To the Graduate Council:

I am submitting herewith a thesis written by Marcos J. Dominguez S. entitled "Design of a Microcomputer Alarm System using a Power-Line Carrier." I have examined this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

R. W. Rochelle

R. W. Rochelle, Major Professor

We have read this thesis and recommend its acceptance:

Robert E. Bodenheimer

Donald W. Bouldin

Accepted for the Council:

C W Minkel

The Graduate School

DESIGN OF A MICROCOMPUTER ALARM SYSTEM USING A POWER-LINE CARRIER

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Marcos J. Dominguez S.

March 1985

A DIOS

A MI LINDA

ACKNOWLEDGMENTS

The author wishes to thank Dr. R. W. Rochelle for his guidance in the preparation of this thesis. Appreciation is also expressed to Dr. R. E. Bodenheimer and Dr. Bouldin for their support as committee members.

The author would like to thank Miss. Francis Delgado for her help in reading the final copy for typographical errors. Finally, the author wishes to thank to his parents, sister and brother for their ever devoted support and to the Fundacion Gran Mariscal de Ayacucho for sponsoring his master study.

ABSTRACT

Currently, microcomputer technology is assigned to specific tasks, such as intercomputer communication and process control. However, a relatively unexplored area of microminiaturization is the application of power-line carriers in microcomputer alarm systems. The purpose of this thesis is to establish a design system which will introduce the power line as the carrier into the area of microcomputer alarm systems. This design is divided into two parts. The first part is the design of the power-line modem that is the interface between the power line and the microcomputer; the second part is the design of the necessary software to make the system operable.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Security is essential in the effective operation of any business. With the increase in burglaries, more and more businesses are turning to alarm systems for increased protection.

The selection of a suitable and reliable alarm system is not a simple matter . Consequently, we should examine factors which determine the requirements of an individual alarm system:

1. The threat--What is to be protected against?

2. The type of protection--What methods are available?

3. The types of sensors--What should be detected?

4. The method of alarm transmission--How is the signal sent and who will respond?

All of the factors listed above lead us to think of the primary function of an alarm system, which is to signal the presence of the intruder. An intrusion alarm system is only part of the overall protection needed; many large businesses will supplement this system with guards and other security personnel, and even the smallest operation will have a locked door. In addition, it is important for us to recognize that the successful operation of a system depends upon not only the system equipment but also upon its proper installation, maintenance, and use.

In general the alarm system is composed of individual intrusion alarm sensors and a master control unit. The individual intrusion alarm sensors are usually connected to a master control unit that receives their signals and causes a local alarm to sound, or alerts the police or the central station to the fact that the system has entered the alarm state or intruder state. The connection between the sensors and the master control unit must be done with electrical wire.

The cost of installing and operating an alarm system will vary significantly, depending upon the type of system, the type of electrical wires and the community in which the business is located. As a rule, we can expect that the cost of installing will be approximately equal to the cost of the components that make up our system.

The main idea of this thesis is to design an alarm system that uses the power line as a carrier in order to reduce the cost of installing and to increase the flexibility of the system.

This system is composed of two main elements: the slave microcomputers, which must be connected near the sensors, and the master microcomputer, which is the equivalent of the master control unit.

This system appears be expensive, but with the progress of microminiaturization the cost of a microcomputer will be just a few dollars.

Chapter two of this thesis explains the general concept of a telephone modem and how to design a power-line modem, as well as presenting the electrical specifications of the modem designed.

Similarly, chapter 3 discusses the possible structure of our microcomputer alarm system in terms of both hardware and software, as well as providing some background about error correcting and error detecting code.

CHAPTER 2

MODEM DESIGN

Telephone, TV and radio, mail, periodicals, books, etc are the principal ways that people use to communicate differents ideas, and the sending and receiving of information. These ways are known as communication systems. However, if the information transmitted is digital data, then they are known as data communication systems. Commonly, data communication systems use computers in their networks in order to add the capacity to process more data. The components of this system may be described as follows:

1. The terminal--It produces digital signals which are in the form of the binary numbers one (ON, Vcc) and zero (OFF, Ground).

2. Transmitter Modem--It converts a digital signal to an analog signal or vice versa. The modem is the interface between the terminal and the communication channel.

3. The communication channel--It can be telephone lines, coaxial cables, microwaves transmission, radio transmission, satellite transmission, fiber optics, power line, etc. The most common and convenient communication channel is the telephone line; however, the selection of a communication channel is a function of the system that we want to install.

4. Receiver Modem--It is like the first one, but it is used for the front-end processor.

5. The front-end processor--It is a computer and its main function is to relieve the main computer of the communication tasks.

6. The computer--Commonly it is a mainframe or a personal computer.

As we can see, it is possible to relate the components of our microcomputer alarm system to the components of the computer data communication system. They may be related in the following way: the terminals are the slave microcomputers; the communication channel is the power line; the front-end processor and the computer are the master microcomputer; and the modems are the power-line modems. The function of a microcomputer is a function of its own program; but what exactly is the function of a modem? How does it work? To answer these questions it is necessary to know how a telephone modem works.

The purpose of the telephone modem is to translate the digital signals which computers or microcomputers use into a form compatible with the telephone system. Modems operate in originate mode (terminal modem) and answer mode (computer modem). The difference between the two modes lies only in the frequencies used by the modem. The basic elements of the modem are the modulator--demodulator, the buffer circuits, the carrier detector, and the ring and dial--tone detectors. The modulator modulates the digital signal in FSK (Frequency Shift Keying) where a zero is represented by 1070 Hz and a one by 1270 Hz, at the originate mode, and by 2025 Hz and by 2225 Hz, at the answer mode. These frequencies are for a low-speed asynchronous modems (transmission rates up to 600 baud). For higher-speed asynchronous modems (1200 baud), a zero is represented by 1200 Hz and a one by 2200 Hz. The reason these values must be chosen is because the bandwidth of the telephone lines is between 300 Hz and 3000 Hz. Since the originate modem and the answer modem work at different frequencies, the modem may operate in full-duplex or half-duplex, where full-duplex is a mode in which transmission and reception may occur at the same time and half-duplex is the opposite. The demodulator demodulates the FSK signal producing the original digital data. The buffer circuits produce compatibility with the telephone lines. The carrier detector is a signal that determines if the modem is operating or not. The ring and dial-tone detectors detect the ring and dial tones if they are presented in the telephone lines.

The main differences between the power-line modem designed in this thesis and the telephone modem are:

	Telephone modem	Power line modem
Lines	telephone lines	power lines
Modulation	FSK	AM
Carrier detec.	yes	no
Modes	Originate, Answer	None
Operation	Half and Full Duplex	Half-Duplex
Carrier frequ.	Between 300-3000 Hz	Between 100-400 Khz
Buffer circuits	yes	yes

This power-line modem is composed of five functional blocks (Fig 2-1):

1. Modulator
2. Demodulator
3. Input-Output Transformer
4. Impedance Adapter
5. Power Supply

The Modulator

This first building block provides the carrier frequency which is amplitude modulated by the digital signal (Fig 2-2). The heart of this block is the integrated circuit XR-2206 made by Exar.

The XR-2206 is a monolithic function generator which can operate over a frequency range of 0.01 Hz to more than 1 Mhz.

The selected carrier frequency is 180 Khz which was chosen for the following reasons:

a. This frequency is not in the spectrum range from 550 Khz to 1600 Khz that is the A.M radio broadcasting range; therefore it does not produce radio interference.

b. The impedance of the power line can be assumed inductive so that the impedance increases when the frequency increases.

c. The demodulator works into a range of 0.01 Hz to 500 Khz.

The frequency of oscillation (fo) is determined by the external timing capacitor C, and the resistor R. The frequency is given by:

$$fo = \frac{1}{R\ C} \tag{2-1}$$

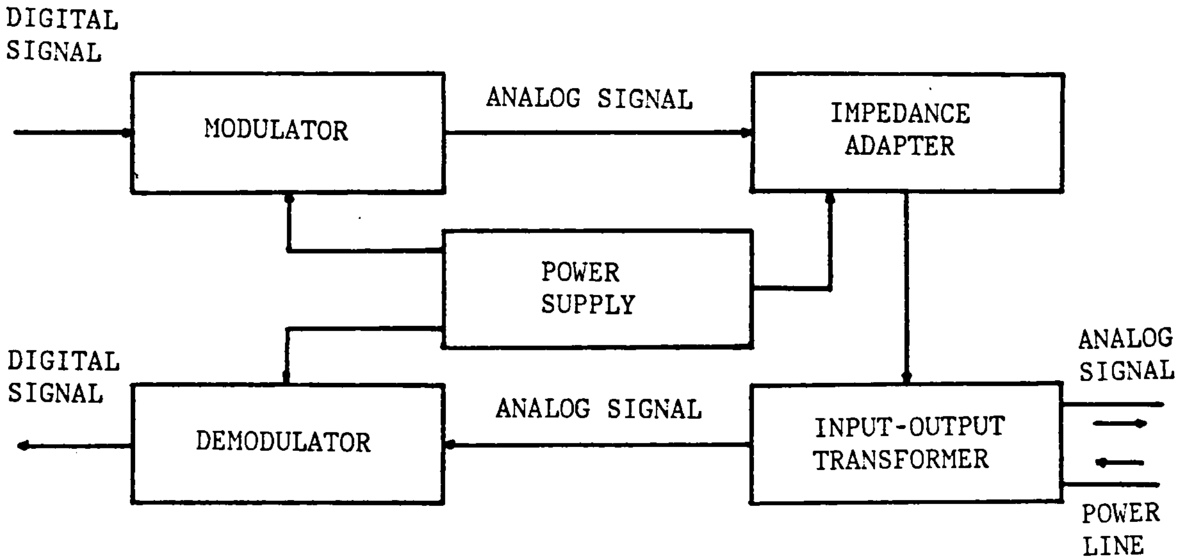


Figure 2-1. Block diagram of a modem.

