1-31-1991

A machine vision system with remote host

Chieyeon Jason Chen
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

Title of Thesis: A Machine Vision System with Remote Host

Cheycon Jason Chen, Master of Science in Electrical Engineering, 1991

Thesis directed by: Dr. Anthony Robbi

Department of Electrical and Computer Engineering

The modem plays an important role in data communication using the Public Swiched Telephone Network. In this thesis, the modem funtion is merged into a low cost machine vision system based on the Motorola 68HC11 Evalation Board. The system comprises the intelligent machine vision controller and a remotely located host computer, such as an IBM-PC. It supports extra long distance communication capability so that the controller and the host can be anywhere in the world where there is a telephone line. Because of the limit of a 2400 bps modem, software data compression is implemented here to save transmission time.

A MACHINE VISION SYSTEM WITH REMOTE HOST

by

Chieycon Jason Chen

Thesis submitted to the faculty of the Graduate School of
the New Jersey Institute of Technology in partial fulfillment of
the requirements for the degree of
Master of Science in Electrical Engineering
1991

APPROVAL SHEET

Title of Thesis: A MACHINE VISION SYSTEM WITH A REMOTE HOST

Name of Candidate: CHIEYEON JASON CHEN
MASTER OF SCIENCE
IN ELECTRICAL ENGINEERING, 1991

Thesis and Abstract Approval:

Dr. Anthony Robbi, Advisor — Date
Associate Professor

Signatures of other members of the thesis committee

Dr. C. Manikopoulos — Date
Associate Professor

Dr. E. Hou — Date
Assistant Professor

VITA

Name: Chieyeon Jason Chen

Address:

Degree and date to be conferred: M.S.E.E., 1991

Date of birth:

Place of birth:

Collegiate institutions attended	Dates	Degree	Date of Degree
National college of Marine Science and technology, Taiwan, R.O.C.	9/76-5/80	B.S E.E.	May, 1980
New Jersey Institute of Technoloy	9/88-1/91	M.S.E E	Jan, 1991

Major: Electrical and Computer Engineering

Position held: Engineer

ACKNOWLEDGEMENT

I am very grateful to Dr. Anthony Robbi, my advisor, for his continued support and guidance during this thesis. I would like to thank Dr. Manikopoulos for his providing research equipment. Mr. Bob Dillon helped the team to write the VGA program (called IMAGE) which shows the CCD image and the status of the analog input levels and control switches. Thanks also go to Mr. Xiao-Fang Qi, Mr. Mirko Delgado, and Mr. Hsueh-Chung Feng for their help.

Contents

1 Introduction **1**

1.1 Role of the Modem in Machine Vision Systems 1

1.2 Machine Vision System Developed at NJIT 2

1.3 Thesis Description 2

2 General System Description **4**

2.1 System Profile 4

2.2 M68HC11 Evaluation Board 5

2.3 Introduction to Modem Interface 7

3 Introduction to Modem Theory **9**

3.1 Modem Fundamentals 9

3.1.1 Modulation 10

3.1.2 Connection Modes 12

3.1.3 Channel Usage Conventions 14

3.1.4 Frequency Modulation 15

3.1.5 Phase Modulation 16

3.1.6 Two-level Phase Shift Keying 17
3.1.7 Quadrature (Four-Phase) PSK 18
3.1.8 Differential Phase Shift Keying 19
3.1.9 Quadrature Amplitude Modulation 20
3.2 Introduction to Modem Chip 22
3.2.1 Description 22
3.2.2 QAM Modulation/Demodulation 23
3.2.3 Passband Filters and Equalizers 23
3.2.4 Parallel Bus Interface 23
3.3 Serial Input / Output 24
3.4 Introduction to Hayes modem 24

4 System Design **26**

4.1 Hardware Design 26
4.1.1 Memory Decoder And Modem Decoder in 68HC11 . . 27
4.1.2 Modem Chip Hardware Interface with 68HC11 and DAA 29
4.2 Software Design 35
4.2.1 Controller Firmware Design 35
4.2.2 PC Side Software and Firmware Design 44
4.2.3 Startmodem Command used in the thesis 49

5 System Analysis of Limits and Performance **52**

5.1 Image transmission 52
5.2 Data compression 53
5.2.1 Introduction to Data Compression Algorithm 53

6 Conclusions **57**

6.1 Conclusion 57
6.2 Suggestions for future work 58

A Circuit Diagrams 60

B Flowchart for Microcontroller Software 63

List of Figures

2.1 System Hardware Block Diagram 5

3.1 The Bandwidth of a Public Telephone Line 10

3.2 The Effects of Bandwidth on Square Digital Signals 11

3.3 Amplitude Modulation 12

3.4 The Simplex Connection 13

3.5 The Half-Duplex Connection 14

3.6 The Full Duplex Connection 15

3.7 FSK Modulation 16

3.8 Tone Assignments for Bell 202 Half-duplex, FSK 1200 bps Modem 17

3.9 Two-level Phase Shift Keying 18

3.10 Quadrature Phase Shift Keying 19

3.11 CCITT-style QAM Modulation for 4800 bps at 600 Baud. . . 21

4.1 Block Diagram 27

4.2 EVB New Memory Map Diagram 30

4.3 Modem Timing Diagram 31

4.4 68HC11 Expansion Bus Timing Diagram 31

4.5 SSI73K224 Pin Description 33

4.6 DAA Pin Description 31

4.7 CCITT V.22 Connect Handshake Sequence 42

5.1 A picture transmitted without data compression 51

5.2 A picture transmitted with data compression 55

5.3 data compression formats 56

A.1 Modem circuit diagram 1 / 2 61

A.2 Modem circuit diagram 2 / 2 62

B.1 Flowchart for Microcontroller Software 61

List of Tables

3.1 Bell 103 vs CCITT V.21 300 bps, Full-duplex Tone Assignment 17

3.2 Tone Assignment for the Bell 212A 20

Chapter 1

Introduction

1.1 Role of the Modem in Machine Vision Systems

With the frequent usage of computers and especially PC's, data communications are of importance. Communication between the various computer systems and terminals is frequently accomplished by means of the Public Switched Telephone Network (PSTN). The essential element for data communication is the modem, which connects computer systems and terminals by means of the telephone network. The modem modulates the serial binary digital data into an analog form that can be transmitted across a telephone line, and it also demodulates the analog signal from the telephone line into the binary digital data.

In last few years the charged-coupled device (CCD) has become an important image sensor in real world industrial applications and commercial markets. In this design, these two elements are combined with some peripheral hardware, firmware and software so that images, analog data and commands can be transmitted between the vision system and a host computer wherever

a telephone line is installed. The 2400 baud modem is currently a popular, cheap, and high speed modem. Because of the speed limit, it takes several minutes to transfer an image of modest resolution. With the help of software data compression, we may shrink the transmission time.

1.2 Machine Vision System Developed at NJIT

This thesis continues the work set forth in three previous theses. Mr. Li first implemented a low-cost intelligent CCD, with the M68HC11 microcomputer TC211 CCD chip [1]. Mr. Feng developed a high speed local transmission interface (2 M bits synchronous transmission) and wrote software for a host computer so that an image could be displayed and manipulated on a monochrome monitor the NMVS system [2]. Mr. Qi expanded the camera memory to three banks of 32K bytes each, modified and improved Mr. Li's hardware and software so that the CCD could take pictures [3]. Mr. Qi also added a RGB color wheel in front of the camera to capture the filtered images and added 9 channel analog input and four on/off switch controls to the system. This thesis describes mainly the modem part of the system. This system will be utilized by the biotechnology research group at the New Jersey Institute of Technology.

1.3 Thesis Description

There are many modem applications in today's market. Combined with the earlier machine vision system, the modem provides an extra-long distance

remote control. This system has a controller side and a PC side. On the controller side, the low cost 68HC11 evulation board (EVB) is used to control a medium resolution CCD image sensor and a 2400 baud modem chip On the PC side, a personal computer PC XT/AT with a VGA monitior and a 2400 baud modem card is used. The EVB has an M68HC11 single chip microcontroller from Motorola. The CCD sensor (TC211) is a 165×192-pixel monochrome image sensor from Texas Instruments and the 2400 baud single chip modem (SSI224L) is from Silicon Systems. This design provides information about using a 68HC11 to control modem and CCD chips. In addition, the modem part can work with existing hardware and software without conflict. This system can be locally controlled by a high speed synchronous port or by lower speed, but distant, remote control through a modem, or by both means

Chapter 2

General System Description

2.1 System Profile

In this chapter, we introduce the overall system operation briefly, emphasizing the parts developed in this thesis. The complete vision control system with a data logger discussed in this thesis has the following basic capabilities: image grabbing and storage, analog signal input channels, digital control outputs, local high speed communication and long distance remote communication with a host computer. Figure 2.1 shows the overall system hardware configuration.

The camera and the image grabber are connected by a 9-pin short cable, which conducts the level shifted pulse trains to shift out the image data and return the analog pixel signals down to the grabber. The host PC can generate a color image from three filtered images derived by a rotating RGB filter. The system has three infra-red sensors so that it can determine the angular position of the RGB wheel.

The optional 2 M bit per sec local connection between the image controller

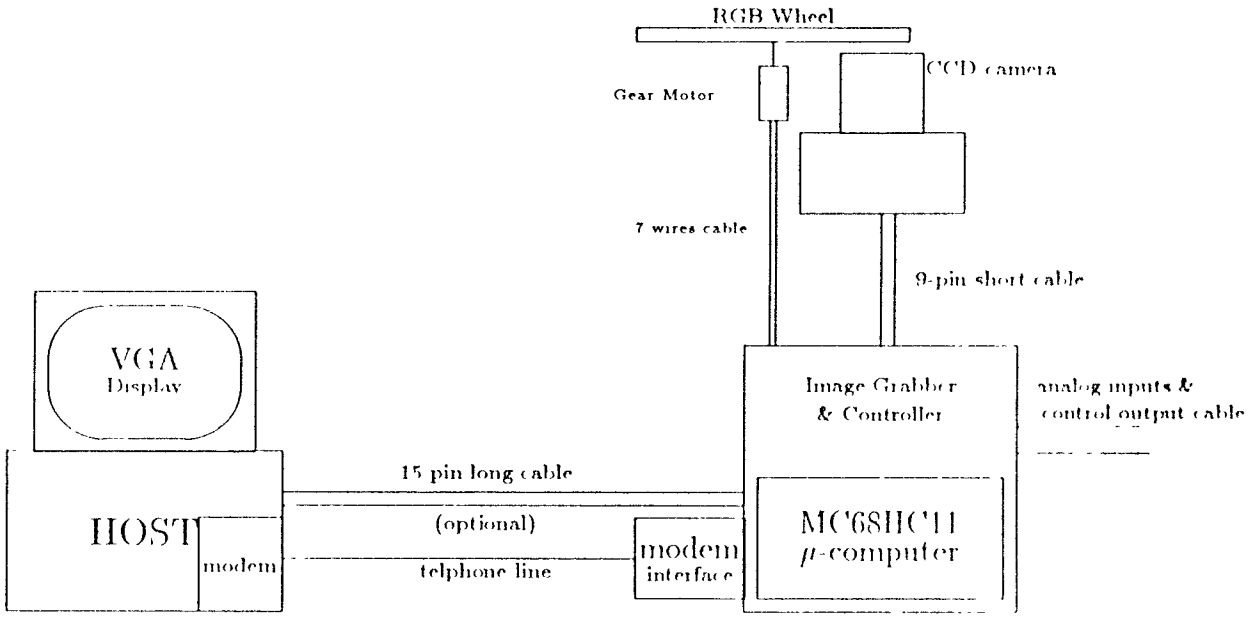


Figure 2.1: System Hardware Block Diagram

and the PC add-on card is a 15 pin long cable, which provides a digital, serial channel for transmitting image data from the controler to the PC add-on card and receiving commands from the PC add-on card . Also, the system supplies a long distance communication capability through a modem interface so that the host PC and image controller can be placed anywhere as long as there is a telephone line installed. The two communication modules can work together. Depending on the controller software and the related software on the PC, the system can be in a local high speed mode or remote long distance mode. In this thesis, we concentrate on modem interface design

2.2 M68HC11 Evaluation Board

The image controller of Figure 2.1 is based upon the M68HC11 Evaluation Board [9]. The information in this section describes that part of the evaluation board related to the modem interface design. The M68HC11 is a

low-cost MCU (microcomputer unit) made by Motorola Corp [11] [12] It can have 512 bytes EEPROM, and 512 bytes RAM on chip, plus several I/O peripherals. In an expanded mode, it can access up to 61K external memory It is one of the Motorola microcontroller product series and can execute not only all of the M6800 and M6801 instructions but also 91 new opcodes [10]. It is a memory mapped I/O processor. All of the on-chip peripheral I/O addresses are reserved from \$1000[1] to \$103F. The system interrupt vector jump table is from \$C4 to \$FF.

This MCU supplies several I/O peripheral functions. It makes an ideal chip for low-cost and small application controller. It has the following I/O functions on chip:

- one asychronous serial communication interface (SCI),
- one synchronous serial peripheral interface (SPI),
- three timer input captures,
- five timer output compares,
- six channel multiplexed A/D converter,
- every pin of port A, port B, port C, port D (all 8 bits) can be parallel I/O or some specific function.

In this design, we use SCI for the modem and SPI for local high speed synchronous communications. The A/D converter converts the analog image signals into an 8 bit data digital form so that an image can be transmitted.

[1] \$ is Motorola symbol for hexadecimal notation

The 68HC11 supplies several independent parallel I/O pins when not used for a dedicated functions. In Qi's design, PA6, PA5, PA4 generate the required timing for the CCD. PA2, PA1, PA0 are used for the three infra-red sensor inputs. PB0 is used to control the DC motor and PB1, PB2, PB3 are used to control four switch outputs. In my design, PA3 output is used to control the on/off hook of the DAA, and PA2 input is shared between the infra-red sensor and the the measurment of ringing width from the telephone interface.

2.3 Introduction to Modem Interface

A device that can modulate binary digital data into analog form for transmission across the telephone line and also demodulate the analog signal from telephone line into digital binary data is called a modem. A serial port that converts parallel data into serial form, and vice versa, can work with a modem interface so that it is possible for computers to talk with one another through the telephone line. Figure shows the block diagram of modem interface. In this design, we chose SSI 73K224L single chip modem from Silicon Systems and the DS6112 DAA (data access arrangement) from Dallas Semiconductor to to perform the PSTN interface [4] [5].

The DAA is a communication component that provides a "direct connect" telephone line interface. The chip provides high voltage isolation, independent ON/OFF HOOK control, ring detection circuity, and a 2 to 4 wire converter hybrid for use in a modem application. With this integrated DAA, the PSTN interface is simple and reliable.

Chapter 3

Introduction to Modem Theory

This chapter explores the topic of how computer data are transferred across ordinary telephone lines. First, we will describe some of the technical aspects of a modem. Having stated the principles involved, we will discuss how computer and related peripherals interact with a modem to control its behavior

3.1 Modem Fundamentals

Digital signals from devices such as teleprinters cannot be transmitted directly over telephone lines. Because a telephone line is intended to carry human speech, which contains frequencies in the range of 200 - 8000 Hz, its frequency response (bandwidth) is rather limited As a matter of fact, since the goal of telephone networks is not fidelity, but intelligibility, the telephone does not even reproduce voices particularly well. Figure 3.1 shows the bandwidth of the public telephone system.

