiarity. This software provides all of the necessary communications between the host and the TASKMASTER board. The result is a very user-friendly and menu-driven program with which any user can easily follow simple directions to achieve desired actions. Upon running, this high-level software reads the node address from the local TASKMASTER and continuously displays it on the screen so that the user can be readily aware of the address in use.

In the process of its execution, commands are issued to the TASKMASTER using the serial communication port. This is usually as simple as a "print" or "write" type of statement. In response to these commands, the TASKMASTER unit executes the jobs specified by the host.

The further development of this high-level support software is currently being undertaken by Mr. Yong Thye at Oregon State University. Current levels of software support are outlined in the following sections.

6.1 Terminal Mode

Terminal mode is a program which sets up the host computer as a dumb communications terminal. All data transmitted to and from the host is displayed on the screen. This program is particularly useful for purposes of experimentation and development.

6.2 Mail Menu

Mail mode presents a fairly simple user environment for the transmission and reception of electronic mail between hosts through the network. This program is structured in order that the desired mailing action of COLAN III can be specified by means of the menu. It provides an extensive degree of user interaction with the TASKMASTER units, allowing the user to manually step through a command sequence. Through the use of menu selections, the user is able to easily send and receive mail through the network.

There are two choices a user can select in this mode: the mail transmit option and the mail receive option. With the mail transmit option, the user can send mail either by directly entering data with a screen editor or by reading a file that is already on diskette. With the mail receive option, the user can display mail on the screen or save mail in a file, while it is being displayed on the screen. In this option, mail which has already been received by the TASKMASTER board and is waiting to be directed either to the host screen or to the local printer can be processed.

6.3 Remote Command

This program is for transferring a command packet to another unit in order that the task may be executed through that unit. It enables the user to specify a complete description of the control activities to be performed by other units.

6.4 Local Command

The local command routine provides a complete interface between the host and the local node for applications which are exclusively involved with a local TASKMASTER unit. This program also enables the user to specify

a complete description of the control activities to be performed through its own local unit.

6.5 Directory List

This feature lets the user know which files are on the diskettes in drive A or B whenever he wishes to see a list of files.

CHAPTER 7

CONCLUSIONS AND SUGGESTIONS

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

The first version of the COLAN III operating system was developed using two implemented nodes and later tested with three nodes. Every feature of COLAN III worked with three nodes, meaning that the system should logically work with any number of nodes.

7.1 Features of COLAN III

COLAN III retains some of the more desirable characteristics of the TASKMASTER SYSTEM, such as simplicity, low cost, and control-oriented functions. By transforming the TASKMASTER SYSTEM into a LAN, COLAN III has improved the performance and utility of TASKMASTER. More complex control activities, such as coordinating robot arms or parallel processing, can be performed since COLAN III accommodates multiple users, using arbitration methods to share the bus. By adding mailing features to the TASKMASTER SYSTEM, network users are able to exchange information sources. Moreover, COLAN III makes it possible for users to share expensive application resources.

7.2 Recommendations for Future Development

The first version of the COLAN III operating system, using the CSMA/CD protocol, has been successfully implemented. Further development of COLAN III is currently being undertaken at Oregon State University and the following suggestions are provided as a guide to future improvements.

7.2.1 Increasing Baud Rate

COLAN III currently operates at 1200 bps, which is not the transmission rate intended for the network. Since the RS-485 will support a data rate of up to 10 Mbps and the 8031 microcontroller will support up to 62.5 Kbps in mode 3, it is recommended that future versions of COLAN III be revised to transfer data at up to 9600 bps.

7.2.2 Speed Up Transmission Process

The polling method for RI and TI flags has been adopted because it works better than using interrupts in the case of COLAN III's algorithm for the serial port. The outgoing network packet sending routine of COLAN III processes transmissions character-by-character, as described in Figure 5-6. Since this fashion of transmission takes a relatively long time, there is a considerable waste of time in polling RI and TI flags. This could be improved by modifying the current routine, using the following algorithm:

1) Set up pointers to the outgoing network packet queue;
2) Read one character and save it;
3) Transmit the character;
4) Read the next character and save it in another register;
5) Wait for the previous character's TI flag;
6) Transmit the second character;
7) Wait for the previous character's RI flag;

8) Compare the received character and the first-saved character (collision detection process);

9) If there is no collision, wait for the second character's RI flag;

10) Compare the received character and the second-saved character (collision detection process);

11) If there is no collision and characters are not the ending delimiter, go to step 2; and

12) Transmission has been completed.

The collision detection process is the same as outlined in previous chapters and is not fully described in the above algorithm. Since the serial port of the 8031 board is full duplex, it is possible to overlap the transmission process (step 6) and the receiving process (step 7) of one character in order to speed up the transmission of the entire packet while monitoring for collisions.

7.2.3 Segmentation of Network Packet

Currently, the entire packet is transmitted as an atomic unit without segmentation into smaller frames. For future version of COLAN III, it would be useful to segment the packet into equal-sized frames to improve performance and channel efficiency.

7.2.4 Completion of Task Library

Since COLAN III 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.

BIBLIOGRAPHY

ANDE 75 Anderson and E.D. Jensen. "Computer Interconnection Structures: Taxonomy, Characteristics and Examples." ACM Computing Surveys, Vol.7, No. 4, December 1975, pp 197-213.

HERZ 87 J. H. Herzog. *A Design Methodology for Distributed Microprocessors in Real-time Control Applications.* Paper presented at the Second International Conference on Computers and Applications, Beijing, People's Republic of China, June 24-26, 1987.

IEEE 82a IEEE 802 Local Area Network Standard Draft. IEEE Standard 802.3, CSMA/CD Access Method and Physical Layer Specifications, Revision D. December 1982.

IEEE 82b IEEE 802 Local Area Network Standard Draft. IEEE Standard 802.4, Token-Passing Bus Access Method and Physical Layer Specifications, Draft D. December 1982.

IEEE 82c IEEE 802 Local Area Network Standard Draft. IEEE Standard 802.5, Token-Ring Access Method and Physical Layer Specifications, Working Draft. December 1983.

IEEE 83 Special Issue on "Open Systems Interconnection (OSI)-Standard Architecture and Protocols." Proc. of the IEEE, Vol. 71, No. 12, December 1983.

INTE 83 Intel Corporation. *Microcontroller Handbook*, 1983.

KANG 86 Yong J. Kang, John D. Spragins, James H. Herzog, and Hossein Jafari. "A New Approach to Mesh Local Area Network Routing." Proc. of IEEE 5th INFOCOM '86, Miami, Florida, 1986, pp 106-115.

MALH 86 M.K. Malhotra and C. Guy Omidyar. "Trends and Issues in Local Area Networks." IEEE Proc. 5th INFOCOM '86, Miami, Florida, 1986, pp 123-129.

SOI 81 Inder M. Soi and Krishan K. Aggrawal. "A Review of Computer Communication Network Classification Schemes." IEEE Communications Magazine, Vol. 19, No.2, March 1981, pp 24-32.

STAL 84 William Stalling. Local Networks. Macmillan Publishing Company, 1984.

TANE 81 A. S. Tannenbaum. Computer Networks. Prentice-Hall Inc., 1981.

ZHEN 86 Y. P. Zheng, "A Simple Local Area Network, COLAN (Control Oriented Local Area Network)." Unpublished master's thesis, Oregon State University, Corvallis, OR, December, 1986.