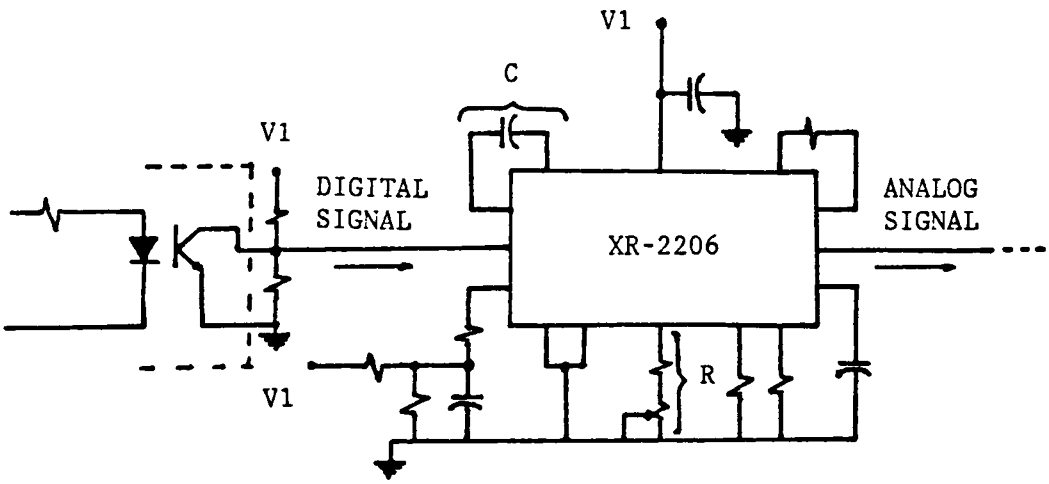


Figure 2-2. AM Modulator

and can be adjusted by varying either R or C. In this implementation of the design, R is varied to adjust the carrier frequency fo.

The Demodulator

This second building block amplifies and demodulates the analog signals. This block is divided into two circuits, a signal amplifier and a tone detector (Fig 2-3).

The signal amplifier uses a single low-noise operational amplifier (NE 5534 N, Signetics) with a small-bandwidth of 10 Mhz.

The purpose of this signal amplifier is to increase the level of the input, increasing the sensitivity of the modem. With a gain of eleven, the sensitivity is approximated 5 mV rms.

The operational amplifier works as a noninverting amplifier (Fig 2-3), and its gain is given by:

$$A = \frac{Vo}{Vi} = 1 + \frac{R2}{R3} \tag{2-2}$$

and the input impedance (Ri) is approximately

$$Ri = \frac{Ra\ Rb}{Ra + Rb} \tag{2-3}$$

The transformer T1 works as a coupled transformer between the tone detector and the signal amplifier. The main function of T1 is to isolate the two power supplies to avoid the power line noise that is in V1.

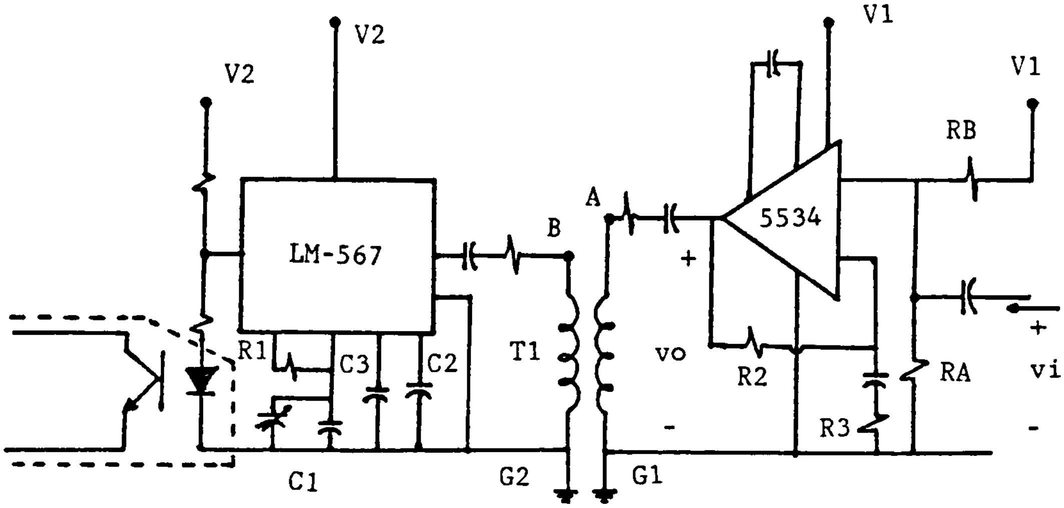


Figure 2-3. AM Demodulator.

If the power supplies have the same ground level, and if the power line noise in V1 is insignificant, then other designs may eliminate the transformer T1 and connect the terminals A with B and G1 with G2.

The tone decoder is the IC LM 567. This IC is a general purpose tone decoder designed to provide a saturated transistor switch to ground when an input signal is present within the passband. It can adjust its center frequency from 0.01 Hz to 500 Khz, and the smallest detectable input voltage is 20 mV rms.

The center frequency (fo) of the tone decoder is equal to the free running frequency of its VCO (voltage controlled oscillator). This carrier frequency, fo, is given by:

$$fo = \frac{1}{1.1\ R1\ C1} \tag{2-4}$$

and the bandwidth is fixed with C2 and C3.

Care is needed when selecting the bandwidth, because the number of input signal cycles that the tone decoder needs to detect the input signal increases as the bandwidth decreases. If the amount of cycles is big, then it will generate a delayed output at the receiver producing a timing error.

The Input-Output Transformer

This third building block provides the isolation between the power line and the modem.

Since the analog signal goes through the input-output transformer, it must have a wide bandwidth to avoid an attenuated analog signal; therefore, T2 should be a pulse transformer.

To study the passive network, shown in Figure 2-4, it is necessary to find the equivalent network at the power-line side. An electrical transmission line model (Fig 2-5) is used in this calculation.

The electrical transmission line model (see figure 2-5) is composed of two impedances XLdx and XIdx. Both impedances, in this case, depend on the distance between the transmitter and the receiver. Assuming that the distance is short, XL is almost a short circuit and XI is almost an open circuit. Therefore, we can ignore XLdx and XIdx.

For the equivalent network (Fig 2-6), letting Ft be much bigger than Fp and assuming an inductive Xp, then some conditions have to be met:

	At Fp	At Ft
1.	$Xt2 \longrightarrow \infty$	$Xt2 \longrightarrow 0$
2.	$Xt1 << Xt2$	$Xt1 >> Xt2$
3.	Xp is a variable load	$Xp > Xp$ at Fp
4.	$Xp // (Xt2+Xt1) \rightarrow Xp$	$Xp//Xt1 \rightarrow Xp$

Since the transmitter and the receiver are the same, its input-output impedances are also the same, and they are related by:

$$Xt1 = \frac{Xc1\ R1}{Xc1 + R1} \tag{2-5}$$

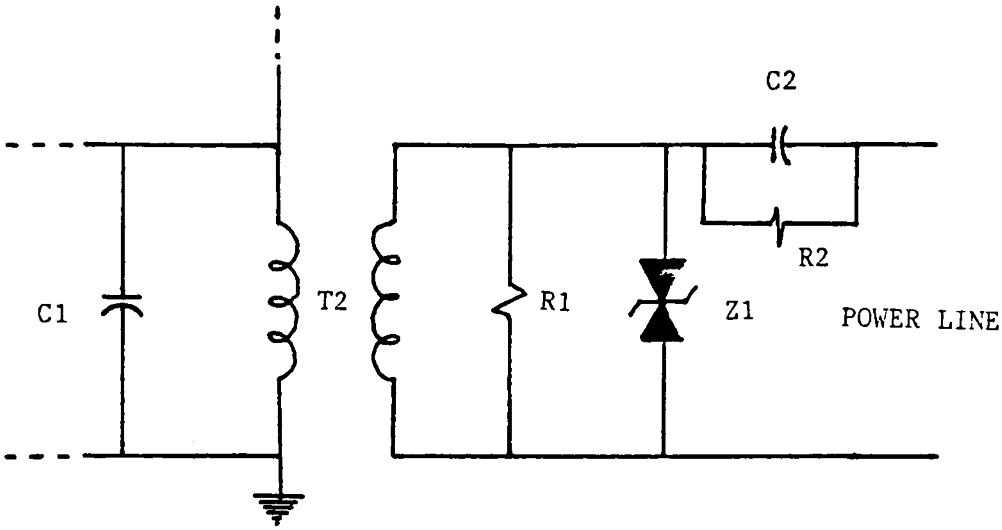


Figure 2-4. Input-Output Transformer

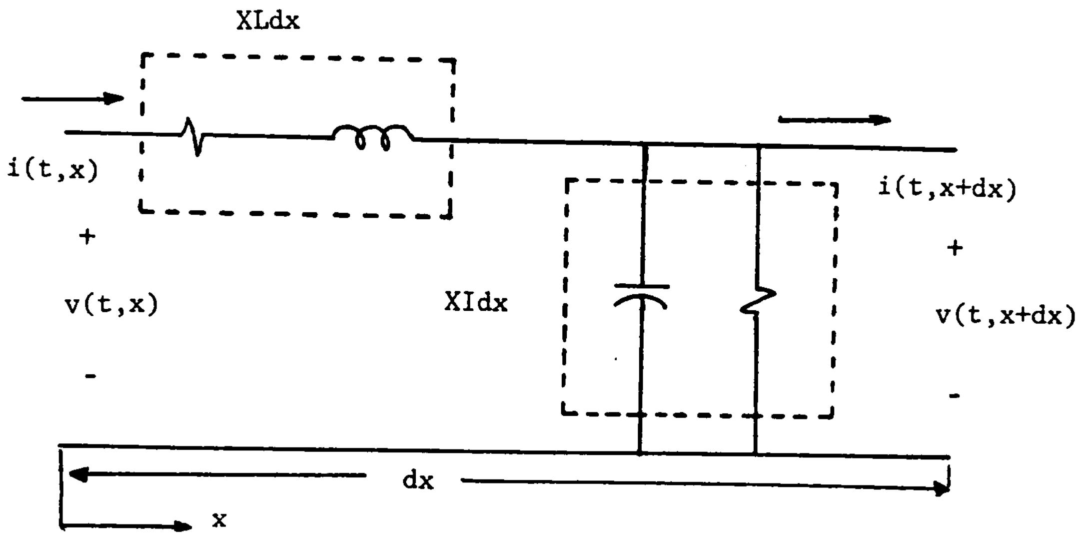


Figure 2-5. Electrical Transmission Line Model.