Now, a square wave can be mathematically analyzed as if it were a com

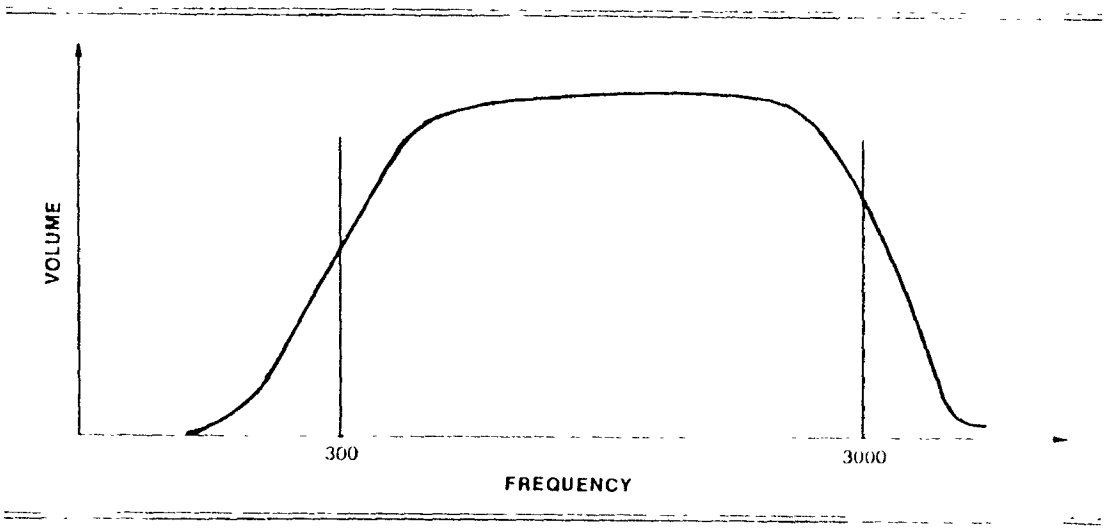


Figure 3.1. The Bandwidth of a Public Telephone Line

posite of sine waves at odd multiples of its frequency. A square wave of 200 Hz, for example, has progressively smaller amounts of 600 Hz, 1000 Hz, and 1400 Hz, and so on. To transmit an acceptable square digital signal the communications medium must have a significantly wider bandwidth than the base frequency of the square wave itself. If the bandwidth is too narrow, then the resulting waves are not recognizable. At the receiver, the logic level of the digital signal becomes ambiguous and communication fails.

3.1.1 Modulation

The problem depicted in Figure 3.2 would occur if we attempted to transmit a computer's square waves over the voice-grade lines of the public telephone network. Clearly, a method is needed to convert data from square waves to a form that can pass unscathed over the telephone network. Sine waves are the best candidate because they can be easily created in a harmonically "pure"

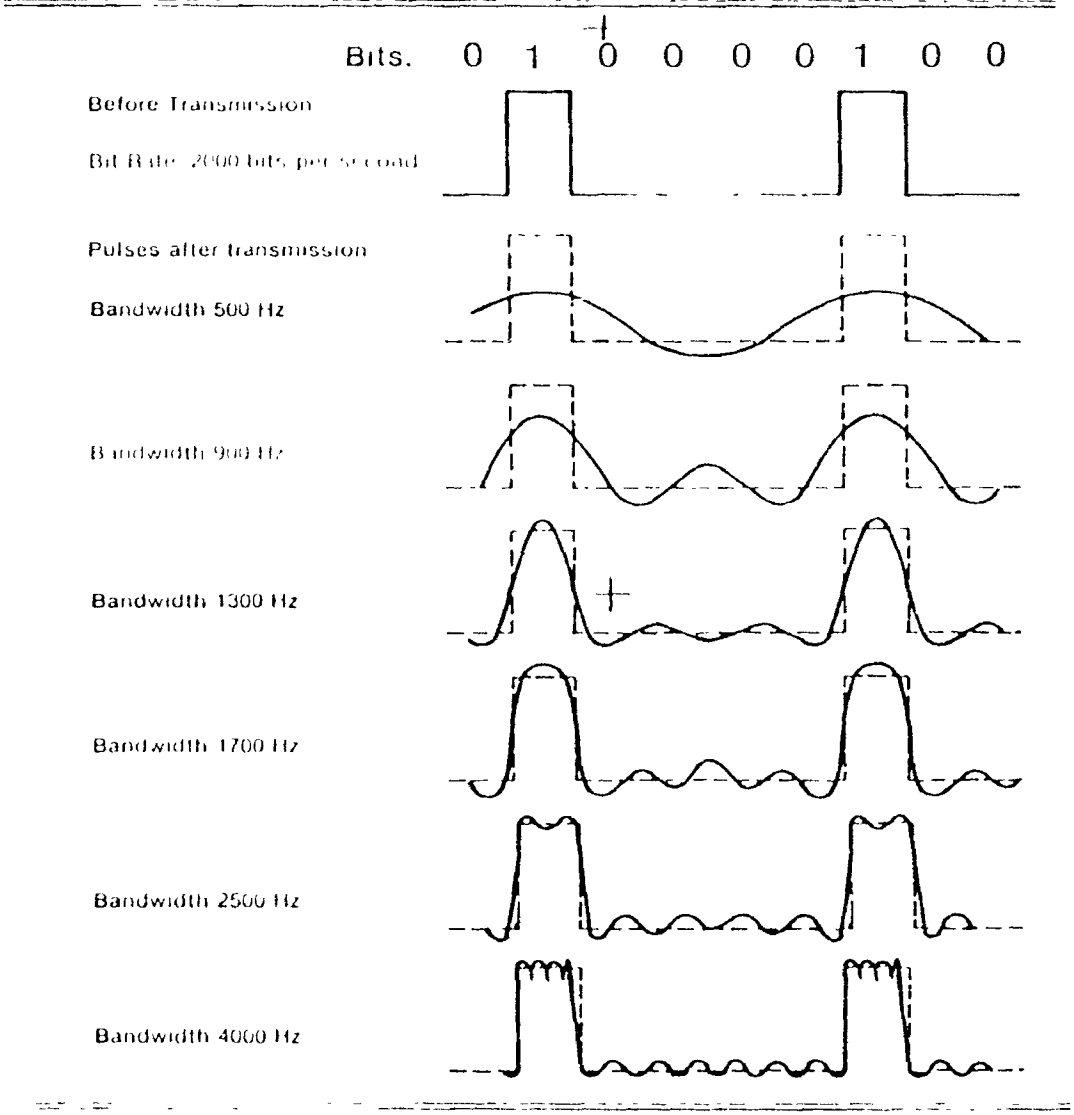


Figure 3.2: The Effects of Bandwidth on Square Digital Signals

form that is less radically affected by the frequency roll off inherent in the telephone line.

The process of encoding one signal with a carrier is modulation and the recovery of the original signal is demodulation. A device that performs both modulation and demodulation on a communication line is known as a *modem*. The simplest form of modulation is amplitude shift keying or *ASK* modu

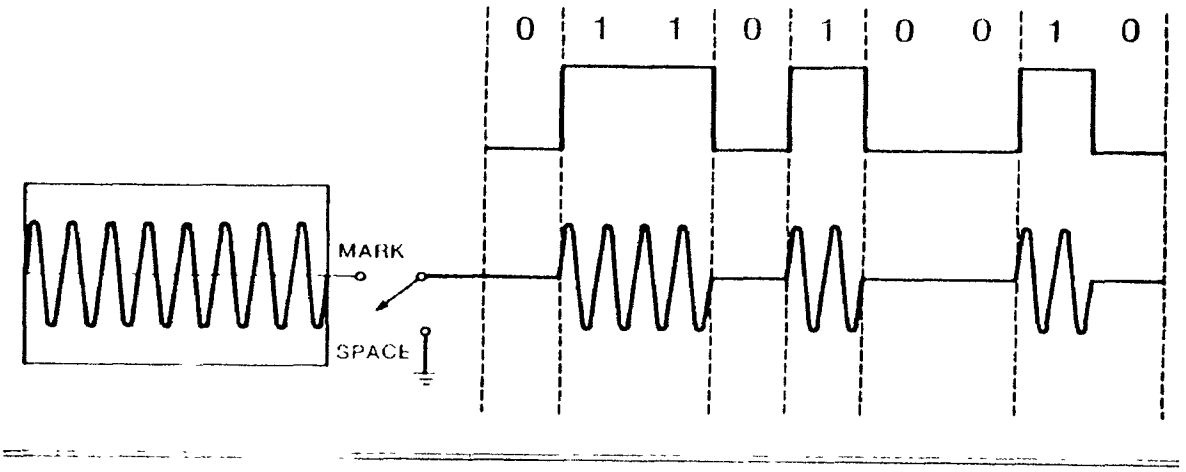


Figure 3.3: Amplitude Modulation

lation, as depicted in Figure 3.3. During transmission, modulation occurs when a single sine wave tone is switched between two amplitudes (volume levels) to represent binary 1 and 0.

3.1.2 Connection Modes

Before we examine more sophisticated modulation techniques, we may review the terminology describing the various ways of utilizing the available bandwidth of a communication line.

The Simplex Connection

A transmitter (such as the simple ASK modulator just described) at one end of the telephone line and a receiver at the other end form a simplex connection — that is, a connection in which data traffic moves in only one direction. This arrangement is illustrated in Figure 3.4. Because they are inherently non-interactive, simplex installations have limited applications.

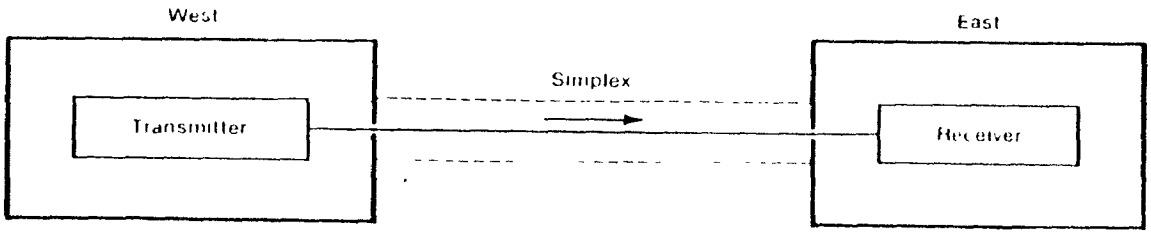


Figure 3.4. The Simplex Connection

The Half-Duplex Connection

A limited kind of two-way system can be built from two simplex connections. As illustrated in Figure 3.5, each modem contains a transmitter and a receiver, one of which is connected to the communications line through a "talk/listen" switch. For West-to-East traffic, modem W connects its transmitter to the line and modem E connects its receiver. This arrangement is reversed for East-to-West traffic. Changing the position of the "talk/listen" switch is called "turing the line around.". The "talk/listen" switch is, of course, not a physical switch but an electronic one controlled by software. The two ends convey the need to turn the line around through half duplex protocols such as the ETX/ACK protocal.

The Full-Duplex Connection

Although the half-duplex arrangement is considerably more flexible than the simplex, it is nevertheless inefficient. The time required to switch the circuitry from transmit to receive mode is commonly on the order of 200 mil

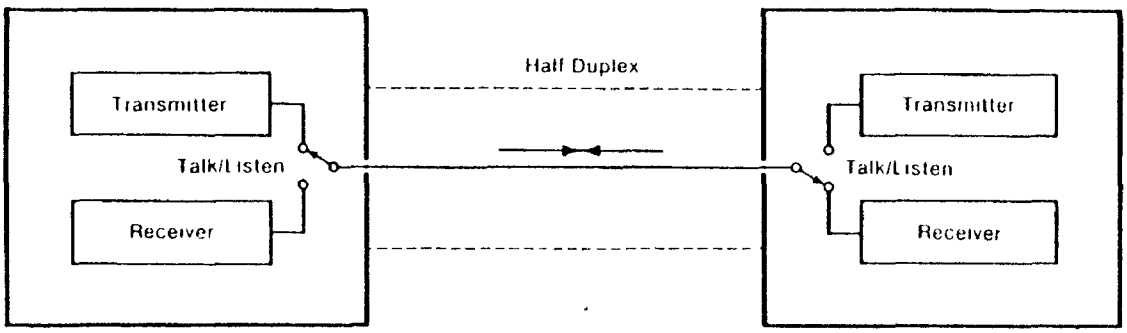


Figure 3.5: The Half Duplex Connection

liseconds. A delay of such duration is intolerable in time sensitive, interactive applications such as remote instrumentation monitoring and control systems. The shortcomings of half-duplex communications can be avoided simply by taking the idea of channel-sharing one step further. Instead of alternately sharing receiver and transmitter, each end of the connection contains both a transmitter and a receiver — one pair for communicating East-to West and another for West to East. The full-duplex connection is shown in Figure 3.6. To prevent interference, a seperate distinct tone is assigned to traffic in each direction. The general idea of dividing a communications channel into smaller frequency bands is called frequency division multiplexing or FDM.

3.1.3 Channel Usage Conventions

Since a full-duplex connection consists of two simplex channels in use simultaneously, modems operating in full-duplex mode must agree upon how they will allocate the signals. For example, the modem might agree beforehand that the modem that uses the higher-frequency channel should be called a Girl modem, while those that use the lower are Boy modems. The problem

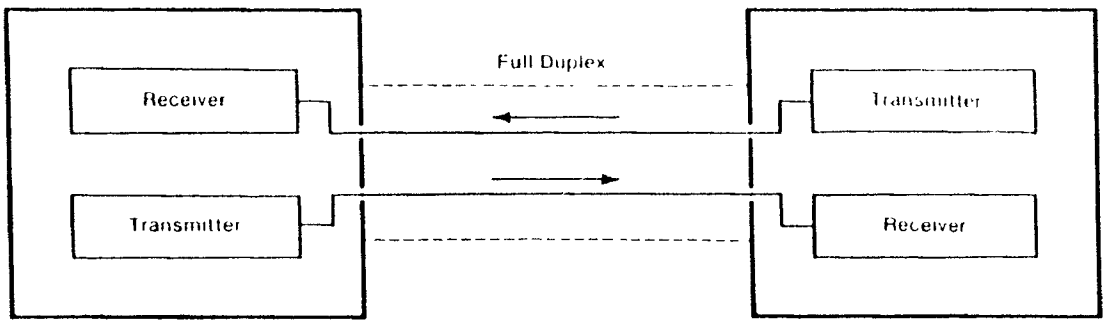


Figure 3.6: The Full Duplex Connection

with this approach is that when the telephone rings there is no way to know whether the caller is a Boy or Girl. Instead of using fixed definitions such as Boy and Girl, channel assignments are determined by the point of origin of the call. The modem placing the call is expected to use the originate channel and the modem receiving the call is expected to use the answer channel

3.1.4 Frequency Modulation

ASK modulation's greatest weakness is its susceptibility to noise because it uses amplitude variation to encode digital logic levels. Since no known natural phenomenom changes a signal's frequency, much more reliable encoding can be attained by frequency modulation. Since a modem need only transmit 1s and 0s, data can be represented by switching between two different frequencies. Mark/Space is defined to represent 1/0. Figure 3.7 depicts FSK modulation, but the frequency difference between the two tones is exaggerated for clarity. Figure 3.8, however, shows how the dual tones are actually assigned in the Bell Model 202, a half duplex 1200 bps modem using FSK modulation. This modem uses two tones 1000 Hz apart, centered about 1700

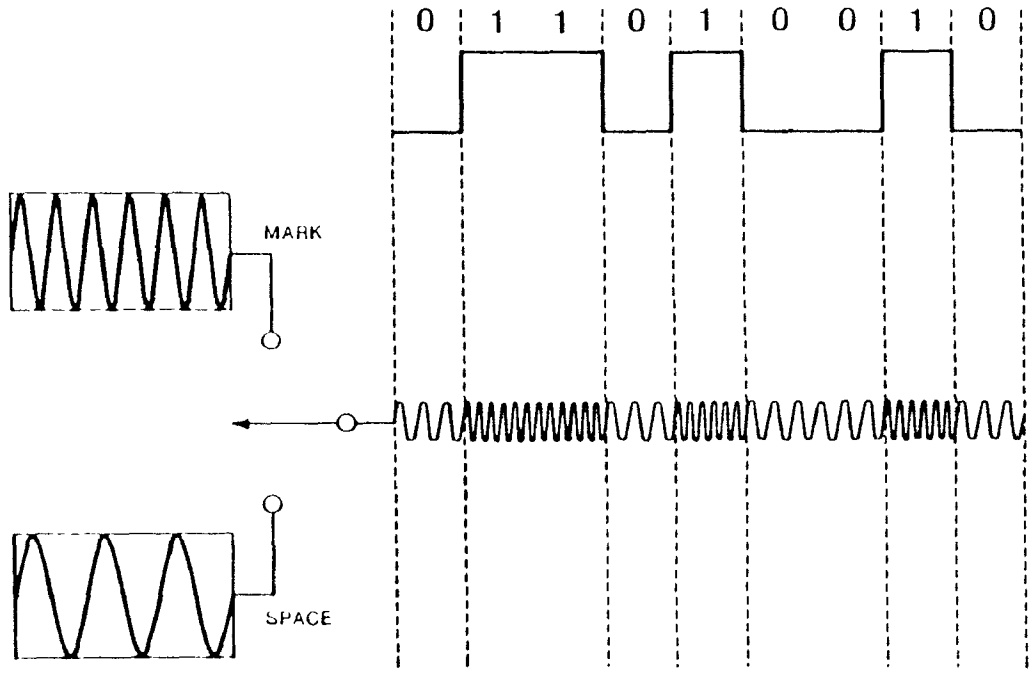


Figure 3.7: FSK Modulation

Hz. The 1200 Hz tone represents a MARK and the 2200 Hz tone a SPACE. Table 3.1 shows the FSK tone assignment for a Bell model 103, a full duplex, 110-300 bps FSK modem. The originate modem uses two tones, 100 Hz apart, centered about the frequency 1170; a SPACE is 1070 and a MARK is 1270. The answer modem tone also 100 Hz apart, but centered about 2125 Hz; here, a SPACE is 2025 and a MARK is 2225. Notice that the frequencies are chosen so that there is very little overlap between their spectra and consequently little interference between the two.