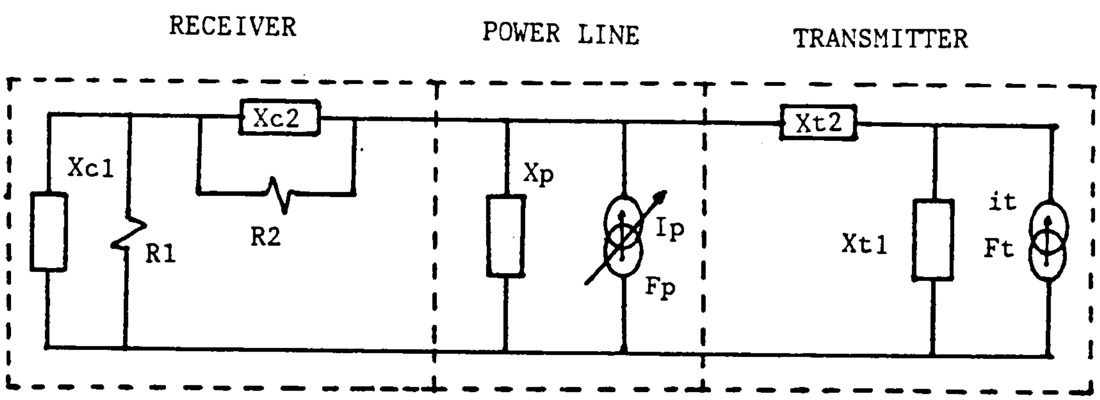


Figure 2-6. Equivalent network at the power line side with one receiver and one transmitter.

$$Xt2 = \frac{Xc2\ R2}{Xc2 + R2} \tag{2-6}$$

It might appear that the capacitor C1, the resistor R2, and the voltage spike protector Z1 have no function at all; however C1 improves the shape of the analog signal, R2 discharges the capacitor C2 when the moden is disconnected and Z1 has the capability to absorb high-voltage transients.

The Impedance Adapter

This fourth building block generates the necessary current that the small power line load needs in order to produce a significant output voltage. This block only has one pnp-transistor (2N3906 with a beta=300), two resistors and one capacitor (Fig 2-7).

The equivalent DC circuit is shown in Figure 2-7(b). Applying Kirchhoff's voltage law

$$V1 = Ie\ R1 + Vbe + Ib\ R2 + \frac{V1}{2} \tag{2-7}$$

using the relations

$$Ie = Ib + Ic \tag{2-8}$$

$$Ic = \beta\, Ib \tag{2-9}$$

and the approximation

$$Vbe \cong 0.7\ \text{Volts} \tag{2-10}$$

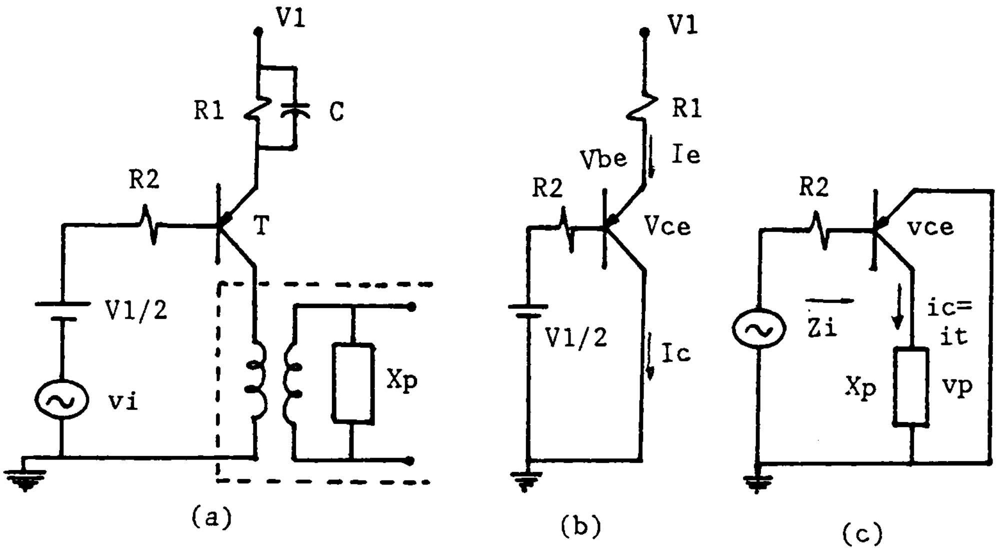


Figure 2-7. a) Common-Emitter Amplifier, b) Equivalent DC Circuit, c) Equivalent AC Circuit.

$$\beta \cong \beta + 1 \tag{2-11}$$

we get the DC base current equation

$$Ib = \frac{(V1/2) - Vbe}{(1 + \beta)\ R1 + R2} \tag{2-12}$$

and the DC collector current as a function of Vce is

$$Ic = \frac{V1 - Vce}{R1} \tag{2-13}$$

which describes the DC load line for this circuit. This load line is superimposed on the transistor characteristics (Fig 2-8). The intersection between the DC load line and the transistor characteristics is the Q-point which shows the values of Ic, Vce and Ib.

The AC circuit configuration is shown in Figure 2-7(c). We may note that the capacitor C provides a short circuit like the power supplies V1 and V1/2. The small-signal parameters (AC parameters) can be found using the h-parameter model or graphic modes. This design uses the graphic mode since it is easier to visualize how the output voltage is affected by the power line load. The small-signal parameters in the collector circuit are related by the equation

$$ic = it = \frac{-vce}{Xp} \tag{2-14}$$

where ic, vce, Xp are phasors. This equation describes the AC load line for this circuit which is superimposed on the transistor characteristics passing through the Q-point with a slope equal to -1/Xp. Looking at the

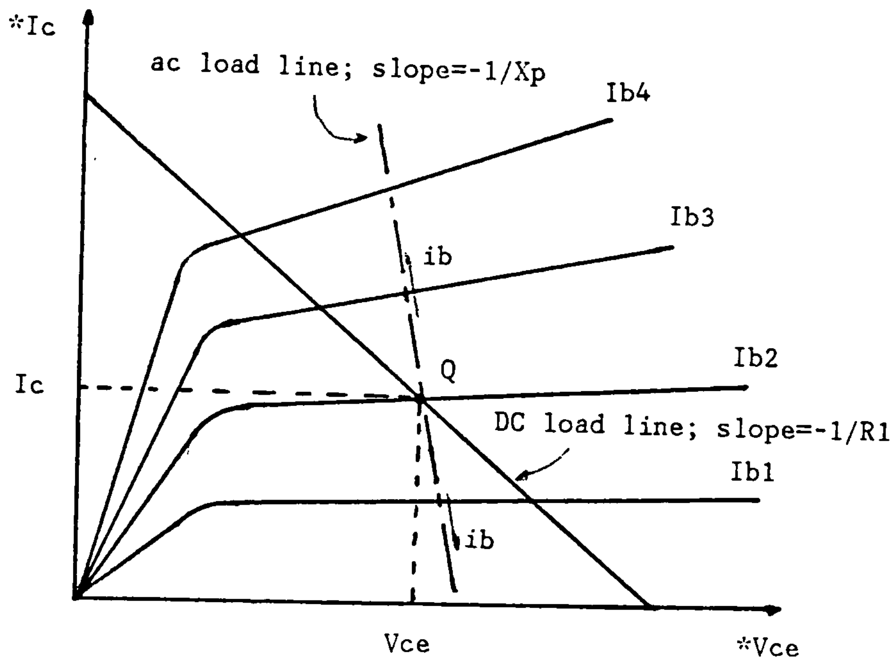


Figure 2-8. The AC and DC load line plots for the circuit of Figure 2-7.

AC load line, and the power line load, we can observe that the output voltage is very small. We can then approximate the power dissipation of the transistor (Pd) which is equal to Vce times Ic.

At this point, all the parameters are found; therefore, the voltage output (Vp) is known. Since the stability of this block is not very good and the impedance Xp is not a constant in time, the output voltage (Vp) is variable in time. However, these problems do not produce a big effect at the receiver end because the sensitivity of this modem is high and the tone detector does not depend on the shape of the analog signal.

The Power Supply

This fifth and final block provides the necessary voltages and currents that the modem requires. In this case the modem uses the power supplies that are facilitated from the microprocessor systems research lab.

Due to the fact that this design is an alarm system, it must have a battery back up in order to maintain the system in operation when the power line suddenly goes down.

If the design of the power supply does not have a transformer, then a radio-frequency choke coil is added to avoid the attenuation of the carrier frequency.

General Characteristics

The general characteristics are shown in table 2-1.

Table 2-1

CHARACTERISTICS

	Receiver	Transmitter
Sensitivity	5 mV rms	----------
Output Vol.	--------	1.6 Vols p-p
Carrier Fre.	180 Khz +10%	180 Khz +10%
Bandwidth	6%	----------
Sine Wave Dis.	--------	< 2.5%
Power Suppy	10 & 5 Vols	10 & 5 Vols
Power Dissipa.	750 m.Watts	750 m.Watts
Cycles Before output	80 cycles	----------
Max. Operating frequency	500 Khz	1 Mhz
Lowest Practical frequency	.01 Hz	.01 Hz

CHAPTER 3

SOFTWARE DESIGN

All the sophisticated hardware of our system is useless without sophisticated software to make it run in accordance with our wishes.

The software discussed in this chapter is necessary to produce a communication between two microcomputers which work under special conditions. Additionally, an idea of a complete alarm system is given.

General System

This alarm system is developed using microcomputers. These microcomputers are divided in two categories, the master microcomputer and the slave microcomputers. The slave microcomputers detect an intrusion if it occurs; and the master microcomputer, by using the information of the slave microcomputers, determines whether or not the alarm should be activated.

A typical system would be composed of one master microcomputer, as many as sixteen slave microcomputers, one alarm, and sensors (Fig 3-1).

If we want a more sophisticated alarm system, then the only component that we need to change is the master microcomputer for a personal computer in order to increase the performance of the initial system. As we have seen, the master microcomputer can be a personal computer, but the slave microcomputers could be designed using inexpensive components in order to reduce the production cost, which will automatically reduce the final cost of the alarm system.

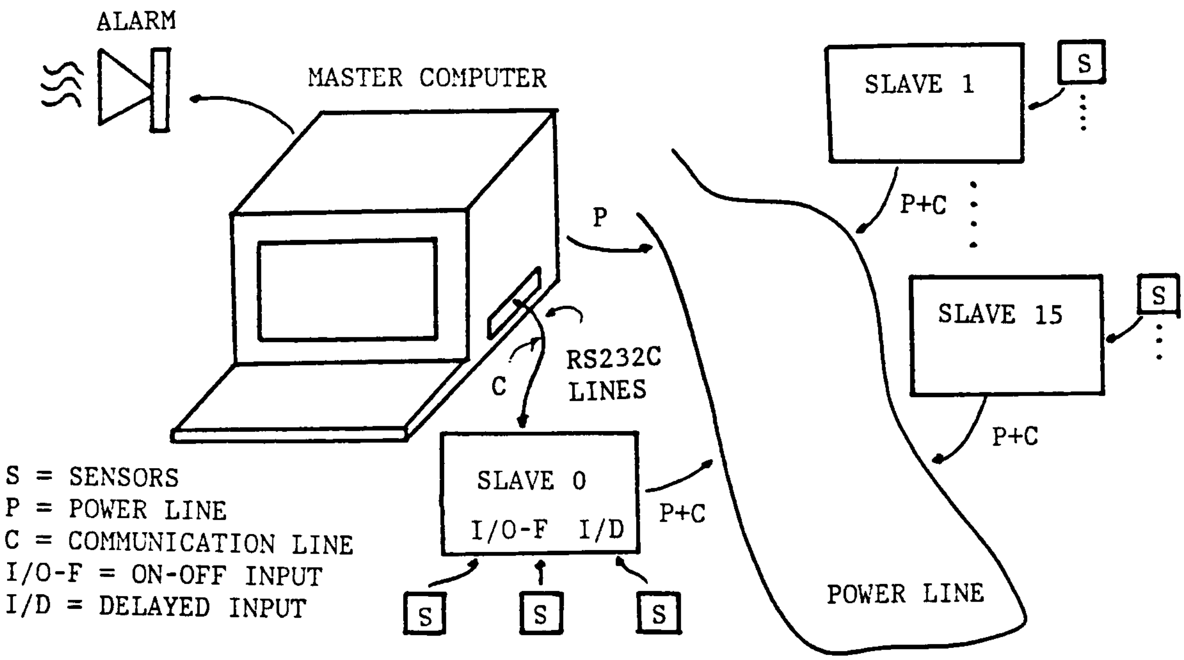


Figure 3-1. General alarm system.

At this point, the term master may be used to identify the master microcomputer, and the slave to identify the slave microcomputer. Similarly, the slave microcomputers may be divided into slave zero, and slave N, where N is a integer between 1......15. As shown in figure 3-1, the slave zero is more complex than the other slaves. The functions of the slave zero are:

1. Protection of the master microcomputer (Tamper Protection)

2. Generation of a standard communication between the slave microcomputers and the master microcomputer; using the RS232C format. These functions are known as slave zero option.

On the other hand, the other slaves just have one function, that is to detect an intruder, and this function is known as normal option. The designer decides whether the slaves will offer both options, slave zero option and normal option, to the user or not. The slaves must have a rotary switch to select its own number (one of the sixteen), a push button to start, a rotary switch to select the carrier frequency, and maybe a switch to select the option.

Our last component is the sensors. The number of sensors that can be connected is a function of the input-output ports that the slaves have, and the mode of detection, on-off input (I/O-F), delayed input (I/D), etc, is a function of its own software.

However, in this project the designed system is composed of one master and one slave. The microcomputer used is the 8048 IMSAI SYSTEM.

In order to know how our design works, we need to study the operational steps of the design, which are stated as follows (Fig 3-2):

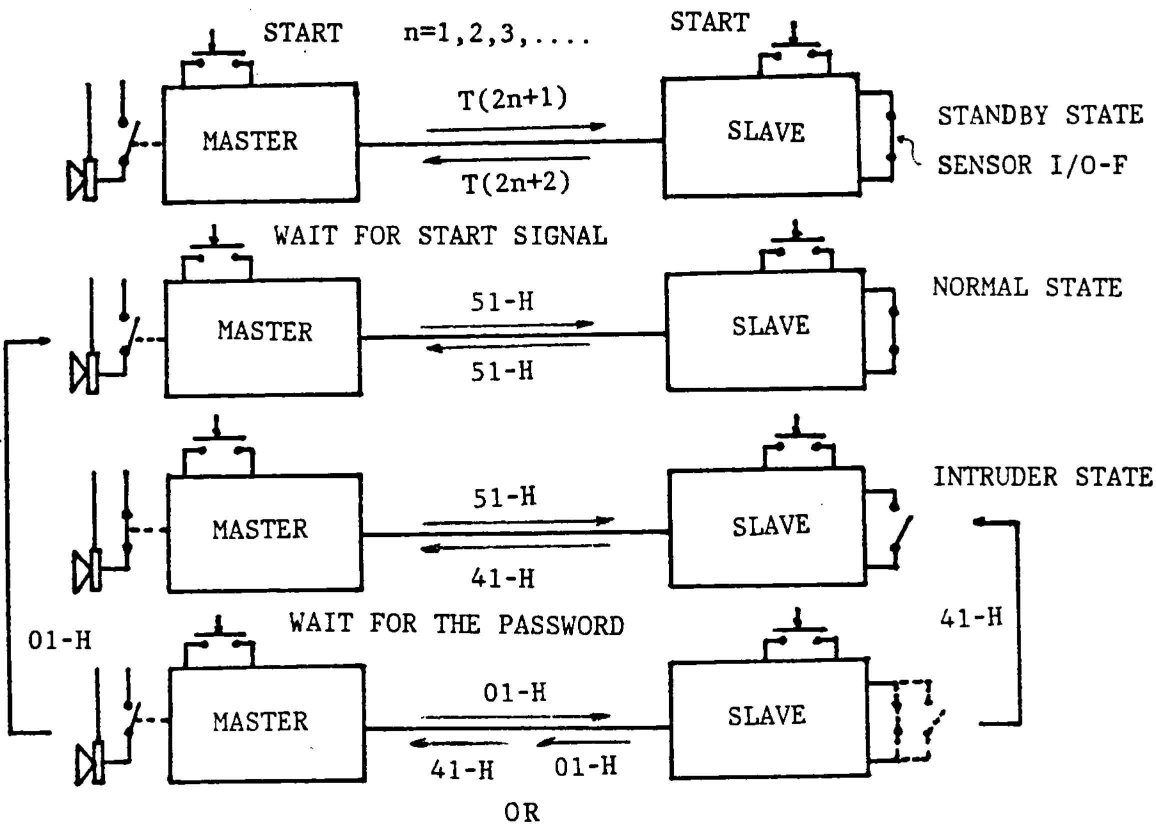


Figure 3-2. Operational steps of the alarm system.

1. When the power is applied, both microcomputers will go to the standby state.

2. To leave the standby state, one must press the start push-button on the slave and on the master, in the same order, to begin the operation.

3. The communication between the master and the slave begins. This state is known as the normal operation state. In this state the master transmits the code 51-H and the slave responds with the same code if no intrusion is detected. However, if the master does not detect the slave's transmission, the master will activate the alarm.

4. If an intrusion is detected, the slave responds with the code 41-H instead of 51-H, and the alarm is activated by the master. This state is known as the intruder state.

5. In the intruder state the master waits for the reset password which will be introduced by the user. When the password is introduced to the master, it sends the reset code 01-H and the slave could respond with one of the following codes:

i. If the intruder's signal disappears, it responds with the code 01-H.

ii. If the intruder's signal persists, it responds with the code 41-H.

As a consequence, the system will become stable in the normal operation state, or in the intruder state.

In order to implement these operational steps in our microcomputer alarm system, the master program (Fig 3-3) and the slave program (Fig 3-4) were developed. The routines that have a star (*) imply monitor routines.

Error Correcting Codes

Since the power line is a random source of noise, the use of error detecting and correcting codes is needed. This design uses a single error correction plus a double error-detection code. To perform the single error correction the system uses the Hamming code (7,4), and to perform the double error detection the system just adds one more parity check that looks over the whole message. Thus, the final code is (8,4).

How does the Hamming error correcting code work to correct single error over four binary digits? It works as follows:

The inequality

$$2^{m} \geqslant (m + k) + 1 \tag{3-1}$$

must be satisfied for a single error correction code, where

"k" is the number of bits of the initial message (in this case k=4)

"m" is the number of independent parity checks and

"(m + k) = n" is the number of bits of the encoded message.

For "k=4", one can observe that for "m" greater or equal than three, the inequality is satisfied. Therefore, one has to select the value of "m" equal to three in order to obtain the minimum length of the

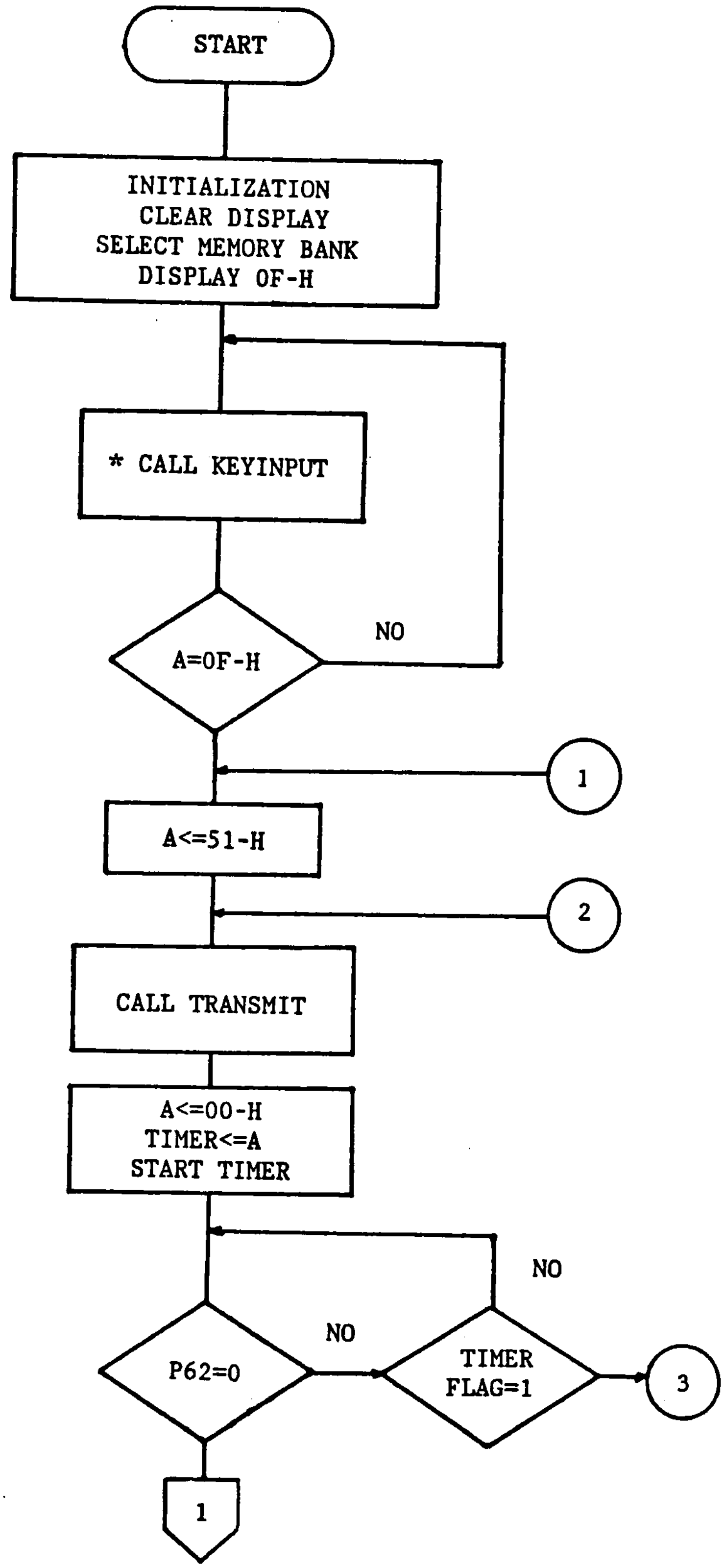


Figure 3-3. Master Program Flow Chart.