3.1.5 Phase Modulation

A periodic waveform has three properties : amplitude, frequency, and phase. We have already examined modulation techniques that encode digital data

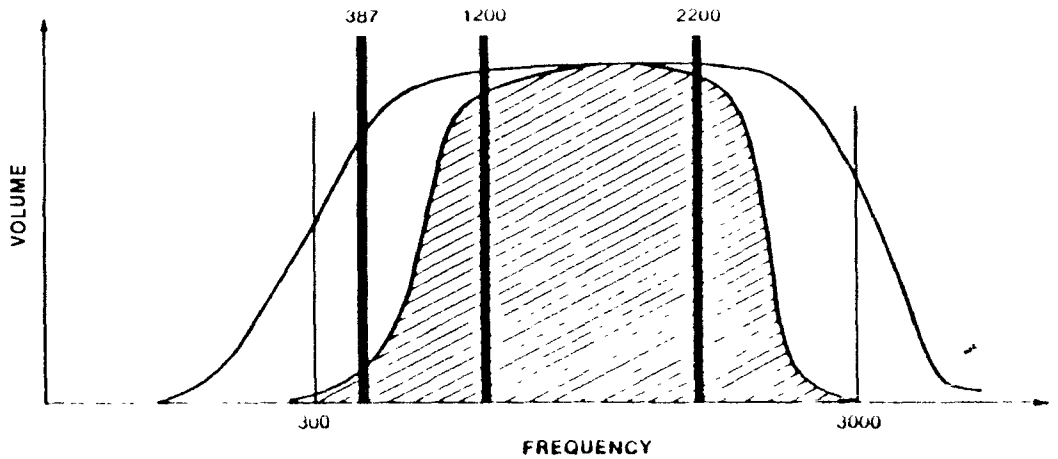


Figure 3.8. Tone Assignments for Bell 202 Half duplex, FSK 1200 bps Modem

	Originate		Answer	
	MARK	SPACE	MARK	SPACE
Bell	1270	1070	2225	2025
CCITT	980	1180	1650	1850

Table 3.1: Bell 103 vs CCITT V.21 300 bps, Full duplex Tone Assignment

into amplitude (ASK) and frequency (FSK). We will now briefly examine phase modulation, where information is encoded in the temporal relationship between two otherwise identical waveforms. In particular, we will examine the technique known as phase shift keying (PSK).

3.1.6 Two-level Phase Shift Keying

Figure 3.9 shows three waveforms, all exactly the same amplitude and frequency, but differing in phase. One complete cycle of any period waveform is expressed as 360^0. With respect to waveform A, waveform B is said to be in phase or, stated differently, its phase angle is 0^0. Waveform C, however, lags behind waveform A by 180^0 and its phase angle is -180^0. Modulation consists of sending waveform C for a SPACE, and waveform B for a MARK (We

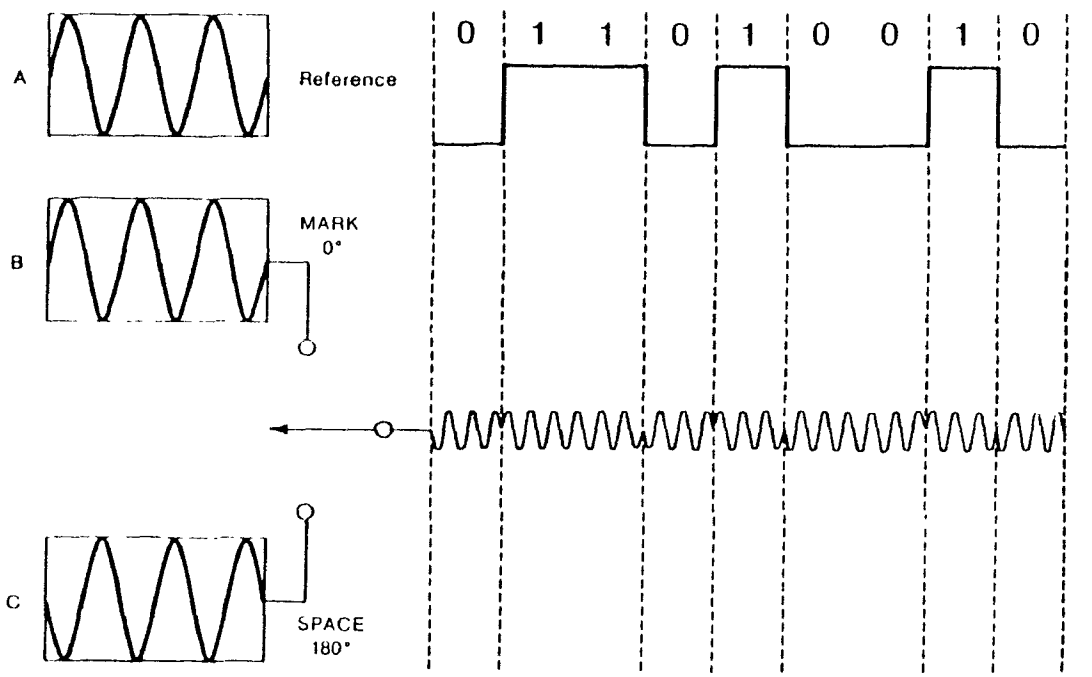


Figure 3.9: Two level Phase Shift Keying

assume that both sender and receiver already have local copies of waveform A which act as references. It need not be transmitted.)

3.1.7 Quadrature (Four-Phase) PSK

The encoding process known as *quadrature* phase shift keying (QPSK) is identical to the previous example except instead of encoding bits in two phase angles (0^0 and -180^0), four phase angles are used -0^0, 90^0, 180^0, 270^0. To understand how a modem attains a bit rate of, say, 1200 bps from a modulation rate of 600 baud, we have to understand that in asynchronous I/O, a byte is broken up into dibits . The new basic unit of data is no longer the bit, but the 2-bit pair, or dibit. Figure 3.10 shows how a dibit is encoded as

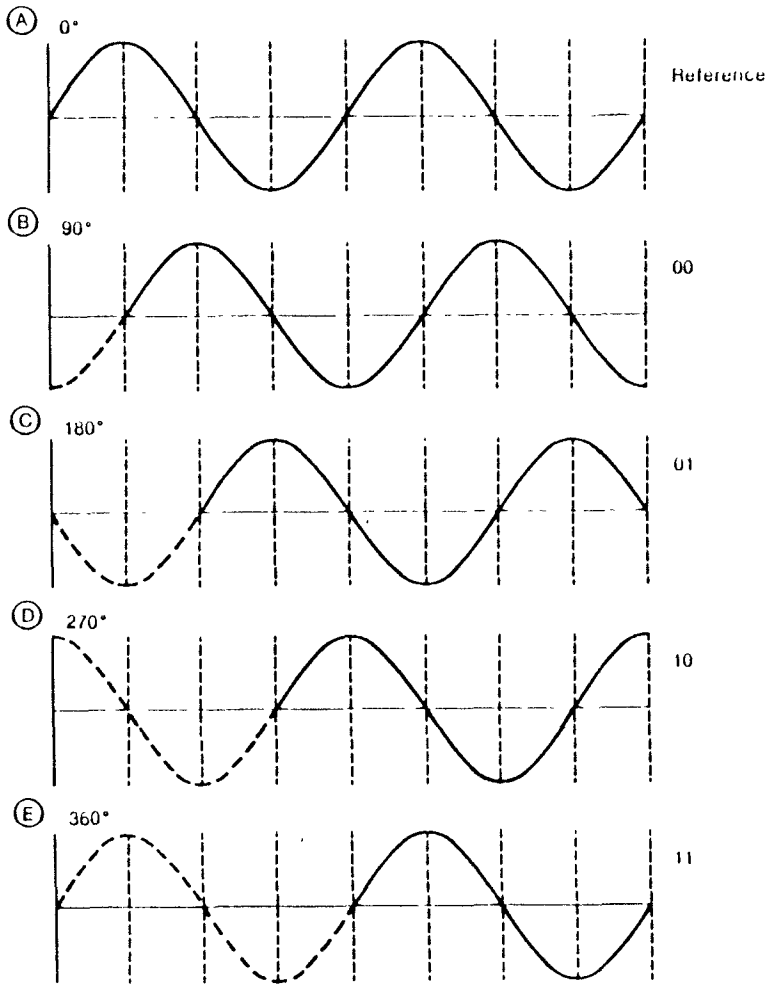


Figure 3.10: Quadrature Phase Shift Keying

one of four phase angles

3.1.8 Differential Phase Shift Keying

Our earlier examples assume that both sender and receiver have local copies of the reference waveform. If this is so, what keeps the sender's and receiver's reference waveform synchronized ? The solution, called *differential* phase shift key (DPSK), abandons altogether the idea of deriving phase differences from a quiescent reference signal (i.e. a sync pulse). Instead, the

Originate		Answer	
Transmit	Receive	Transmit	Receive
1200	2400	2400	1200

Table 3.2: Tone Assignment for the Bell 212A

phase angle for each cycle is calculated relative to the *previous* cycle. A modulation rate of 600 baud using four-level PSK provides a data rate of 2100 bps. This can be used in its entirety as a single 2100 bps half duplex channel as in the Bell 201 modem, or divided into two full duplex 1200 bps channels. The latter is the configuration of the popular Bell 212A modem whose frequency assignments are shown in Table 3.2.

3.1.9 Quadrature Amplitude Modulation

We have seen how data can be encoded into a signal's amplitude (ASK) and its phase (PSK). Aspects of both of these forms of modulation are combined to produce yet another modulation technique known as Quadrature Amplitude Modulation, or QAM. This technique results in a total data rate of 1800 bps with 600 baud modulation. QAM is specified by the CCITT V.22 *bis* full-duplex standard and has been adopted by the Hayes Smartmodem 2400 and others.

Just as the basic data unit for PSK is the dibit, QAM uses the *quabit*, or nibble. The CCITT version of QAM employs twelve phase angles in conjuction with three amplitudes. Of the resulting thirty-two unique phase/amplitude combination, only sixteen are used to encode a quadbit. The remaining six

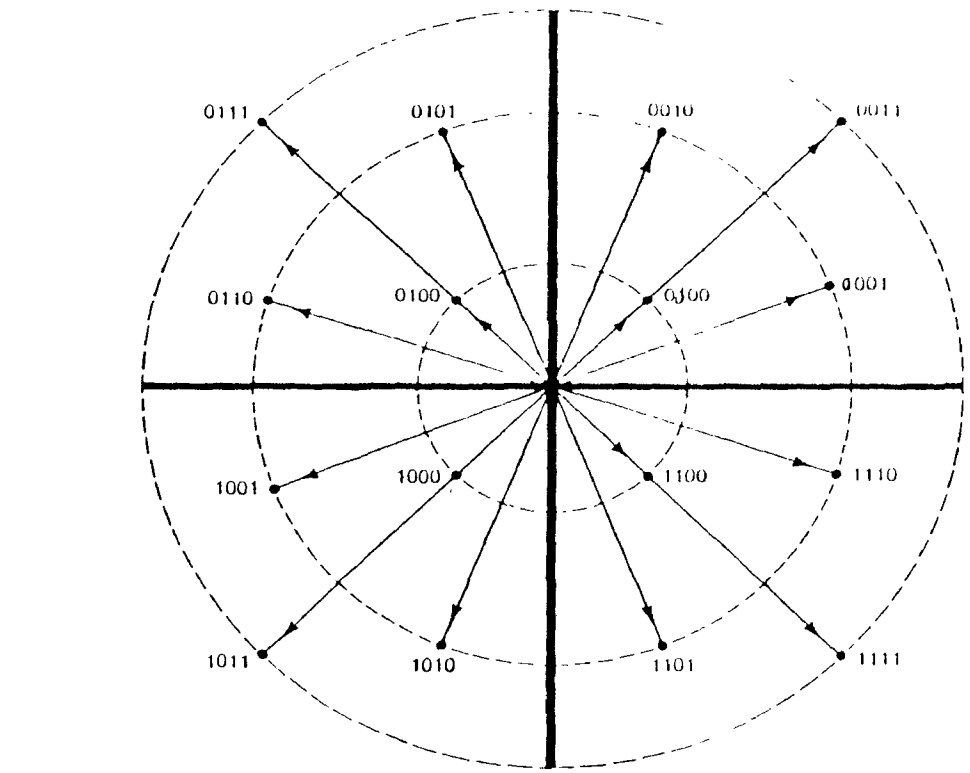


Figure 3.11: CCITT style QAM Modulation for 1800 bps at 600 Baud

teen combinations will be used in the future to attain a data rate of 9600 bps (1800 bps in full duplex). These are depicted in the phase "constellation" in Figure 3.11.

One of the chief benefits of frequency and phase modulation techniques such as FSK and PSK is their immunity to noise. By adding amplitude modulation to PSK, QAM sacrifices some noise immunity to achieve increased data rate. Indeed, in a noisy environment where a 1200 baud full duplex modem operates flawlessly, the 2400 full duplex QAM modem may be unusable

3.2 Introduction to Modem Chip

3.2.1 Description

In this design, The SSI 73K224L single-chip modem from Silicon Systems, Inc is used [1]. It is a highly integrated single-chip modem IC which provides the functions needed to construct a V.22bis compatible modem, capable of 2400 bit/s full-duplex operation over dial-up lines. The SSI 73K224L offers excellent performance and a high level of functional integration in a 28 pin DIP. This device supports all V.22bis, V.22, V.21, Bell 212A and Bell 103 modes of operation, allowing both synchronous and asynchronous communication

The chip is designed to appear to the system designer as a microprocessor peripheral. It will easily interface with popular single-chip microprocessors for control of modem functions through its 8-bit multiplexed address/data bus or via an optional serial command bus. An ALE control pin simplifies address demultiplexing. Data communications normally occur through a seperate serial port. The SSI 73K224L is pin and software compatible with the SSI 73K212L, which just supports the BELL 212A (1200 bps) and BELL 103 (300 bps) standards. It can operate from a single +5 volt supply for low power consumption. It is ideal for use in either free-standing or integral system modem products where full-duplex 2400 bit/s data communication over the 2-wire telephone network is desired. A complete modem requires only the addition of the phone line interface or data access arrangement (DAA), a control microprocessor, and RS-232 level converters. In this design, RS-232 level converters are not needed.