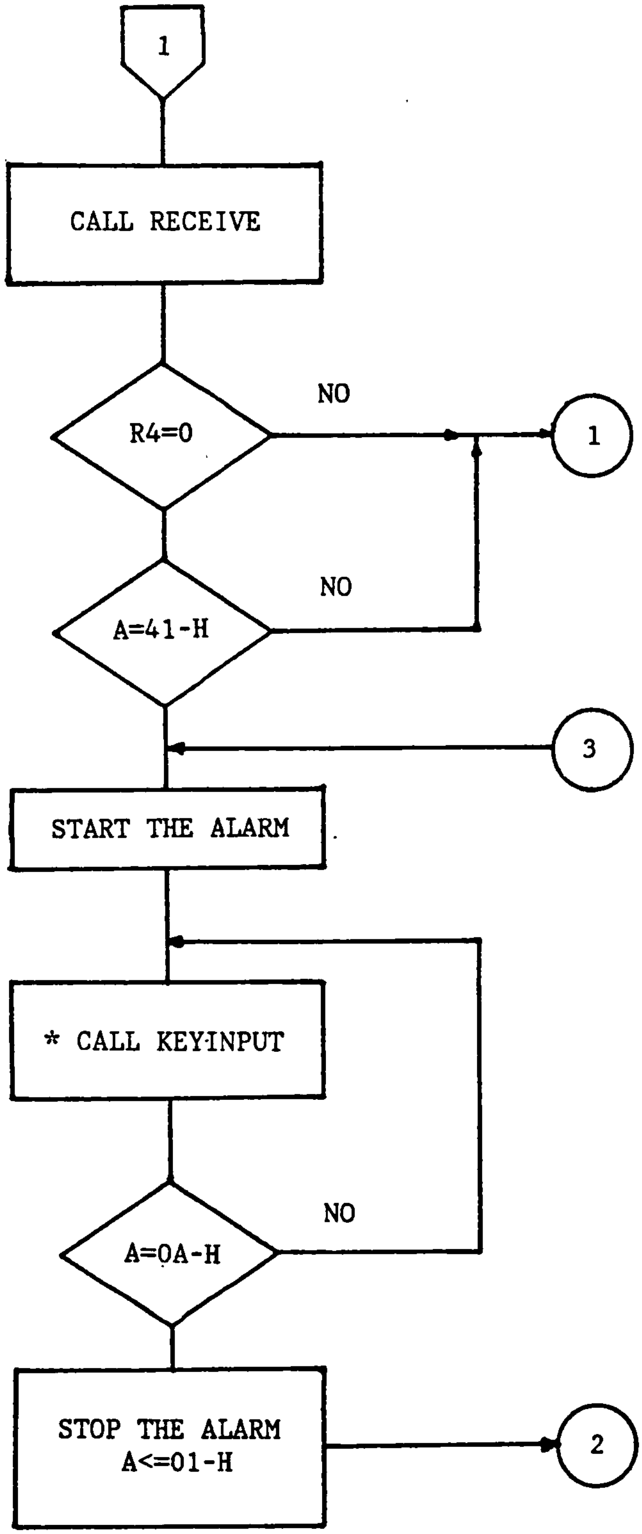


Figure 3-3. (Continued).

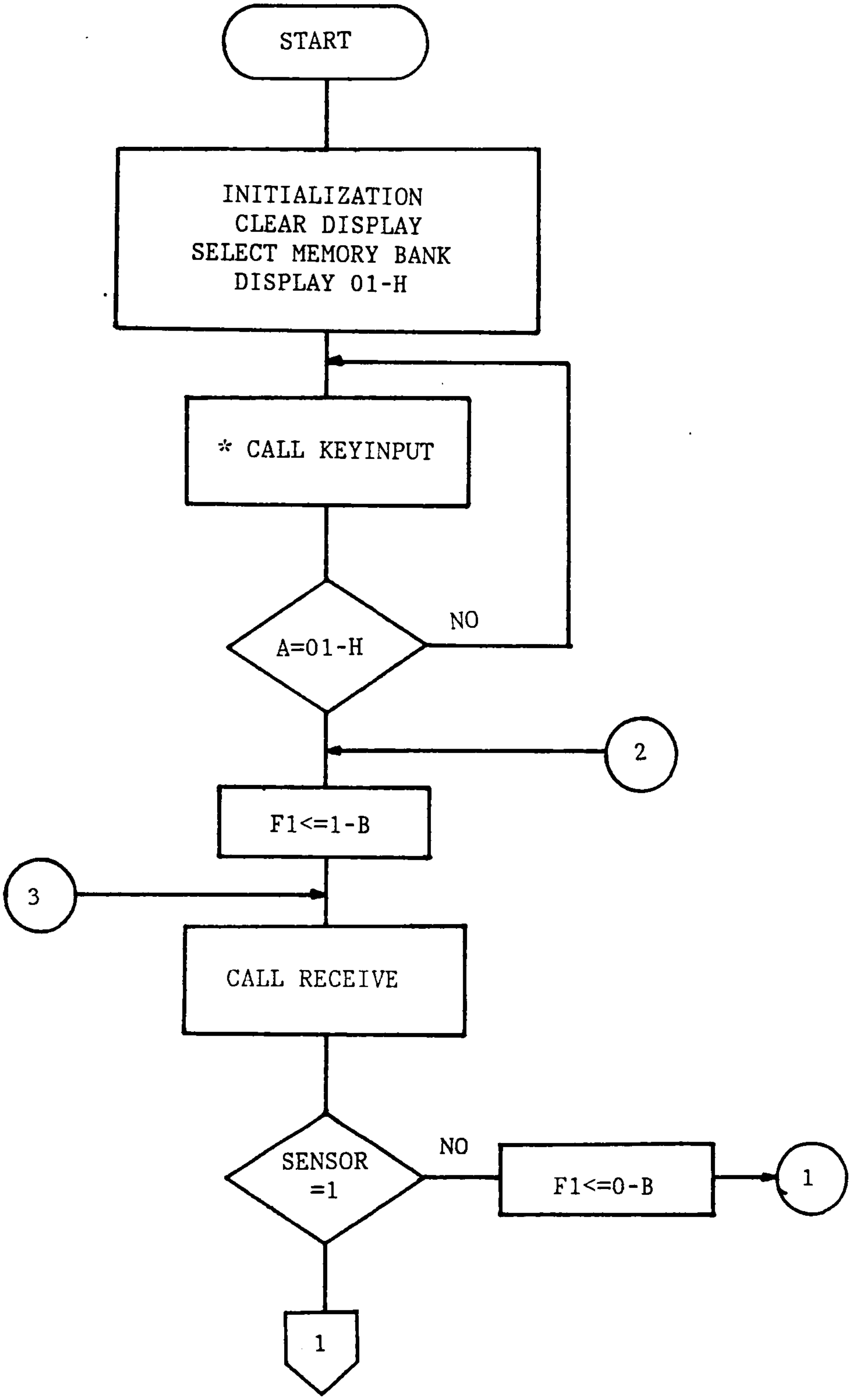


Figure 3-4. Slave Program Flow Chart.

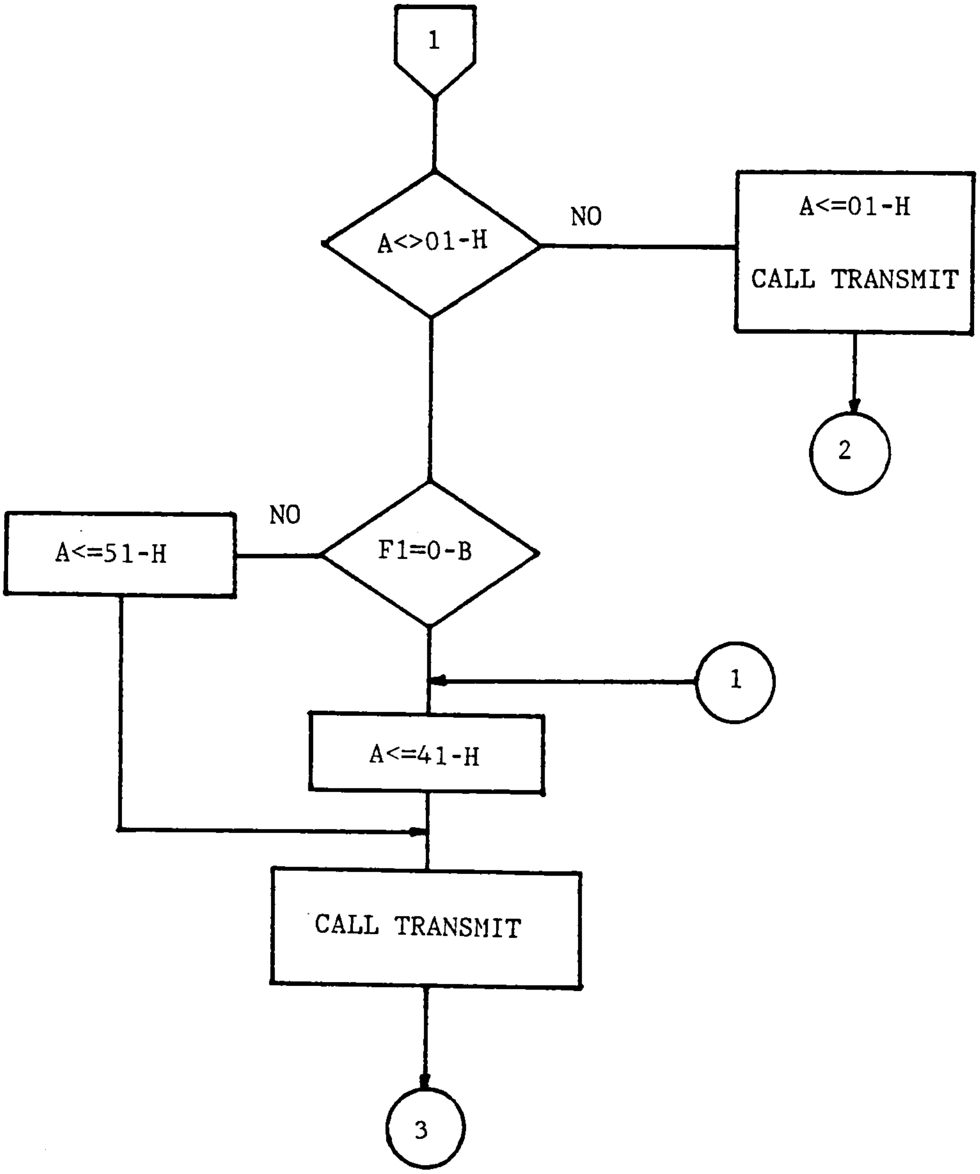


Figure 3-4. (Continued).

encoded message. Thus, the length of the encoded message is equal to seven.

Now, we know that the number of parity checks is three ("m"). These parity checks are given by:

$$H0 = D0 + D1 + D3 \tag{3-2}$$

$$H1 = D0 + D2 + D3 \tag{3-3}$$

$$H2 = D1 + D2 + D3 \tag{3-4}$$

where H0, H1, H2 are the parity checks, D0, D1, D2, D3 are the bits of the initial message and "+" is addition in the Galois field of order 2 or the Exclusive-Or function.

Therefore, our encoded message is

1	2	3	4	5	6	7
H0	H1	D0	H2	D1	D2	D3

where 1......7 are the relative positions of the bits.

Now, let us illustrate how to find the parity checks:

Suppose the initial message is 1011, where D0=1, D1=0, D2=1 and D3=1. If we want to compute the values of the parity checks (H0, H1, H2), then we need to plug the initial message values into the parity checks equations (3-2), (3-3), (3-4)

$$H0 = 1 + 0 + 1 = 0$$

$$H1 = 1 + 1 + 1 = 1$$

$$H2 = 0 + 1 + 1 = 0$$

giving us as a result the following encoded message:

1	2	3	4	5	6	7
0	1	1	0	0	1	1

At this point, we know how to find the necessary parity checks to correct a single error, but we need to find the necessary parity check in order to detect double errors. This parity check H3 watches over the whole encoded message; thus , H3 is given by:

$$H3 = H0 + H1 + D0 + H2 + D1 + D2 + D3 \tag{3-5}$$

Similarly, another equation of H3 is derived from substituting H0, H1, H2 in their own equations, leaving us with:

$$H3 = D0 + D1 + D2 \tag{3-6}$$

Thus, the final encoded message is:

1	2	3	4	5	6	7	8
H0	H1	D0	H2	D1	D2	D3	H3

When encoding, it is not too difficult to show that if the bits of the initial message are four, then the number of different initial messages is 16. Since this number is not very big, it is possible to generate a table with all the final encoded messages, which are also 16 (Table 3-1).

Observing this table, the relative positions of the final encoded message are the same, but they are in different places. This is done in order to design a simple decoder routine.

1	2	3	4	5	6	7	8		1	3	7	6	5	2	4	8
H0	H1	D0	H2	D1	D2	D3	H3		H0	D0	D3	D2	D1	H1	H2	H3

At this point, we can see that the process of encoding is reduced to a small table. This table is the encoder routine that the microcomputer uses.

Since the final encode message is created and transmitted, the decoder process at the receiver end begins. For a single error correction code, the process of decoding is not very difficult. The decoding process uses the syndrome parity checks (Px) in order to determine the relative position of the error (the syndrome). The syndrome parity checks are given by:

$$P0 = H0 + D0 + D1 + D3 \qquad (3\text{-}7)$$

$$P1 = H1 + D0 + D2 + D3 \qquad (3\text{-}8)$$

$$P2 = H2 + D1 + D2 + D3 \qquad (3\text{-}9)$$

Table 3-1

ENCODED MESSAGE

INITIAL MESSAGE	1				2					
	RELATIVE POSITION									
	1	3	7	6	5	2	4	8	Hex.	
Hex. No	H0	D0	D3	D2	D1	H1	H2	H3	1	2
0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	1	C	5
2	1	0	0	0	1	0	1	1	8	B
3	0	1	0	0	1	1	1	0	4	E
4	0	0	0	1	0	1	1	1	1	7
5	1	1	0	1	0	0	1	0	D	2
6	1	0	0	1	1	1	0	0	9	C
7	0	1	0	1	1	0	0	1	5	9
8	1	0	1	0	0	1	1	0	A	6
9	0	1	1	0	0	0	1	1	6	3
A	0	0	1	0	1	1	0	1	2	D
B	1	1	1	0	1	0	0	0	E	8
C	1	0	1	1	0	0	0	1	B	1
D	0	1	1	1	0	1	0	0	7	4
E	0	0	1	1	1	0	1	0	3	A
F	1	1	1	1	1	1	1	1	F	F
	A7	A6	A5	A4	A3	A2	A1	A0		

If no error occurs P0=P1=P2=0; if an error occurs, then the relative position of the error (the syndrome) is the decimal representation of the binary number (P2 P1 P0), where P0 is the less significant bit.

To illustrate this function, suppose that the transmited message is

1	2	3	4	5	6	7
0	1	1	0	0	1	1

but the received message is

1	2	3	4	5	6	7
0	0	1	0	0	1	1

Plugging the values of the received message into the syndrome parity check equations (3-7), (3-8), (3-9)

$$P0 = 0 + 1 + 0 + 1 = 0$$

$$P1 = 0 + 1 + 1 + 1 = 1$$

$$P2 = 0 + 0 + 1 + 1 = 0$$

We obtain from P2=0, P1=1, P0=0 the decimal number 2, which represents the relative position of the error (the syndrome).

For a double error-detection code, the decoding process uses one more syndrome parity check P3, that looks over the whole received message. This is found by:

$$P3 = H0 + H1 + D0 + H2 + D1 + D2 + D3 + H3 \qquad (3\text{-}10)$$

Knowing P3, the system can determine whether a simple error or a double error is detected:

i. If a simple error is detected, then it must be corrected.

ii. If a double error is detected, then it must indicate its occurence.

The procedure to determine when to correct a single error or when to indicate a double error, is shown in table 3-2.

Table 3-2

ERRORS

Situation If (P2, P1, P0) and	P3	Conclusion	Action
0	0	no error	Do nothing
0	1	error in H3	Do nothing
Syndrome	1	one error	Correct the error
Syndrome	0	two error	Indicate two errors

Now, a good question is how will the microcomputer do this? At the receiver, the transmitted message is stored in the accumulator as follows:

1	3	7	6	5	2	4	8	(Relative position)
H0	D0	D3	D2	D1	H1	H2	H3	
A7	A6	A5	A4	A3	A2	A1	A0	(Accumulator bits)

Next, the values of the syndrome parity checks P3, P2, P1, P0 are found using the procedure that is shown in figure (3-5). This procedure is used by the Decode-I Routine (Fig 3-6) in order to find the syndrome parity checks.

With the values of P3, P2, P1, P0 we can determine if the error is single or double:

i. If the error is double, then the microcomputer sets the register four (R4) different to zero (R4 in the register bank zero).

ii. If the error is single, then it is corrected using the table 3-3.

Our system determines the kind of error using the Decode-II Routine (Fig 3-7) which uses the procedure that is in the table 3-2, corrects the error using the Corword Routine (Fig 3-8(a)) which uses the table 3-3, and restores the initial message using the Fix-D Routine (Fig 3-8(b))

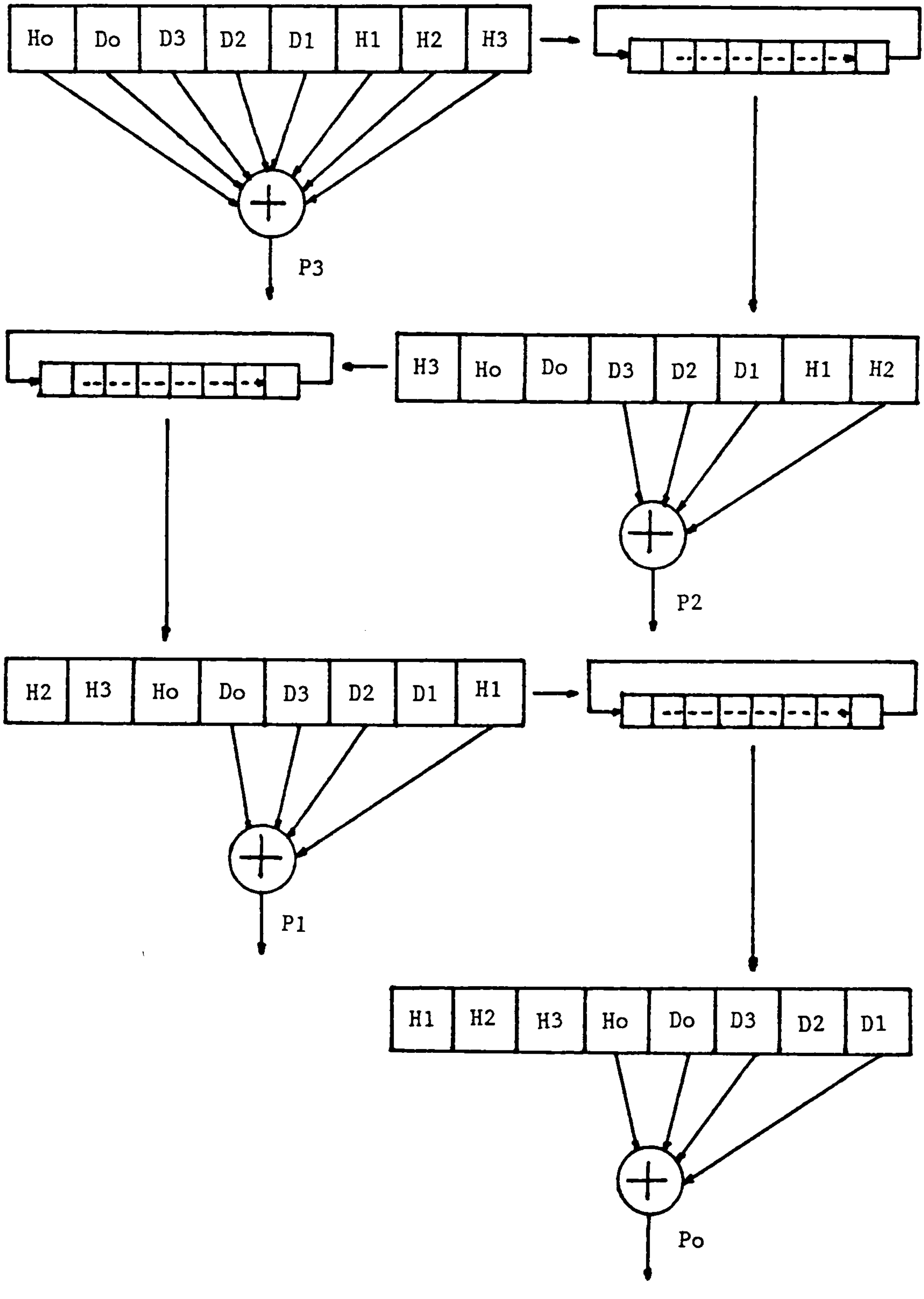


Figure 3-5. Procedure to determine the syndrome parity checks.