3.2.2 QAM Modulation/Demodulation

The SSI modem encodes incoming data into quadbits represented by 16 possible signal points with specific phase and amplitude levels. The baseband signal is then filtered to reduce intersymbol interference on the bandlimited telephone network. The demodulator, although more complex, essentially reverses this procedure while also recovering the data clock from the incoming signal. Adaptive equalization corrects for varying line conditions by automatically changing filter parameters to compensate for line characteristics.

3.2.3 Passband Filters and Equalizers

High and low band filters are included to shape the amplitude and phase response of the transmit and receive signals and provide compromise delay equalization and rejection of out-of-band signals. Amplitude and phase equalization are necessary to compensate for distortion of the transmission line and to reduce intersymbol interference in the bandlimited receive signal

3.2.4 Parallel Bus Interface

Seven 8-bit registers are provided for control, option select, and status monitoring. These registers are addressed with the AD0, AD1, and AD2 multiplexed address lines (latched by ALE) and appear to a control microprocessor as seven consecutive memory locations. Five control registers are read/write The status detect and ID register are read only and cannot be modified ex-

cept by modem response to monitored parameters.

3.3 Serial Input / Output

For most personal computers, serial input and output peripherals are used to reduce the number of connections to the outside world and are important and very popular I/O peripherals. SIO is a serial I/O that converts parallel data into serial data and vice versa. In this system an SIO port interfaces to a modem at the host PC and another, the SCI port of the 68HC11, interfaces to the modem chip at the image grabber.

3.4 Introduction to Hayes modem

Before micros, the Bell 212A or 103A modems were unchallenged standards, at least in America. But in the micro word and increasingly in the rest of computerdom as well, the Hayes Smartmodem product family has come to be the standard in the same fashion as the IBM line of micros has become the standard microcomputer. So dominant is the Hayes influence that market analysts estimate that upwards of 90 percent of all modems manufactured for the microcomputer market are compatible (or claim to be) with the Hayes Smartmodem.

Smartmodem States

Whenever the Smartmodem is engaged in a carrier link with another modem (called its online state), it behaves as a conventional modem, passing all RS-232 input directly to its transmitter. When not online, however, the Smartmodem is said to be in the command state, and RS-232 data is treated as potential commands. It is possible to switch the Smartmodem from the online to the command state and back again - without breaking the carrier link.

Startmodem Command State

In command state, the Smartmodem monitors the byes incoming from the RS-232 port in search of a particular sequence of bytes referred to as the Command Sequence Introducer, or CSI. After the CSI is encountered, the Smartmodem places subsequent character in its internal 40-character buffer, until the buffer becomes full or it encounters the Command terminator character. The Smartmodem's command terminator is nll, so by default, Carriage Return. The Smartmodem's CSI is AT, supposedly an abbreviation for "ATTENTION." AT also provides the modem with the transmission speed, character length, and parity set by your terminal or computer, and must appear at the start of every command line except for A/. AT also clear the last executed command line that stored in the command buffer. See chap4 for specific use of Hayes commands for this application.

Chapter 4

System Design

In this chapter, the system design related to modem will be described in detail. The system consists of two sides, the Motorola 68HC11-based controller side and the PC side. The controller side has the following capabilities: grabbing the images from the CCD, accepting the 9 channel analog input, controlling 4 digital switches, controlling SPI for high speed local communication, or controlling SCI and modem chip for long distance communication through public telephone system. The controller hardware block diagram is sketched in Figure 4.1. The PC side should have a VGA interface card and a VGA monitor to display the image in 64 gray levels. The high speed synchronizing card can interface SPI . The 2400 bps modem card can interface to the telephone system.

4.1 Hardware Design

The hardware design in the thesis is the modem interface on the controller side. Because the modem interface must work with the rest of the previously designed hardware module, it is designed to do so.

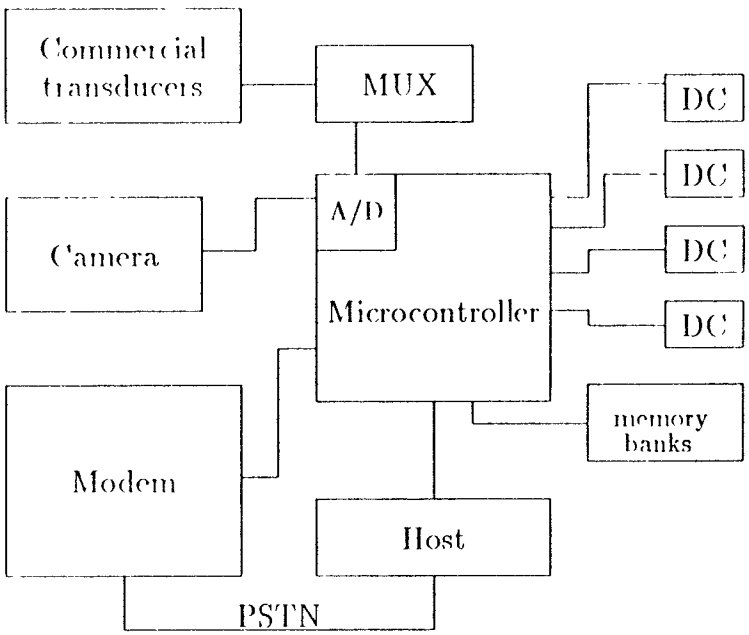


Figure 4.1: Block Diagram

4.1.1 Memory Decoder And Modem Decoder in 68HC11

Three banks of memory has been added to the EVB so that we can store an image in each. Since the original EVB did not support continuous memory with size more than 32K ($8000 [1]), the hardware and firmware were modified [3]. The bank memory starts from $2000 to $9FFF, totally 32 K, and the bank switch selector is at $B800 (not fully decoded). The new memory map is shown in Figure 4.2. In the thesis, modem addresses are from $2000 to $3FFF but in bank four. Every time the firware program wants to access a register of the modem, it enables bank four first. The following program segment is an example of selecting bank four memory.

[1] $ means this is hexadecimal

```
        :
        :
LDAA    #$04
STAA    $B800          ; enable bank 4.
        :
        :
```

The first bank of memory is normally used to store image data that is gotten from the A/D converter with no compression. The second bank is chosen to store compressed data. The data compression program must enable the respective banks before accessing them. The following program is an example of part of data compression program.

```
        :
        :
LDAA    #$00        ;enable bank 0.
STAA    $B800
        :           ;then fetch uncomoressed data.
        :           ;process it into compressed data.
        :
LDAA    #$01        ;enable bank 1.
STAA    $B800
        :           ;then store compressed data
        :
```

The decoder (74LS138) in EVB can't be used to be a modem decoder because timing of both sides do not match each other. We need to build a new modem decoder. The modem timing diagram is shown in Figure 4.3. The modem circuit diagram is shown in Appendix A. The 68HC11 expanding mode bus timing diagram is shown in Figure 4.4. The AS (address strobe) of 68HC11 is connected to ALE (address latch enable) of modem chip. When the chip is selected and AD0-AD7 carry the internal register address, the $\overline{CS}$ (chip select, active low) of SSi 73K224L should be active and the falling edge of ALE latches the address on AD0-AD2. The decoded $\overline{CS}$ should be active at first phase of every instruction. Connecting E clock to $\overline{G2B}$ (pin 5 of LS138) enables decoding at the first phase.

4.1.2 Modem Chip Hardware Interface with 68HC11 and DAA

This section describes general hardware connection. The 68HC11 pin description is shown in Figure 4.5. AD0-AD7 of the modem chip is an address/data bus whose bidirectional tri-state multiplexed lines carry information to and from the modem registers. AD0-AD7 of the modem chip are connected with AD0 AD7 of the 68HC11. Reset pin of the modem which is active high will put the chip into an inactive state. This RESET pin is connected with a inverted $\overline{RESET}$, an active low 68HC11 signal generated from external reset button. $\overline{RD}$ pin of the modem chip is active low signal which requestes a read of the SSI 73K224 internal registers. Because $\overline{RD}$ is active in second phase of each intruction, this signal is gererated by NAND of E clock and R/W of 68HC11. $\overline{WR}$ pin of the modem chip is an active low signal which

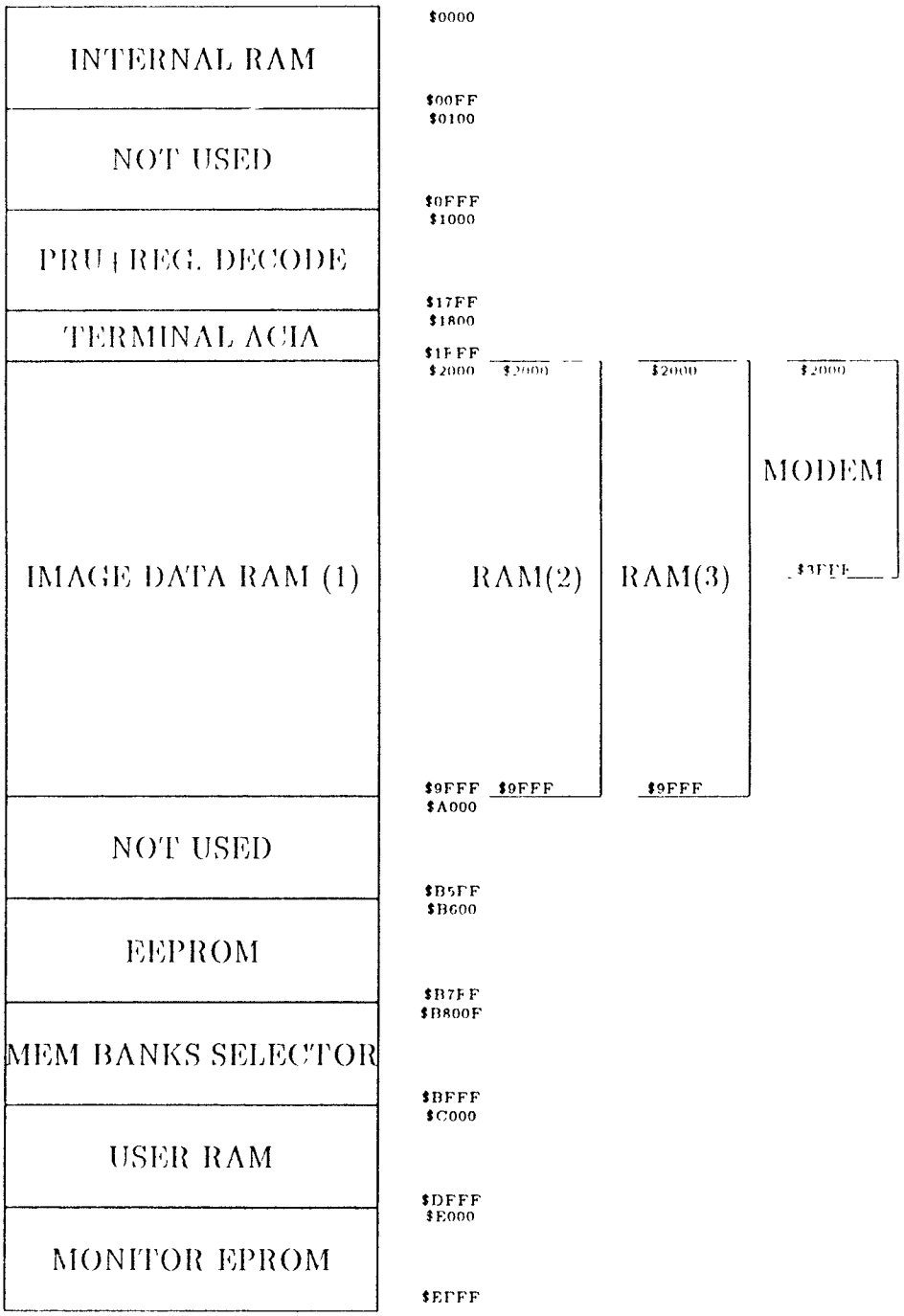


Figure 4.2: EVB New Memory Map Diagram

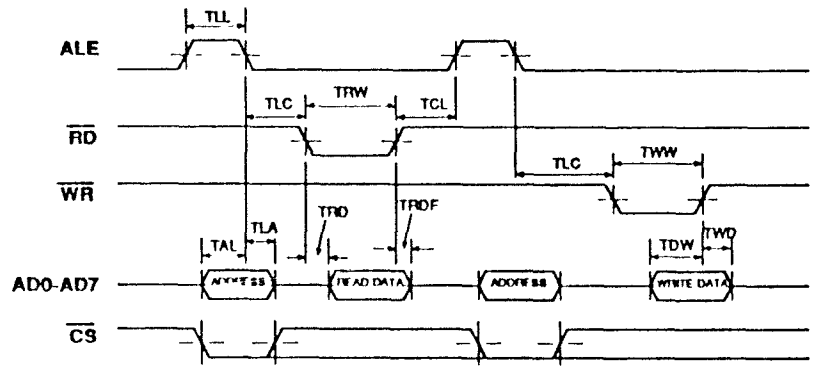


Figure 4.3: Modem Timing Diagram

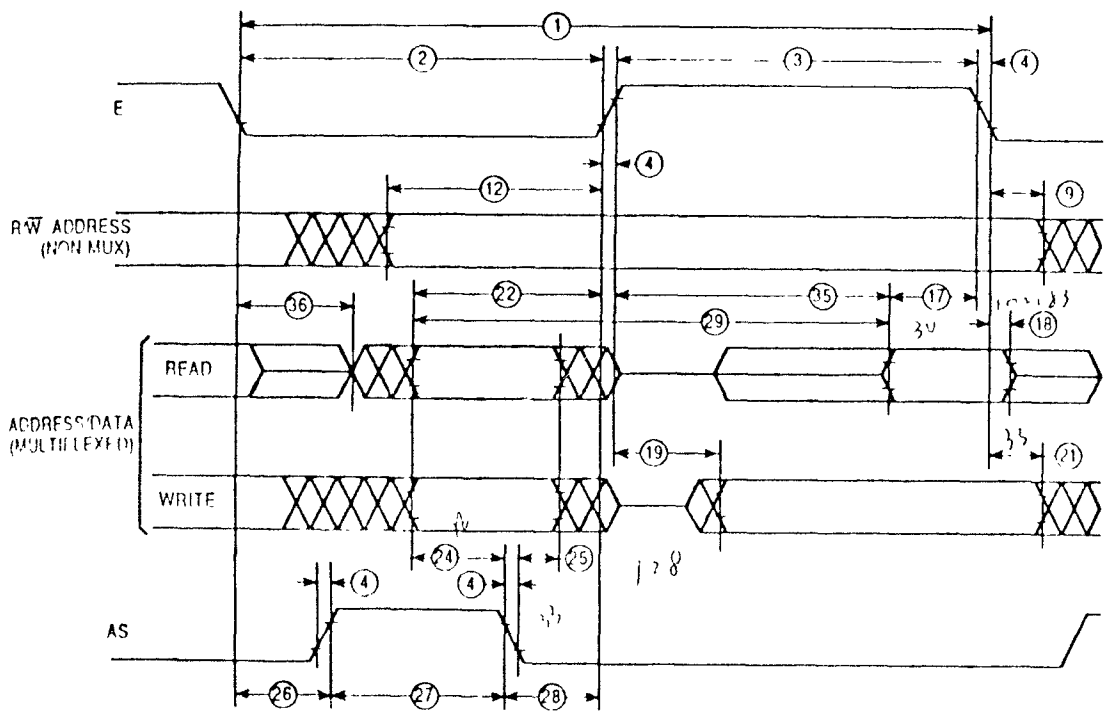


Figure 4.4: 68HC11 Expansion Bus Timing Diagram

informs the SSI73K224 that data is available on AD0-AD7 for writing into an internal register. This signal is also gated with E clock.

At this point we know how to interface the modem chip with the 68HC11 for command and status interchange. The chip modulates a serial digital signal into an analog signal or demodulates the analog into a digital signal The serial data appear on RXD and TXD. RXD is Received Digital Data and TXD is a Transmitted Digital Data. We just connect the respective RXD and TXD pins of the modem chip to RXD and TXD of the 68HC11.