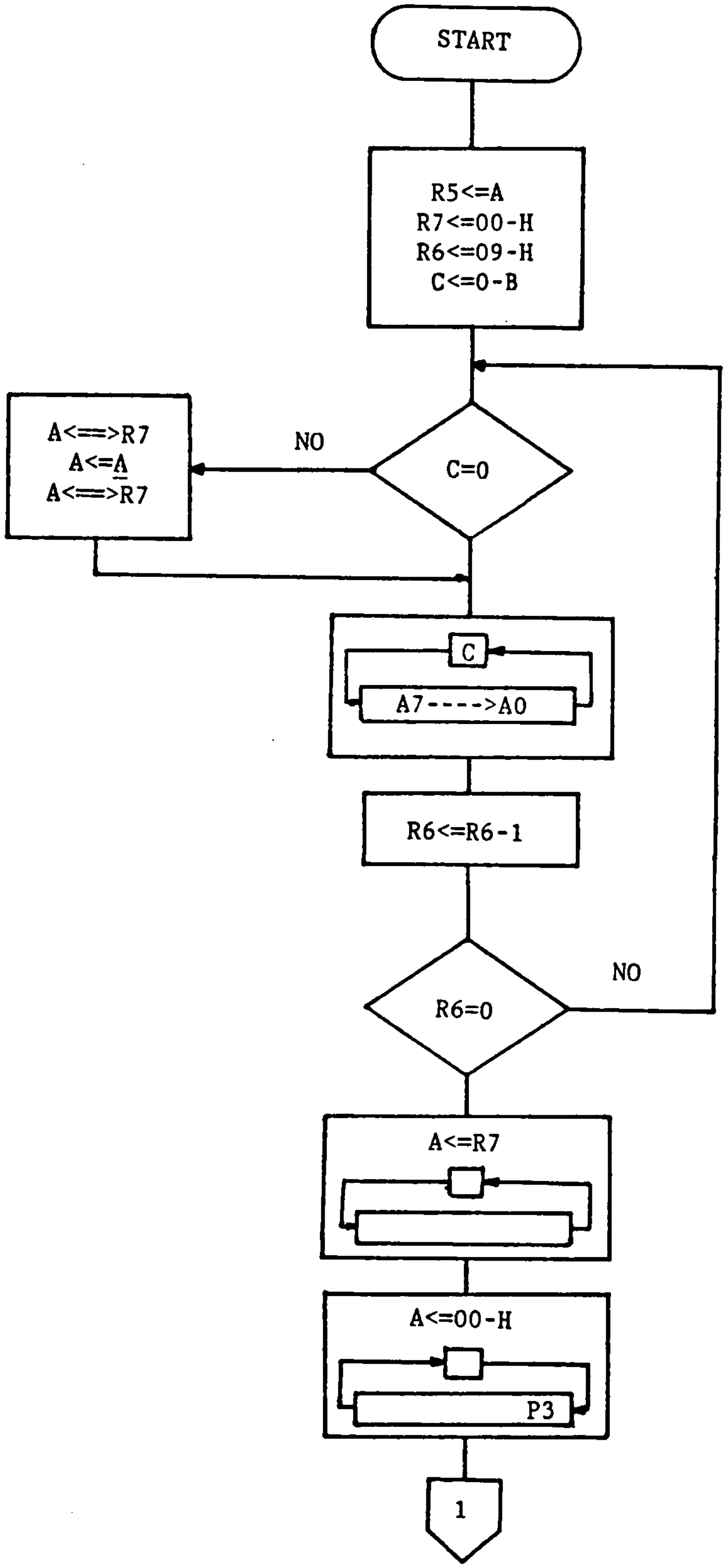


Figure 3-6. Decode-I Routine Flow Chart.

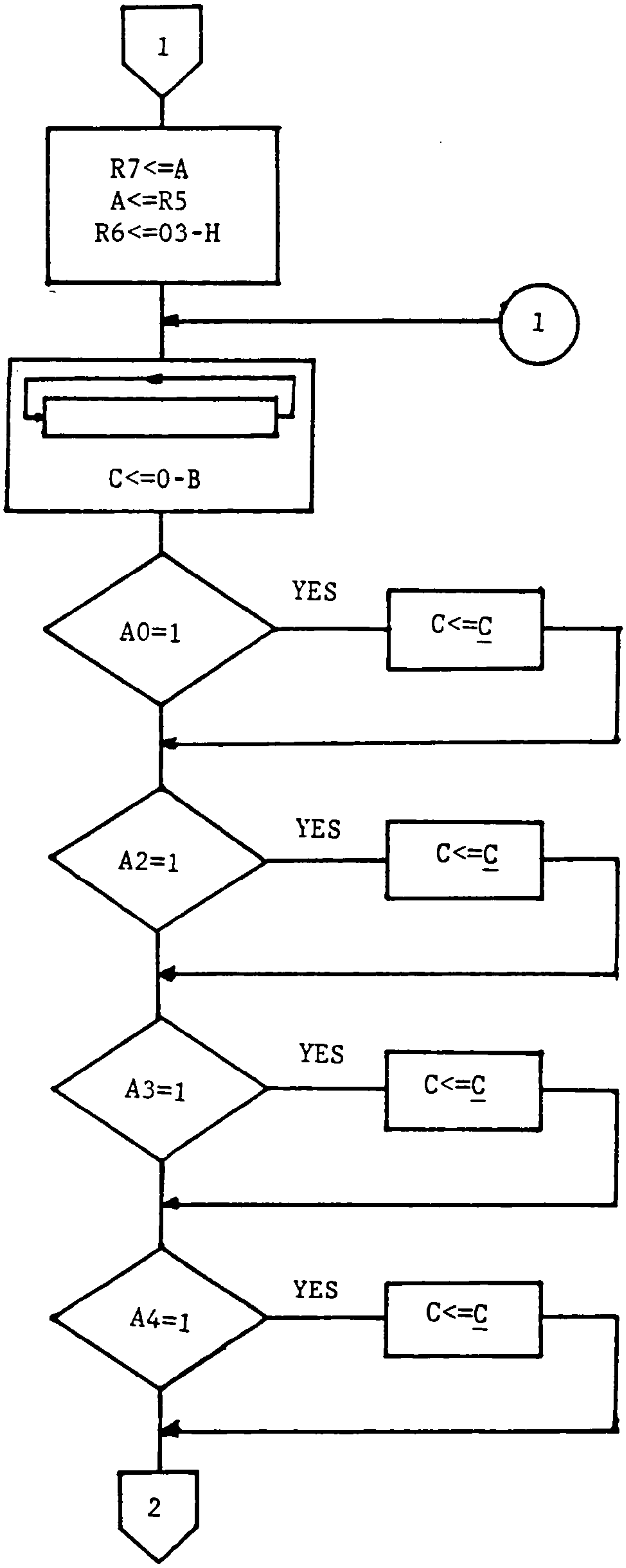


Figure 3-6. (Continued).

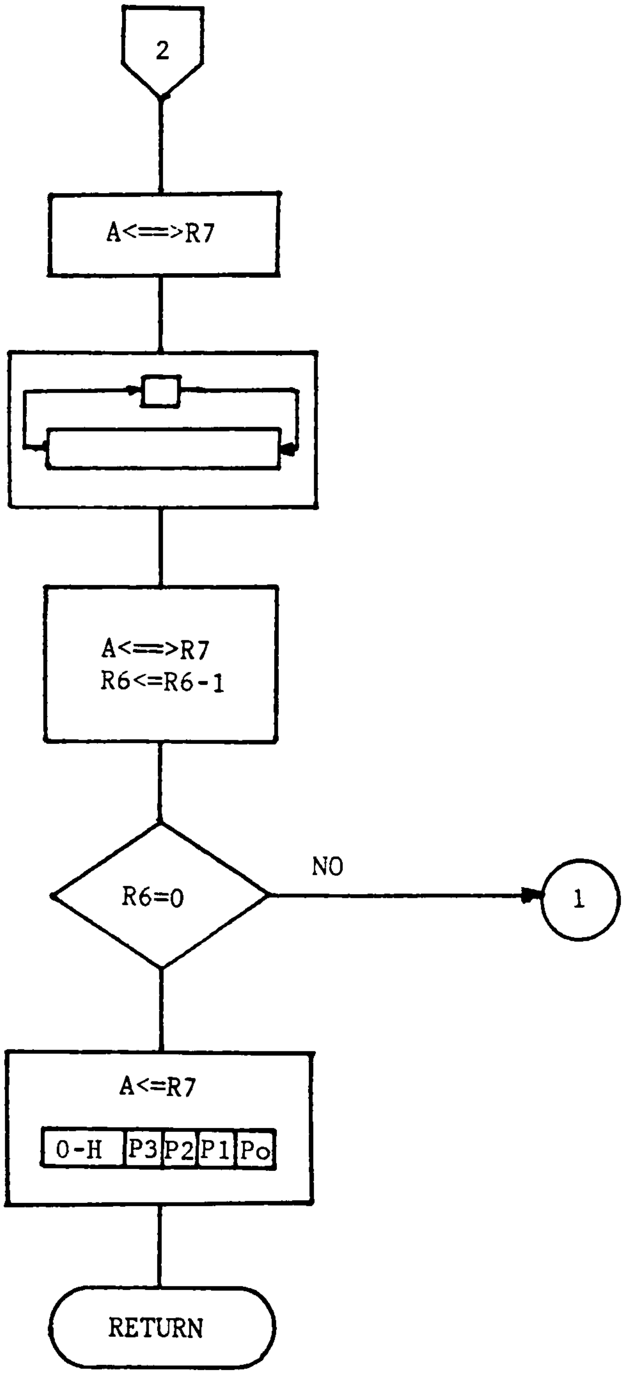


Figure 3-6. (Continued).