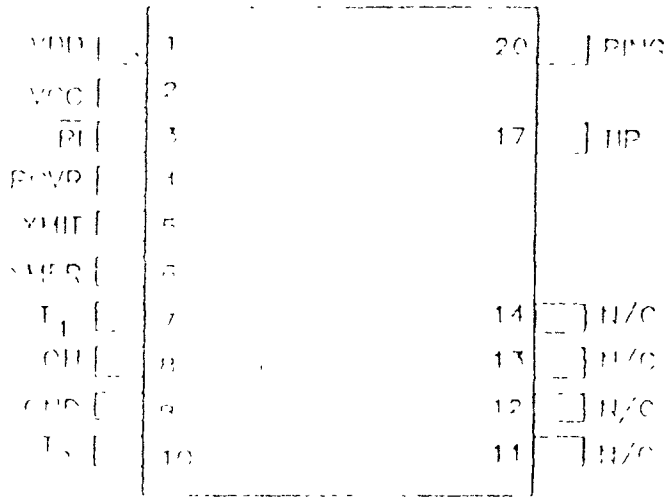
The DAA was described before as a communications component that provides a direct connect telephone line interface. The pin description is shown in Figure 4.6. It needs +/- 5 volt power supplies. LM7905 is a very good voltage regulator which input is -12 voltage and output is -5 voltage. The TIP and RING pin (pin 17 and 18) of DAA are telephone line inputs, which connect to the telephone line via a standard RJ11C jack. The $\overline{RI}$ (pin 3) of the DAA is a active low input ring detect, which indicates when ring voltage is present at TIP and RING. The $\overline{RI}$ (ring detect) is connected to PA2, shared with optical sensor 2 input. When we enable bank four, PA2 pin is connected to $\overline{RING}$ pin ,but when bank one is enabled, PA2 is connected to optical sensor 2 input. The OH (pin 8) of the DAA is off hook input to the DAA. The OH pin is connected to the PA3 pin (configured as ouput) of the 68HC11 which can be software controlled to high ("receiver" picked up) or low ("receiver" hung up).

NAME	28-PIN	44-PIN	TYPE	DESCRIPTION
GND	28	44	I	System Ground
VDD	15	23	I	Power supply input, 12V +10%, -20% (or 5V ±10%) Bypass with 1 and 22 µF capacitors to ground
Vref	26	42	O	An internally generated reference voltage Bypass with 1 µF capacitor to Ground
Iset	24	36	I	Chip current reference Sets bias current for op amps The chip current is set by connecting this pin to VDD through a 2 MΩ resistor Iset should be bypassed to GND with a 1 µF capacitor

ALE	12	20	I	Address latch enable The falling edge of ALE latches the address on AD0 AD2 and the chip select on $\overline{CS}$
AD0-AD7	4-11	4,9-15	I/O	Address/data bus These bidirectional tri-state multiplexed lines carry information to and from the internal registers
$\overline{CS}$	20	32	I	Chip select A low on this pin allows a read cycle or a write cycle to occur AD0-AD7 will not be driven and no registers will be written if $\overline{CS}$ (latched) is not active $\overline{CS}$ is latched on the falling edge of ALE
CLK	1	1	O	Output clock This pin is selectable under processor control to be either the crystal frequency (for use as a processor clock) or 16 x the data rate for use as a baud rate clock in QAM/DPSK modes only The pin defaults to the crystal frequency on reset
$\overline{INT}$	17	25	O	Interrupt This open drain weak pullup, output signal is used to inform the processor that a detect flag has occurred The processor must then read the detect register to determine which detect triggered the interrupt $\overline{INT}$ will stay active until the processor reads the detect register or does a full reset
$\overline{RD}$	14	22	I	Read A "low" requests a read of the SSI 73K224 internal registers Data cannot be output unless both $\overline{RD}$ and the latched $\overline{CS}$ are active or "low"
RESET	25	37	I	Reset An active signal "high" on this pin will put the chip into an inactive state All control register bits (CR0, CR1, CR2, CR3, Tone) will be reset The output of the CLK pin will be set to the crystal frequency An internal pull down resistor permits power on reset using a capacitor to VDD

NAME	28-PIN	44-PIN	TYPE	DESCRIPTION
$\overline{WR}$	13	21	I	Write A "low" on this informs the SSI 73K224 that data is available on AD0-AD7 for writing into an internal register Data is latched on the rising edge of $\overline{WR}$ No data is written unless both $\overline{WR}$ and the latched $\overline{CS}$ are active ("low")
Note	The serial control mode is provided in the 28 pin version by tying ALE high and $\overline{CS}$ low In this configuration AD7 becomes DATA and AD0, AD1 and AD2 become A0, A1 and A2, respectively			

Figure 1.5. SSI73K221 Pin Description



PIN	SYMBOL	TYPE	DESCRIPTION
1	V_{EE}	-	Negative Supply, -5.0 volts
2	V_{CC}	-	Positive Supply, +5.0 volts.
3	RI	O	Ring Detect. Active low output; indicates when ring voltage is present at TIP and RING.
4	RCVR	O	Receive. Extracted receive signal
5	XMIT	I	Transmit. Transmit signal input.
6	XMFR	O	Transformer. Connect to T1 when using internal 2 to 4 wire converter; leave open when using external 2 to 4 wire converter.
7	T1	I	Tap 1. Connect to XMFR for internal 2 to 4 wire converter; to XMIT for external 2 to 4 converter.
8	OH	I	Off Hook. Controls telephone line hook status; when low line is "on hook".
9	GND	-	Signal Ground. 0.0 volts.
10	T2	-	Tap 2. Tie to GND for proper operation.
11,12 13,14	N/C	-	No Connect. Leave these pins open for proper operation.
17,18	TIP/ RING	I	Telephone Line Inputs. Connect to telephone line via standard RJ11C jack

Figure 1.6: DAA Pin Description

4.2 Software Design

As the system design is divided into two parts, a controller side and a PC side, so is the firmware design. The design of the firmware described here is modem related. The modem and data compression portion will be in detail but the whole system will be generally described.

4.2.1 Controller Firmware Design

The whole idea to add modem into the local transmission system and replace it is very easy knowing of modem connecting sequence, data compression firmware and whole system.

Initialization of Serial Communication Interface

The modem modulates serial data into analog form and demodulates the analog signal into serial form. Before description of modem, we will discuss serial port first. The three SCI control registers in the 68HC11 are initialized and its TXD pin is routed on the EVB. In this design, 2400 baud, 8 bits data, no parity and one stop bit asynchronizing transmission is the standard format for both side. To get 2400 baud, write #%00110010[2] binary data to the baud rate register ($102B). Store #0 to SCCR1 (SCI Control Register 1 , $102C) to select one start bit, eight data bits, one stop bit and idle line. Write #%00001100 to SCCR2 (SCI Control Register 2) to select TDRE interrupts disabled (software polling), TC interrupts disabled (software polling

[2]# represent this is immediate data and % means binary data for Motorola serial

mode), RDRF and OR interrupts disabled (software polling mode), IDLE interrupts disabled (software polling mode), SCI transmitter enabled, SCI receiver enabled, normal SCI receiver operation, and normal transmitter operation. In 68HC11 evaluation board hardware design, write #0 to $1000 to select PD0 on P1 connector to PD0 (TXD) of 68HC11 chip, not from host computer I/O port connector. The following program segment is an example of initialization of SCI.

Initialize the SCI .....

```
        :
LDAA    #%00110010      ;initialize 2400 baud
STAA    BAUD
LDAA    # 0             ;
STAA    SCCR1           ;sci mode
LDAA    #%00001100      ;
STAA    SCCR2           ;interrupt control
LDAA    # 0             ;enable flip flop
STAA    $4000           ;route TXD
        :
```

Polling of the Serial Communication Interface

Because a polling system is easier for firmware design and the whole system is insensitive to real time response, polling is chosen for both transmitter and receiver in the 68HC11 communication design. Every kind of interrupt caused

by the serial communication is disabled, and the status register is polled every time the system wants to transmit or receive data. There are two registers used in the communication polling firmare. The SCI status register is used to check if the transmit buffer is empty or the receiver buffer has a datum so serial buffer then can be loaded or stored a dataum. for more detailed information, please refer to appendix A. The following program segment is an example of SCI polling :

```
            :
TX    LDAA  SCSR          ;load status register.
      BITA  #%10000000    ;check Transmit Data Register Empty bit.
      BEQ   TX            ;if it is not empty then poll again.
      STAB  SCDR          ;store data to serial data register.
            :
RX    LDAA  SCSR          ;load status register.
      BITA  #%00100000    ;check Receiver Data register full bit.
      BEQ   RX            ;if it is not full then poll again.
      LDAA  SCSR          ;load data from serial data register.
            :
```

The Ring Input Pulse Width Measuring

The ringing input is connected to PA2 of 68HC11. The inputs PA0-PA2 of the 68HC11 have an input-capture function which can be used to measure pulse width. Each input-capture function includes a 16-bit latch, input edge-detection, and interrupt generation logic. The 16-bit latch captures the

current value of the free running counter when a selected edge is detected at the corresponding timer input pin. The edge-detection logic includes control bits so that user software can select the edge polarity that will be recognized. Each of the three input-capture functions can be independently configured to detect rising edges only, falling edges only, or any edge (rising or falling). The interrupt generation logic includes a status flag, which indicates that an edge has been detected, and a local interrupt enable bit, which determines whether or not the corresponding input-capture function will generate a hardware interrupt request.

The central element of each input-capture function is the input-capture latch, which can be read by software as a pair of 8-bit registers. When an edge has been detected and synchronized, the 16-bit free-running counter value is transferred into the input-capture register pair as a single 16-bit parallel transfer.

To measure pulse width, the input-capture edge sensitivity must be reconfigured between the capture of the first edge and the second edge. Since this ringing detect program measures the period of a low-going pulse, the input capture is first configured to capture on a falling edge at PA2. After detecting the first edge, the input capture is reconfigured to detect a rising edge.

Since this pulse width measure program is interrupt driven, it must have an interrupt service routine, which is automatically called as a result of an

interrupt, an initialization portion, and a mainline program portion. The following program segment is an example of initialization of TOF and IC1 service routine. TOF is Timer Overflow service routine and IC1 is Input Capture 1 service routine.

Initialization of TOF and IC1 service routine jump table...

```
        :
LDAA    #$7E        ;Jump (extended) Opcode.
STAA    PVTOF       ;Pseudo vector.
STAA    PVIC1       ;IC1 pseudo vector.
LDX     #SV3TOF     ;address of TOF service routine.
STX     PVTOF+1     ;finish jump instruction to TOF routine.
LDX     #SV3IC1     ;address of IC1 service routine.
STX     PVIC1+1     ;finish jump instruction to IC1 routine.
        :
```

SV3TOF-timer Overflow service routine ...

```
SV3TOF  TST   IC1MOD          ;if 0 or 1 IC1 active_count TOFs
        BMI   OUT3TOF         ;if neg, IC1 not active
        INC   OVCNT1          ;increment IC1 overflow count
OU3TOF  LDAA  #$80            ;
        STAA  REGBAS+TFLG2    ;clear overflow flag
        RTI   ;return from TOF service
              :
```

SV2IC1-Input Capture 1 service routine ...

```
SV3IC1   LDX    #REGBAS        ;point to top of register block
         INC    IC1MOD         ;$FF → at 1st edge
                               ;0 → at 2nd edge
         BNE    NO1ST3         ;if not 0,this is not second edge
         CLR    OVCNT1         ;Zero the overflow count
         BCLR   TCTL2,X $30    ;EDG1B:EDG1A → 0:0
         BSET   TCTL2,X $10    ;EDG1B:EDG1A → 0:1 rising
         LDD    TIC1,X         ;Read time of first edge
         STD    RES1           ;Save till next capture
         BMI    OU3IC1         ;Done if IC was before any TOF
         LDAA   TFLG2,X        ;Check for TOF IN MSB
         BPL    OU3IC1         ;If no overflow ,you're done
         DEC    OVCNT1         ;This TOF shouldn't count
         BRA    OU3IC1         ;Done processing first edge
NO1ST3   LDD    TIC1,X         ;Get time of second edge
         BMI    ARNOV1         ;If MSB=1, skip TOF check
         TST    TFLG2,X        ;Check for overflow
         BPL    ARNOV1         ;If no TOF, skip increment
         INC    OVCNT1         ;TOF was before edge so count it
ARNOV1   SUBD   RES1           ;Time of last minus time of 1st
         STD    RES1           ;Update result
         BCC    RES1OK         ;Check for borrow
         DEC    OVCNT1         ;If borrow, fix overflow count
```

```
RES10K   BCLR   TCTL2,X $30    ;Disable IC1
         BCLR   TMSK1,X $04    ;disable IC1
         BCLR   TMSK2,X $80    ;disable TOF
         LDAA   #1
         STAA   IC1DUN         ;Signal period measured
OU3IC1   BCLR   TFLG1,X $FB    ;Clear IC1F
         RTI                   ;return from IC1 service
```

The Connect Handshake for CCITT V.22bis Modems

The modem chip is programed for answering mode only. When we enter Figure 1.7 at the left, the calling modem has placed a call, heard ringback and (if it was listening) and is waiting to hear unscrambled binary ones from the answering modem. The answering modem went off hook when it detected ringing. The answering modem must be silent for two seconds after going off hook for billing protection. There are many places in the firmware needing time delay. The following program segment is an example of delaying two seconds.

Time Delay 2 sec routine ...

```
         LDAA   #$7E       ;Jump (extended) Opcode
         STAA   PVOC2      ;Pseudo vector see manual text
         LDX    #SV5OC2    ;address of OC2 service routine
         STX    PVOC2+1    ;finish jump instruc to TOF svc
         LDAA   #64        ; value for delay 2 sec
         STAA   COUNT
```

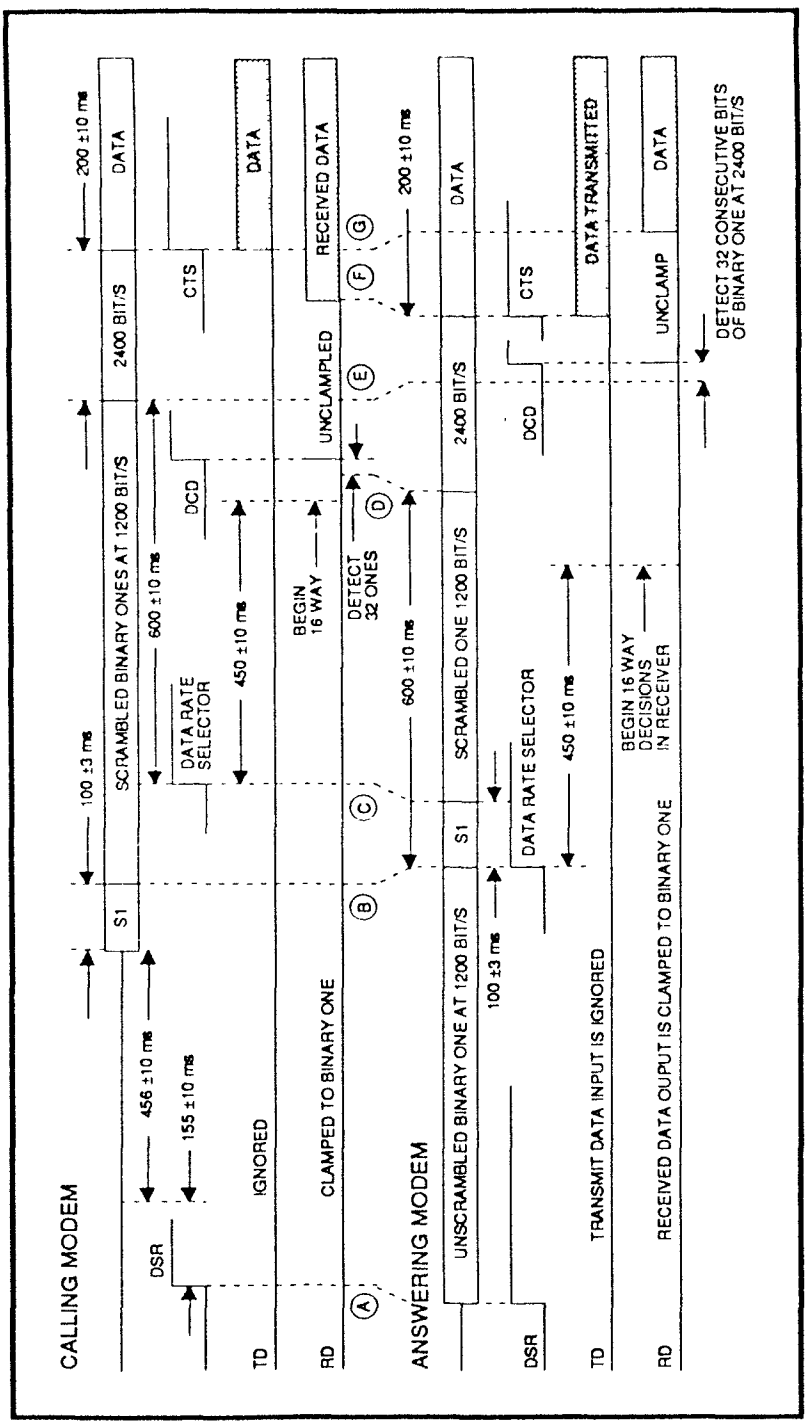


Figure 1.7 CCITT V.22 Connect Handshake Sequence

```
        LDD   #1
        STD   DELAY
        LDX   #REGBAS
        LDAA  #0
        STAA  TCTL1,X   ;no change when compare ocurrs
        LDD   TCNT,X    ;load current counter value
        STD   TOC2,X    ;store to compare register
        LDAA  #$40
        STAA  TFLG1,X   ;clear any pending OC2F
        STAA  TMSK1,X   ;enable OC2 interrupts
        CLI
WAIT1M  LDAA  COUNT     ; wait here for 2 sec
        BNE   WAIT1M
```

SV5OC2-Output Compare 2 service routine ...

```
SV5OC2  LDD   DELAY     ;get delay time for 1/2 cycle
        ADDD  RTOC2     ;add to last compare value
        STD   RTOC2     ;update OC2
        LDAA  #$40      ;clear OC2f
        STAA  RTFLG1
        DEC   COUNT
        BNE   EXIT
        CLR   RTMSK1
EXIT    RTI             ;return from OC2 service
```

This portion is main modem firmware program. For more information about the detail please refer to K-series Modem Design Manual [5].

4.2.2 PC Side Software and Firmware Design

The image controller with data logger program running for modem is called IMAGEX. The operation and function of the IMAGEX is similar to IMAGE described by Mr. Qi [3]. The IMAGE program can run only for the local high speed add-on card designed by Mr. Feng [2]. To run with the modem, the only thing needed to do is to modify the driver. Because data compression is used on the controller side, the PC side includes a decompression program in IMAGEX.

Assembly language interface with C language

To interface assembly language subroutines with C, you must follow a few general guidelines. First, you must give a specific segment name to the code segment of your assembly language subroutine. For example, Microsoft C requires the segment name _TEXT.

Second, your C compiler may require specific names for data segments (if the data is being referenced outside the code segment). Microsoft C require that the segment be named _DATA.

Third, you must realize how variables are passed to assembly language subroutines through the stack. In the function-calling syntax of

```
function_name(arg1,arg2,arg3,...,argn);
```

the values of each argument are pushed on the stack in reverse order. Thus, argument n (`argn`) is pushed on the stack first, and argument 1 (`arg1`) is pushed last. An actual value, or a pointer to a variable, can be passed on the stack although most values and pointer are passed as word-length stack elements, the longer data elements such as long or unsigned long-type variables require 32 bits (2 words) of stack space. If the memory model being used is compact, large, or huge, or if the data item has a segment override, data pointers also require 32 bits of stack space.

Fourth, in the assembly language source file, the assembly language routines to be called from Microsoft C must begin with an underline. However, the underline is not included when the routines are invoked in C.

Fifth, remember to save any special-purpose registers (such as CS, DS, SS, BP, SI, and DI) that your assembly language subroutine may disturb. Failure to save them may have undesired consequences when control is returned to the C program. You do not have to save the contents of AX, BX, CX, or DX because these registers are considered to be volatile by C; that is, the language assumes that you will change them and even requires you to do so in order to return a value.

The following program segment is an example of assembly subroutine interfacing with Microsoft C.

assembly routine interfacing with C ...

```
_siofil  proc
         push   bp
         mov    bp,sp
         push   ds
         push   es
         push   di
         push   si
         mov    ax,[bp+6]   ;bx:buffer
         mov    bx,ax
         mov    ax,[bp+8]   ;cx:165*192
         mov    cx,ax
         :                  ;driver routine here.
         :
         pop    si
         pop    di
         pop    es
         pop    ds
         pop    bp
         ret
```

Initialization of 8250 UART

The 8250 UART is a serial interafce chip which convert parallel data into asynchronous serial data with programmable stop bit, parity bit, data bit, and one fix start bit. The basic asynchronous serial I/O functions and many others are built into the Universal Asynchronous Receiver/Transmitter, or UART. The idea behind a UART is to relieve programmer and processor of the toil associated with asynchronous serial I/O. To receiver and send data, the program simply reads and writes bytes to UART, which appears to the processor as one or more ordinary memory locations or I/O ports. The UART supplies the following four kinds of I/O ports to program: Input Status Port, Output Control Port, Output Data Port and Input Data port. The following program segment is to initialize the 8250 as 2400 baud, 8 bits, one stop bit, no parity bits, and interuupt enable state. For more information about each register, please reference to "C Programmer's Guide to Serial Communications" [6].

INITIALIZE THE 8250 UART ...

```
add     dx,3        ;ADDRESS OF LINE CONTROL REGISTER
mov     al,80h
out     dx,al       ;TO ADDRESS BAUD RATE DIVISOR REGISTERS
mov     dx,bx       ;ADDRESS OF BAUD RATE DIVISOR LSB
mov     al,30h      ;LSB VALUE FOR 2400 BAUD
out     dx,al
mov     dx,bx       ;ADDRESS OF BAUD RATE DIVISOR MSB
add     dx,1
```

```
mov     al,0        ;MSB VALUE FOR 2400 BAUD
out     dx,al
                    ;THE BAUD RATE HAS NOW BEEN INITIALIZED
                    ;NOW INITIALIZE THE LINE CONTROL REGISTER
mov     dx,bx       ;ADDRESS OF LINE CONTROL REGISTER
add     dx,3
mov     al,03h      ;no PARITY, 1 STOP BIT, 8 DATA BITS
out     dx,al
                    ;NOW INITIALIZE THE MODEM CONTROL REGISTER
                    ;FOR REQUEST TO SENT AND DATA TERMINAL READY
mov     dx,bx       ;ADDRESS OF MODEM CONTROL REGISTER
add     dx,4
mov     al,03h      ;SET MODEM CONTROL SIGNALS
out     dx,al
mov     dx,bx       ;ADDRESS OF INTERRUPT ENABLE REGISTER
add     dx,1
mov     al,0
out     dx,al
```

3F8H-3FFH are I/O addresses of COM1. In the thesis design, we choose COM1 as the 8250 address and reserved the flexibility to change it to any address. It is very easy to change the 8250 I/O address in software and hardware. In software design, simply modify 3F8H to 2F8H (COM2), 3E8H (COM3), or 2E8H (COM4). In hardware, adjust the modem card dip switch or serial card jumper to the selected I/O port. The following

program segment is part of C showing easy change to different I/O address and the modem driver.

modem driver and I/O address ...

```
#define SIO1 0x3f8


int extern mexpo1( int sio, int time );
int extern inisio1( int sio );
int extern hayes( int sio, char *dial, char responce[] );
int extern siofil( unsigned char *a, int count );
int extern digital( int sio, int switches );
int extern analog( int sio, unsigned char *status );
```

4.2.3 Startmodem Command used in the thesis

ATDn is the only command used in the thesis. The "D" command causes a telephone number to be dialed. N represents an ASCII string composed of dial digits and dial modifiers. For pulse dialing, the dial digits include the decimal value 0 through 9 and for Touch-Tone dialing, the digits 0 through 9 plus the symbols A, B, C, D, #, and *. The D command also can combine with other dial modifiers such as T, P, R,; , ... which will be discussed later.
Example: ATD 5965814
After the command ATD 5965814 is entered, your modem will auto-dial the telephone number 5965814. The P command causes your modem to pulse dial the numbers that follow. The T command causes your modem to touch-tone dial the numbers that follow. You can use the P command in combination

with the T command when both pulse touch-tone dialing are required. The following program is a part of dialing-out driver written in 8088 assembly language.

Hayes Command driver ...

```
          mov     ax,[bp+6] ;bx:sio address
          mov     bx,ax
          mov     si,[bp+8] ;si:hayes command string adress
          mov     di,[bp+10];di:modem response string adress
          cli
start:    mov     dx,bx
          add     dx,5
ready?:   in      al,dx
          test    al,20h
          jz      ready?
          mov     dx,bx
          mov     al,[si]
          or      al,al
          jz      null
          out     dx,al
          inc     si
          jmp     start
null:     mov     si,di
rec:      add     dx,5
ready2:   in      al,dx
```

```
            test    al,01
            jz      ready2
            mov     dx,bx
            in      al,dx
            mov     [si],al
            inc     si
            cmp     al,0dh
            jz      termı
            cmp     al,0ah
            jnz     rec
termi:      mov     al,0
            mov     [sı],al
```

Chapter 5

System Analysis of Limits and Performance

5.1 Image transmission

The 2400 baud modem is a popular, cheap, high speed modem in today's market. The whole image has 165 × 192 pixels and each pixel is represented by one byte so a whole image contains 165 × 192 = 31680 bytes. Each pixel has 256 gray levels. A local high speed SPI transmsmits the whole image in about 0.25 sec.[2]. For a 2400 baud modem, it would take:

$$\frac{10 \times 165 \times 192}{2400} = 132sec.$$

= 2 min 12 sec to send a full frame image. 10 bits/pixel include start and stop bits. Modem transmission time is the bottleneck of the system performance when the system is transmitting the image. For 9 channel analog input, it takes

$$\frac{10 \times 9}{2400} = 0.04sec$$

to send 9 channel analog data. For 4 bit output switch, it takes

$$\frac{10 \times 1}{2400} = 0.001sec.$$

For 9 channel analog input and four digital switch ouput response, it takes the same time as local high speed SPI. The image grabbing time and polling time usually take about 4 sec regardless of transmission system.

5.2 Data compression

If we could save half of modem transmission time, we could save about 1 min. The idea of data compression is to compress the whole image data into small amount of data so that it saves the transmission time and therefore saves the whole system time. If we look into the VGA display system carefully, the processed image data can look exactly the same as the original image data. See Figure 5.2 and Figure 5.1. The display system is VGA card and monitor and the diplaying image is in VGA 256 colors mode. The monochrome image in 256 colors mode can show 64 gray levels because the RGB pallete register in RAMDAC (programmable digital to analog converter) is 6 bits. So, the received data image can be divided by 4 to get the 64 scale values. Some quality of image gray level is lost. This can be perceived if images produced by high quality printer are compared.

5.2.1 Introduction to Data Compression Algorithm

If you look into the image data carefully, you will find most of the image data amplitude change by small special amounts in adjacent pixels. Where possible, we just transmit the difference between adjacent gray levels instead of the whole amplitude, a form of delta modulation. The data compression

Figure 5.1: A picture transmitted without data compression

Figure 5.2: A picture transmitted with data compression

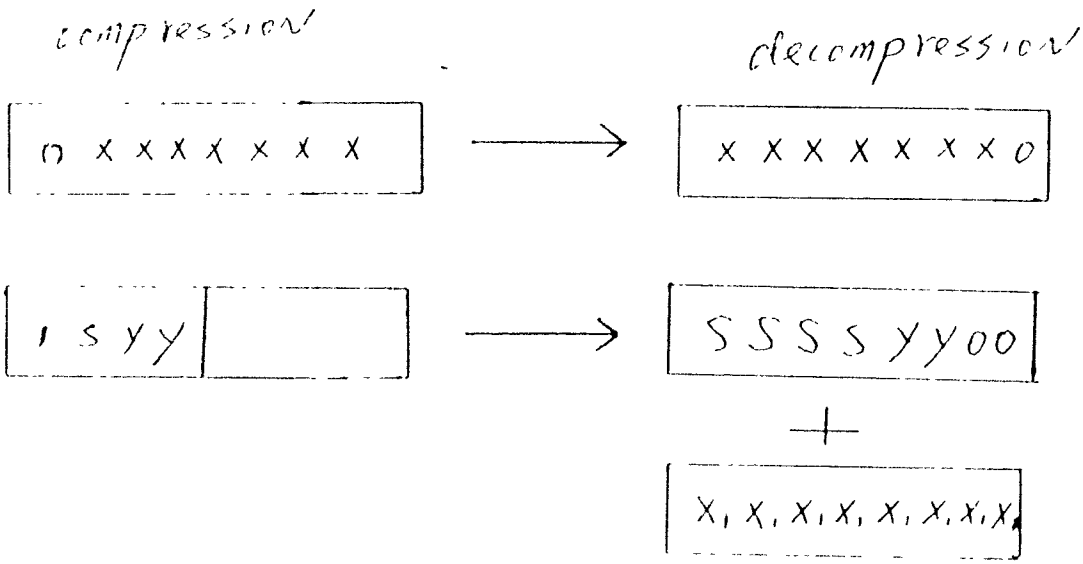


Figure 5.3: data compression formats

algorithm is as follows: If the difference of two adjacent pixel is within the scope of compression, then send the difference. If the difference is out of the scope, then send the original amplitude. The two format are shown in Fig 5.3. The first bit is compression ID bit which can be located in any byte or nibble boundary. If this bit is zero, then following 7 bits is the original amplitude divided by two. If this bit is one, then the following 3 bits is the compressed data. The second bit is a sign bit and the last two bits are the difference that is obtained by subtracting the preceeding pixel amplitude from the current amplitude and dividing the difference by four. The data compression on controller side flow chart and data decompression on host side flow chart are in Appendix B.

Chapter 6

Conclusions

6.1 Conclusion

The modem part of the image control system supplies extra-long distance communication capability. With the help of software data compression, we can save almost half of the image transmission time, and the compressed image looks the same as the uncompressed image on a VGA display.

Figure 5.2 and Figure 5.1 show two pictures taken under the same conditions. The transmission time without data compression is 2 min 12 sec but the transmission time with data compression is about 1 min 12 sec depending on the nature of the image. The less busy the image, the more time saved.

This modem-based intelligent camera design is very good for applications where frequent images are not required and distances to a host computer are too great for economical high speed communication. Example applications would range from security to process control to animal management. These applications share a need for cost-effective vision systems at modest image data rates

6.2 Suggestions for future work

Since the modem should use a serial port in the PC, it would be flexible for the PC side software to be able to configure the modem port to any one of four serial ports under user control. COM1 is selected as modem port in IMAGEX. To change to a differenet port, we must modify COM1 3F8 to say COM2 2F8 in IMAGEX C source program and recompile it. That is not a friendly way for a user to change the host's modem port.

To make IMAGEX more flexible, the dial out phone number should be changeable in the program's setup menu by the user so that the program can dial out to any controller phone number to establish the data link between host and controller. In this IMAGEX C program, "ATDT5814" is the dial string. One could change the string to another like "ATDT9914324" to cause the program to dial my home.

For some application, this system can be controlled by both a local and a remote host at the same time. The modem hardware was well-designed without usurping other hardware. In such a case, either the local or the remote host can know the image and control the same system. This can be achieved by modifying the controller firmware without any change in its hardware. At the polling loop check SPI and SCI receive buffer once to see who wants to communicate first. Then execute either the SPI or SCI subprogram according to which host requests service.

A PC can access four serial cards with different I/O address and interrupt pins respectively. The host system can be designed to talk to four different controllers at different times. Because each serial card has a different interrupt pin, the program can be modified as an interrupt driven program. Under interrupt driven enviroment, the user can switch from one controller to another during transmission.

The exposure time setting [2] [3] is supposed to be changeable by user But the controller firmware does not take care of it well. Correct the controller firware to meet the user request correctly.

Using bank switching, memory space or I/O space can be extended. Addresses in bank 4 can be used to add I/O control such as auto focus, auto brightness, and camera head position.

Since DTMF generator is on this modem chip, modem chip can dial out to anywhere the remote host is located. The remote host and controller need not be always connected together. In certain specific situations, the controller could call the remote host and establish a link, therefore saving telephone fees if this is not a local call.

The above suggestions are not in conflict and could be implemented in any combination.

Appendix A

Circuit Diagrams

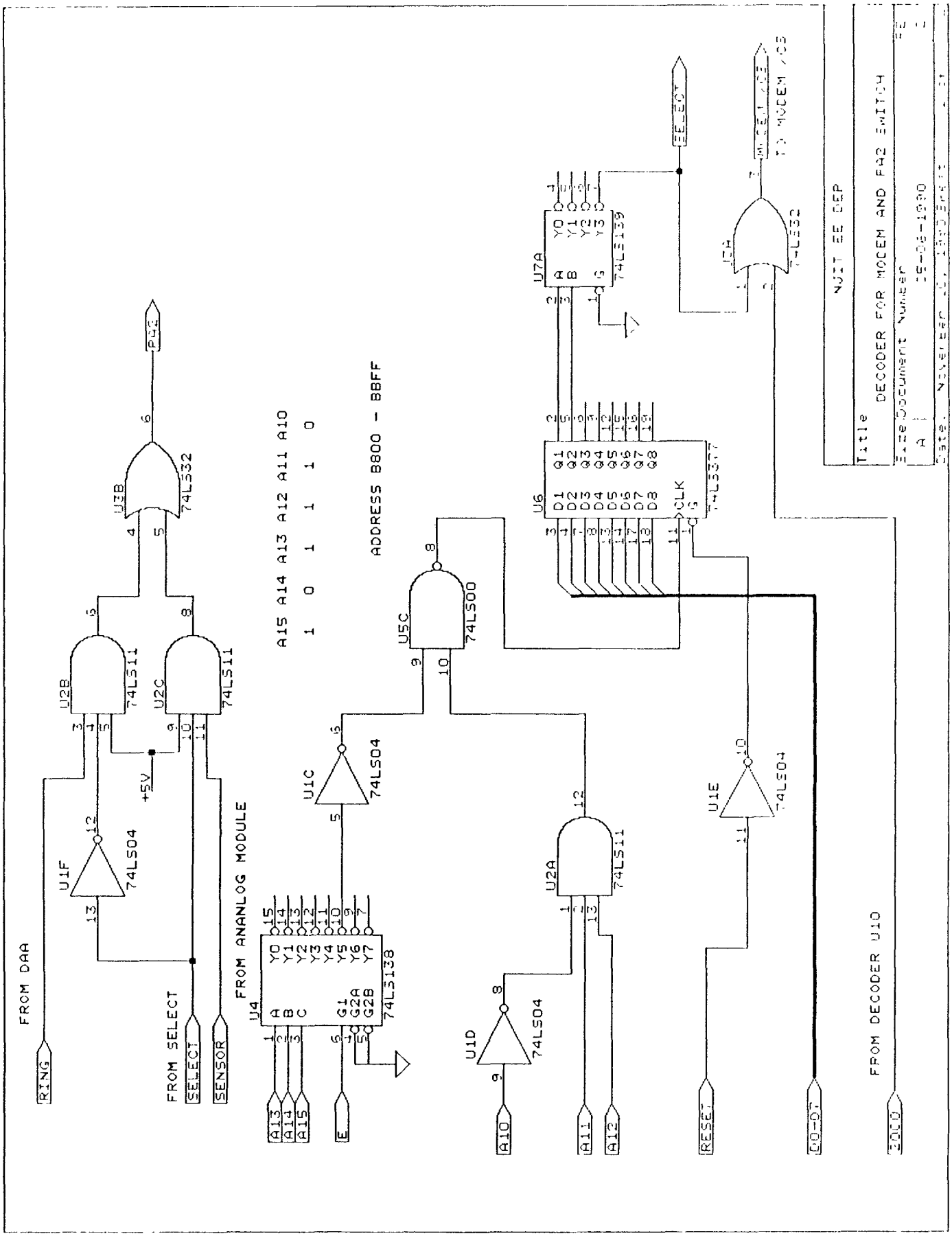


Figure A.1: Modem circuit diagram 1 / 2

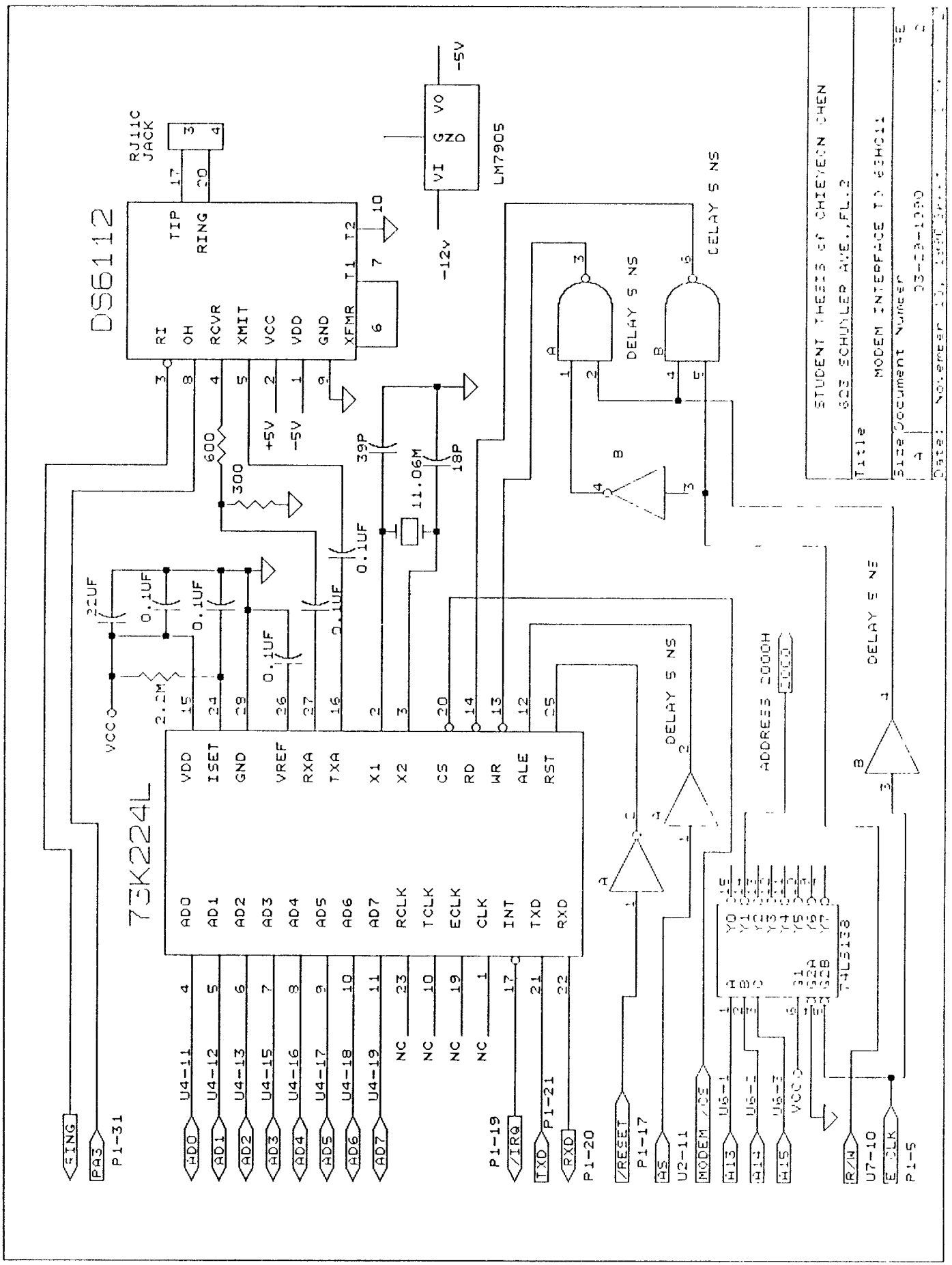
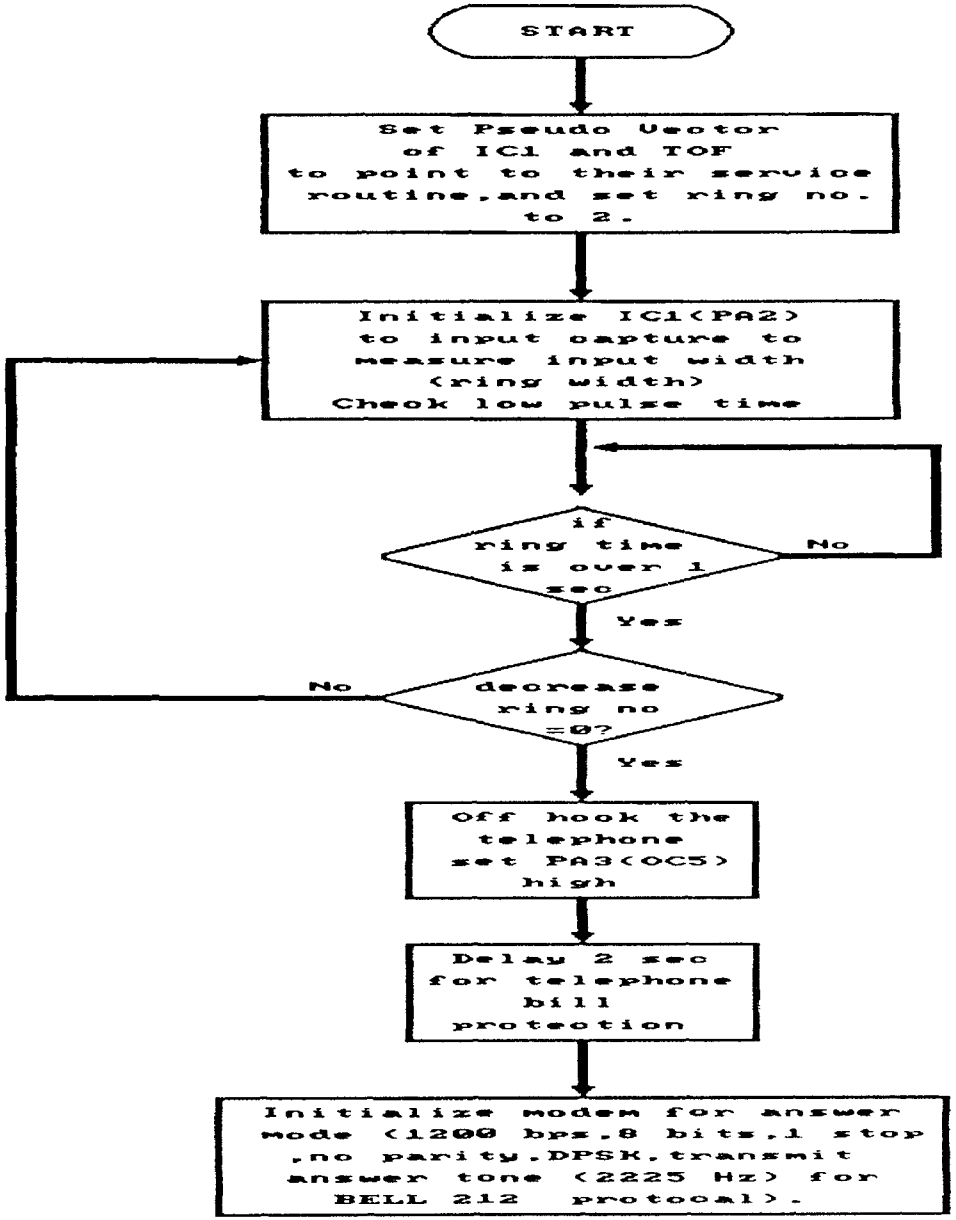


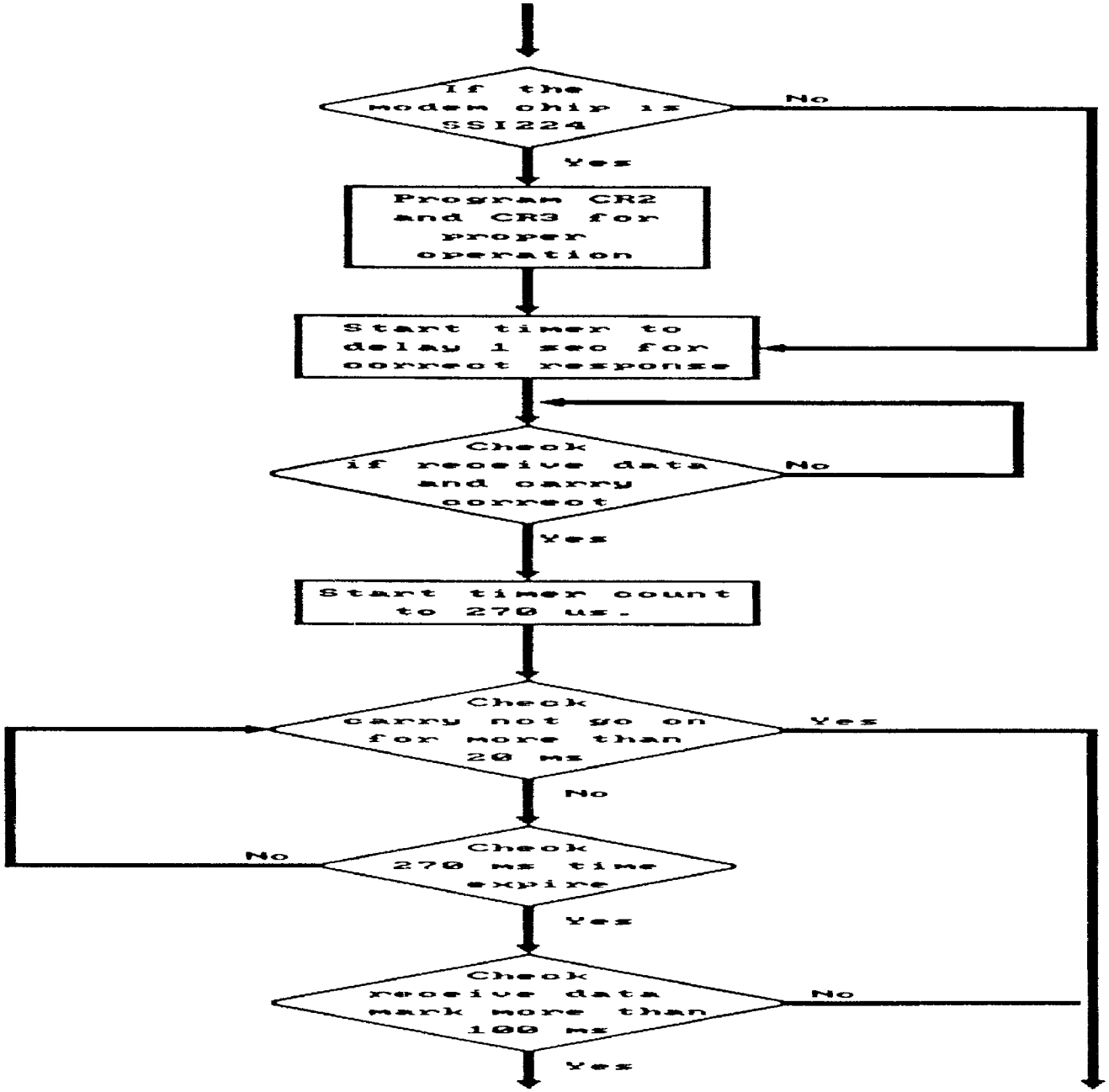
Figure A.2: Modem circuit diagram 2 / 2

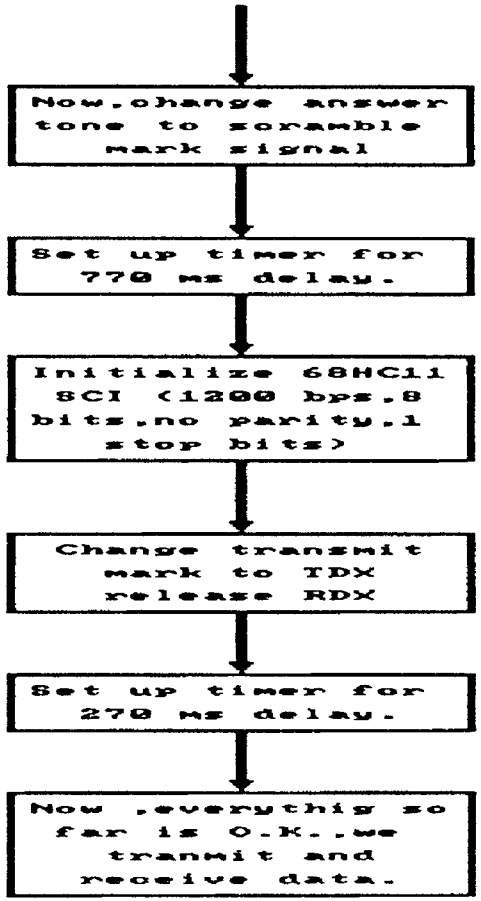
Appendix B

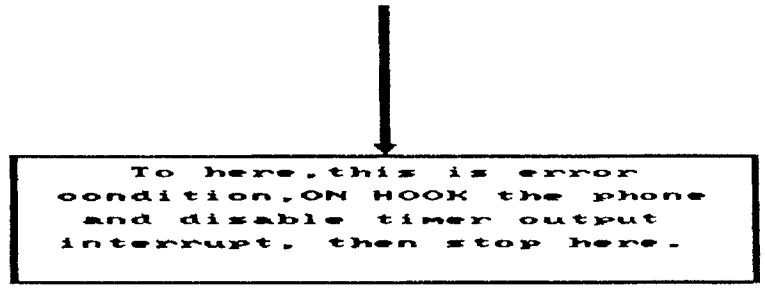
Flowchart for Microcontroller Software

MODEM PROGRAMMING FLOW CHART

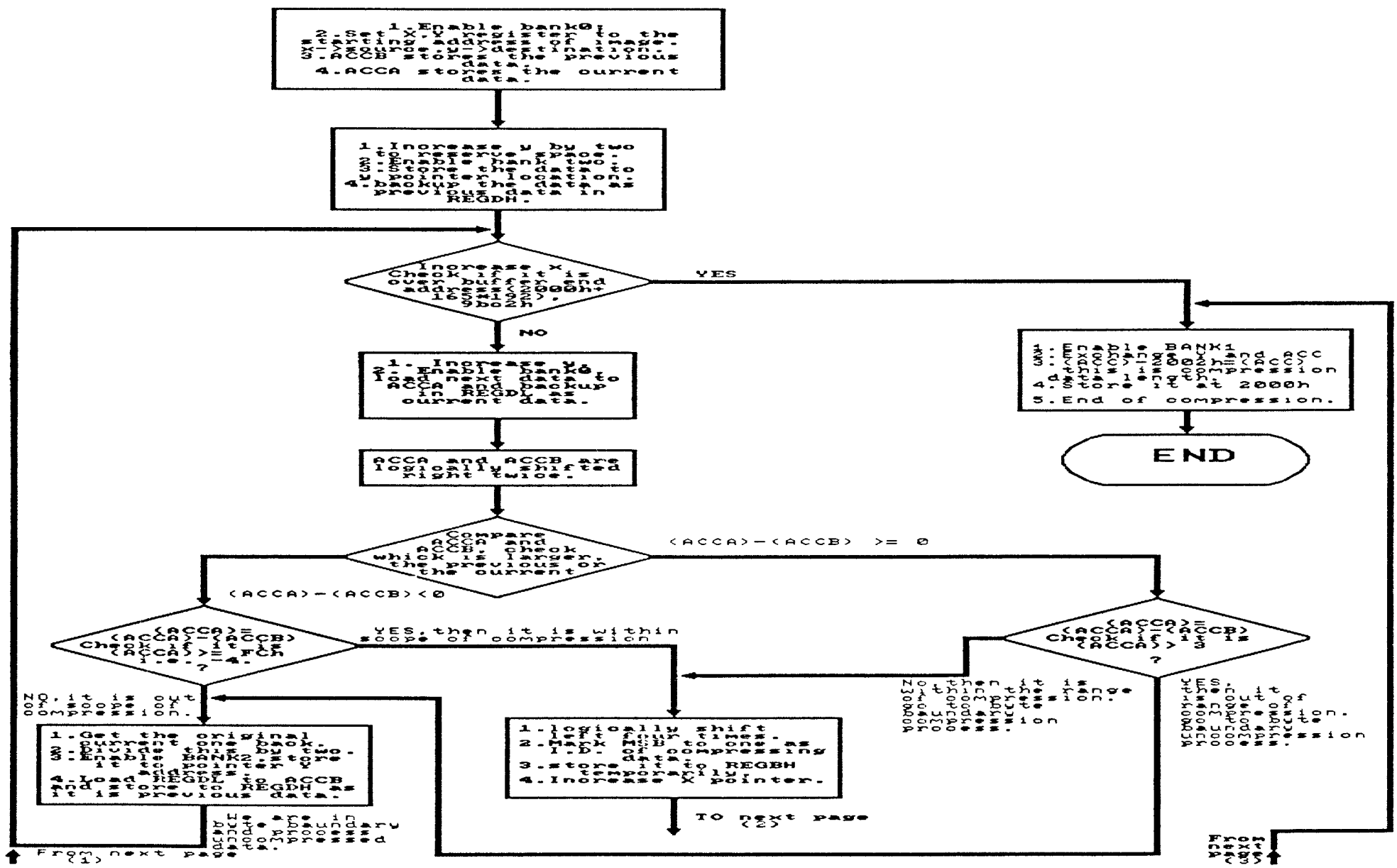


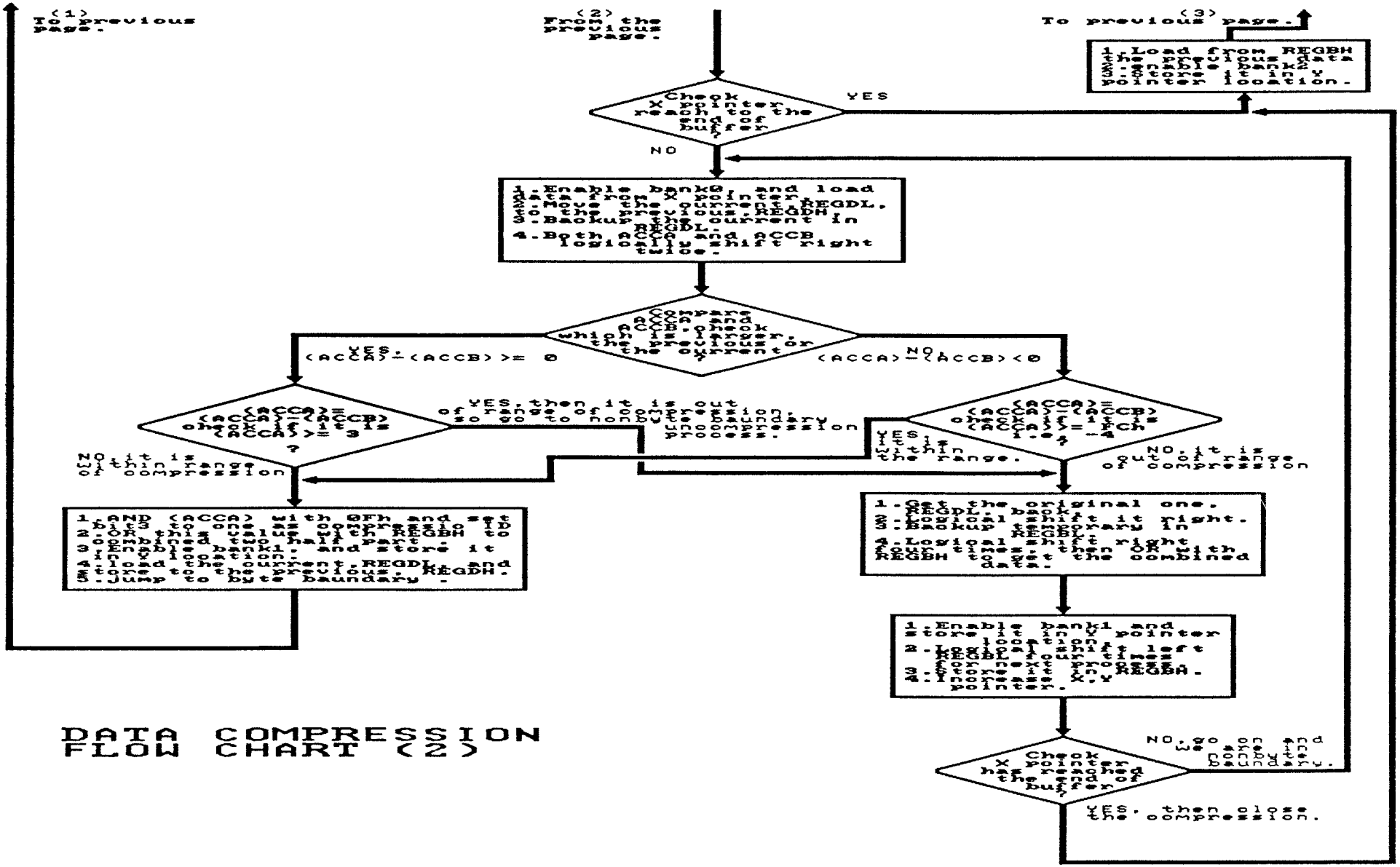

If the modem chip is SSI224
No
Yes
Program CR2 and CR3 for proper operation
Start timer to delay 1 sec for correct response
Check if receive data and carry correct
No
Yes
Start timer count to 270 us.
Check carry not go on for more than 20 ms
Yes
No
Check 270 ms time expire
No
Yes
Check receive data mark more than 100 ms
No
Yes


Now,change answer tone to scramble mark signal
Set up timer for 770 ms delay.
Initialize 68HC11 SCI (1200 bps,8 bits,no parity,1 stop bits)
Change transmit mark to TDX release RDX
Set up timer for 270 ms delay.
Now ,everythig so far is O.K.,we tranmit and receive data.


To here,this is error condition,ON HOOK the phone and disable timer output interrupt, then stop here.

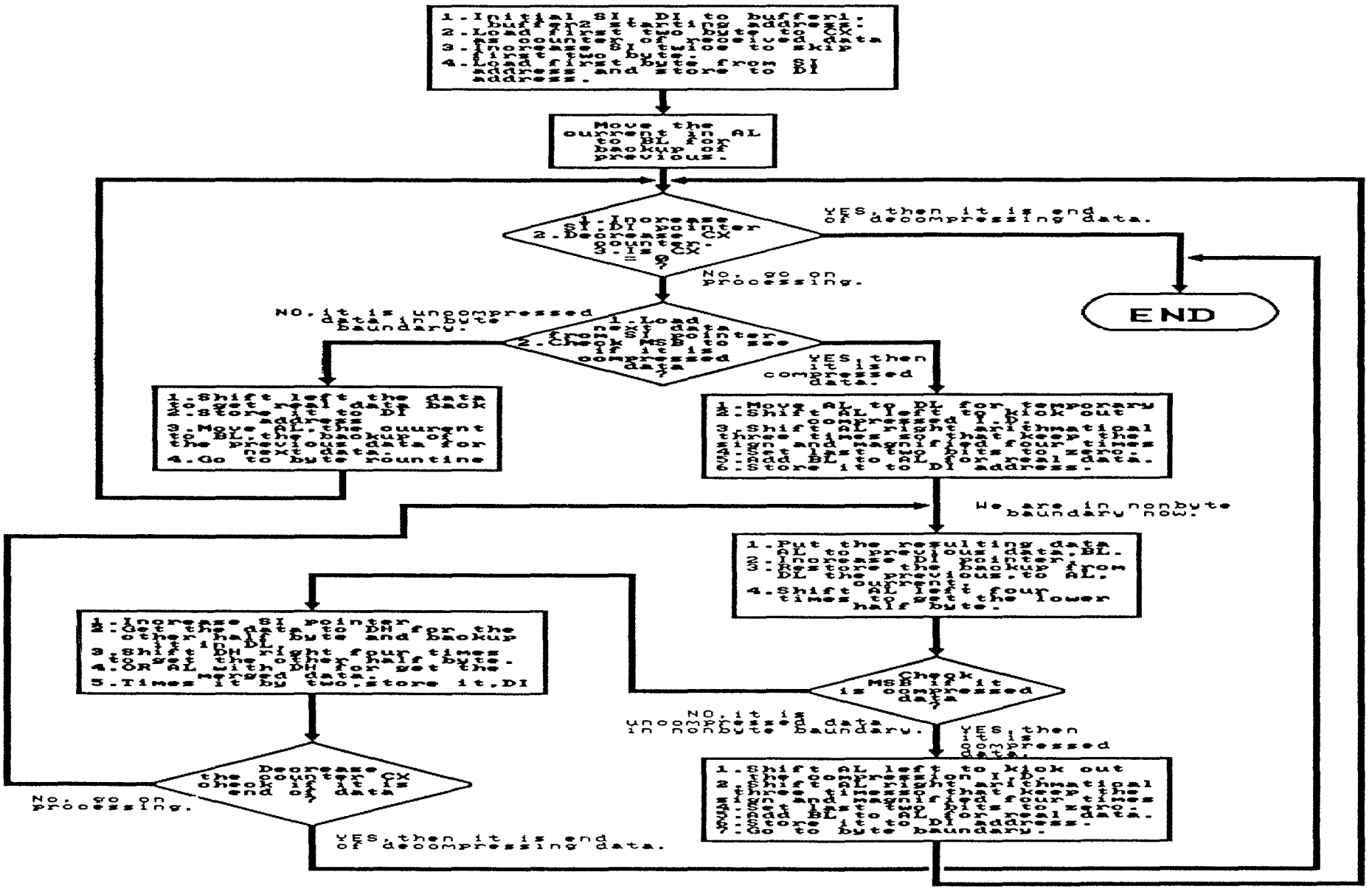
DATA COMPRESSION FLOW CHART





DATA COMPRESSION FLOW CHART (2)

DATA DECOMPRESSION FLOW CHART



Bibliography

[1] Li, Chishin,"Microcomputer-based CCD Imager," Masters thesis, New Jersey Institute of Technology, 1989.

[2] Feng, Hsueh-Chung Michael, "Microcomputer based Machine Vision System," Masters thesis, New Jersey Institute of Technology, 1990.

[3] Qi, Xiao-Fang, "Remote Vision Controller and Data Logger System," Masters thesis, New Jersey Institute of Technology, 1990.

[4] "Silicon Systems Communications Products 1989 Data Book," Silicon Systems, Inc., 1989.

[5] "K-Series Modem Design Manual," Silicon Systems, Inc., 1989.

[6] Joe Campbell: "C Programmer's Guide to Serial Communications," Howard W. Sams & Company, 1989.

[7] Allen L. Wyatt: "Using Assembly Language," QUE Corporation, 1989

[8] Robert Lafore, "The Waite Group's Microsoft C Programming for the PC", Howard W. Sams & Company, 1989.

[9] "M68HC11EVB Evaluation Board User's Manual", Motorola Inc , 1986

[10] "M68HC11 HCMOS Single-Chip Microcontroller Programmer's Reference Manual", Motorola Inc., 1986.

[11] "M68HC11 Reference Manual", Motorola Inc., 1989.

[12] "M68HC11 Single-Chip Microcontroller Reference Manual", Motorola Inc., 1986.