Table 3-3

CORRECTOR

Syndrome	Bit to Complement	Corrected Words
1	A7	80-H
2	A2	04-H
3	A6	40-H
4	A1	02-H
5	A3	08-H
6	A4	10-H
7	A5	20-H

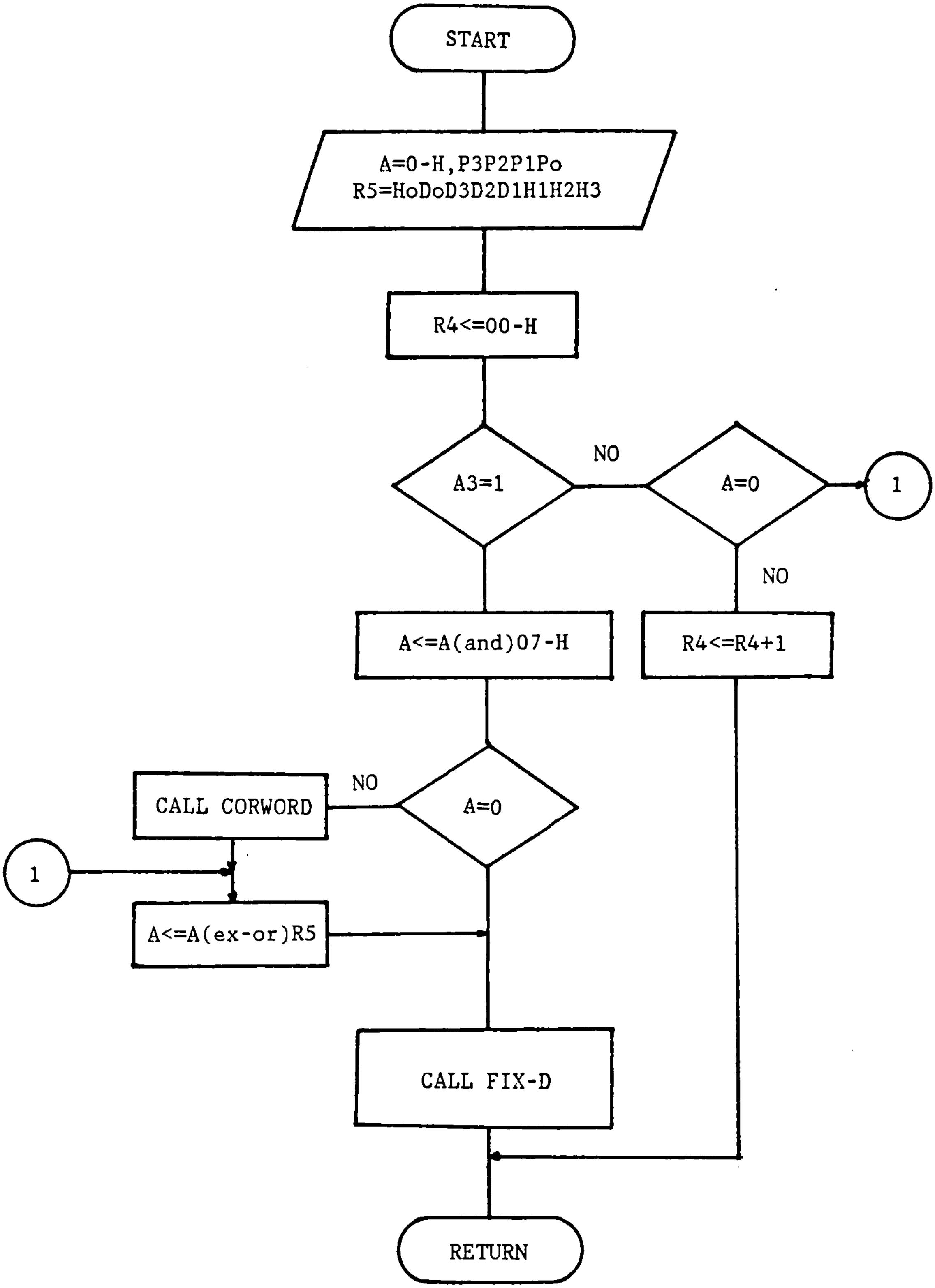


Figure 3-7. Decode-II Routine Flow Chart.

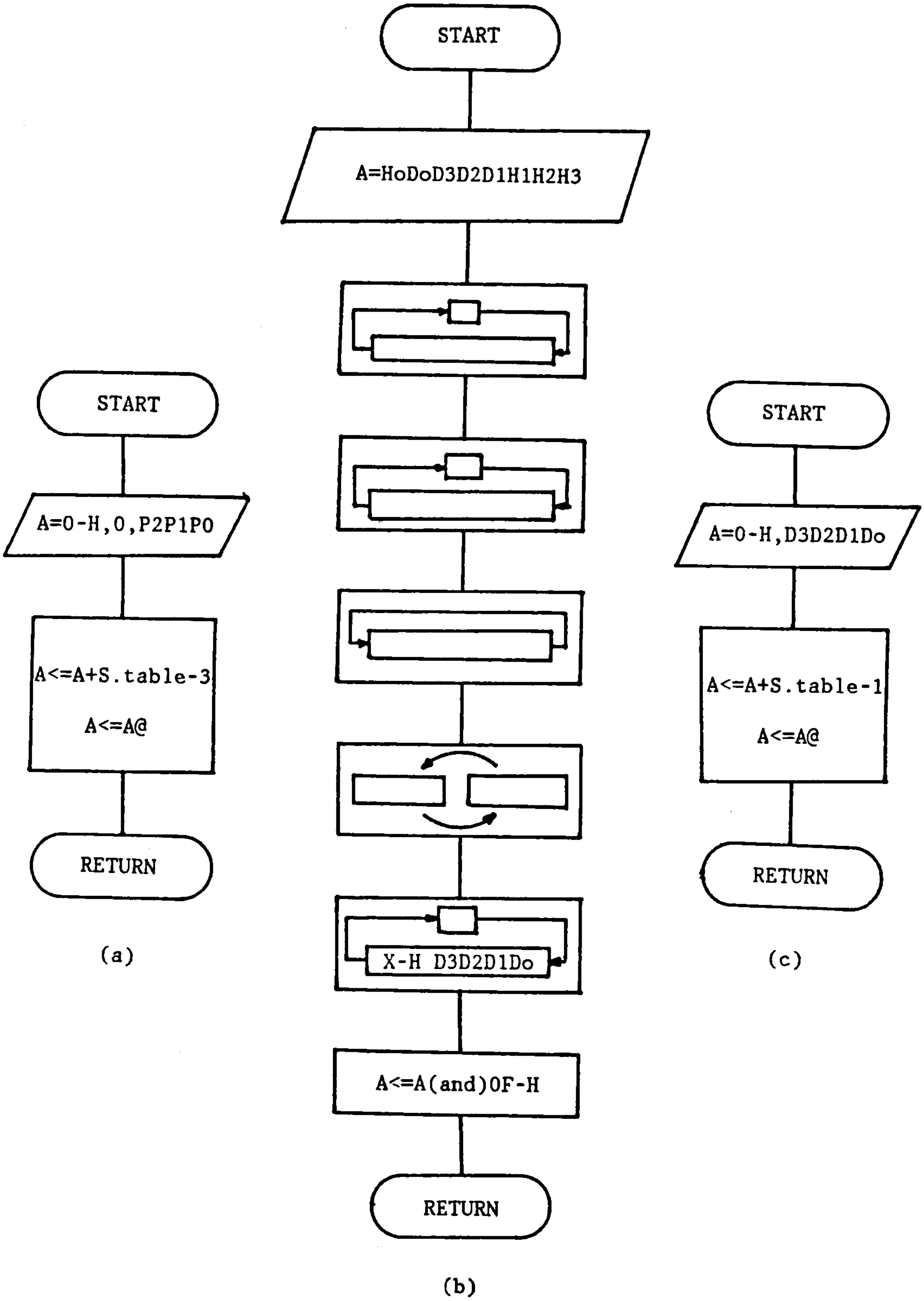


Figure 3-8. Flow charts of a) Corword routine, b) Fix-d routine, c) Encode Routine.

As we have shown, the transmitted message is composed of four hexadecimal characters instead of two. The Transmit Routine (Fig 3-9) transmits the four hexadecimal characters using the procedure that is shown in the figure 3-10, and the Receive Routine (Fig 3-11) receives the transmited message using the procedure that is shown in figure 3-12.

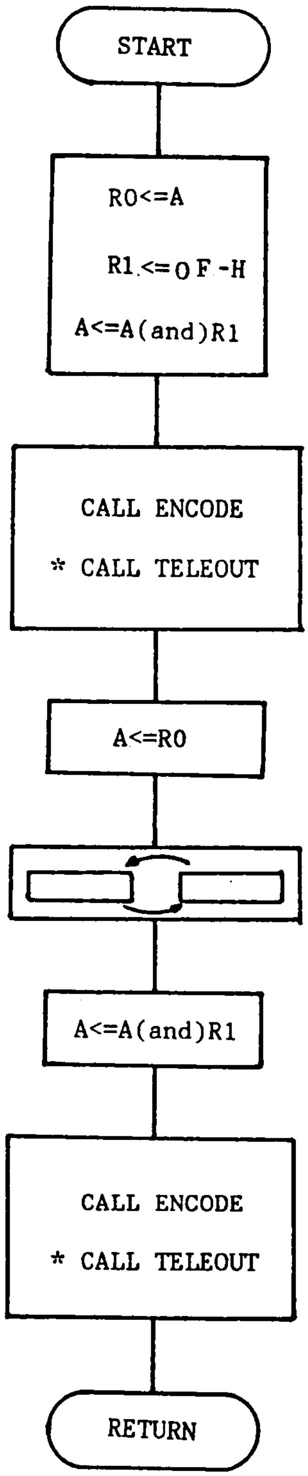


Figure 3-9. Transmit Routine Flow Chart.

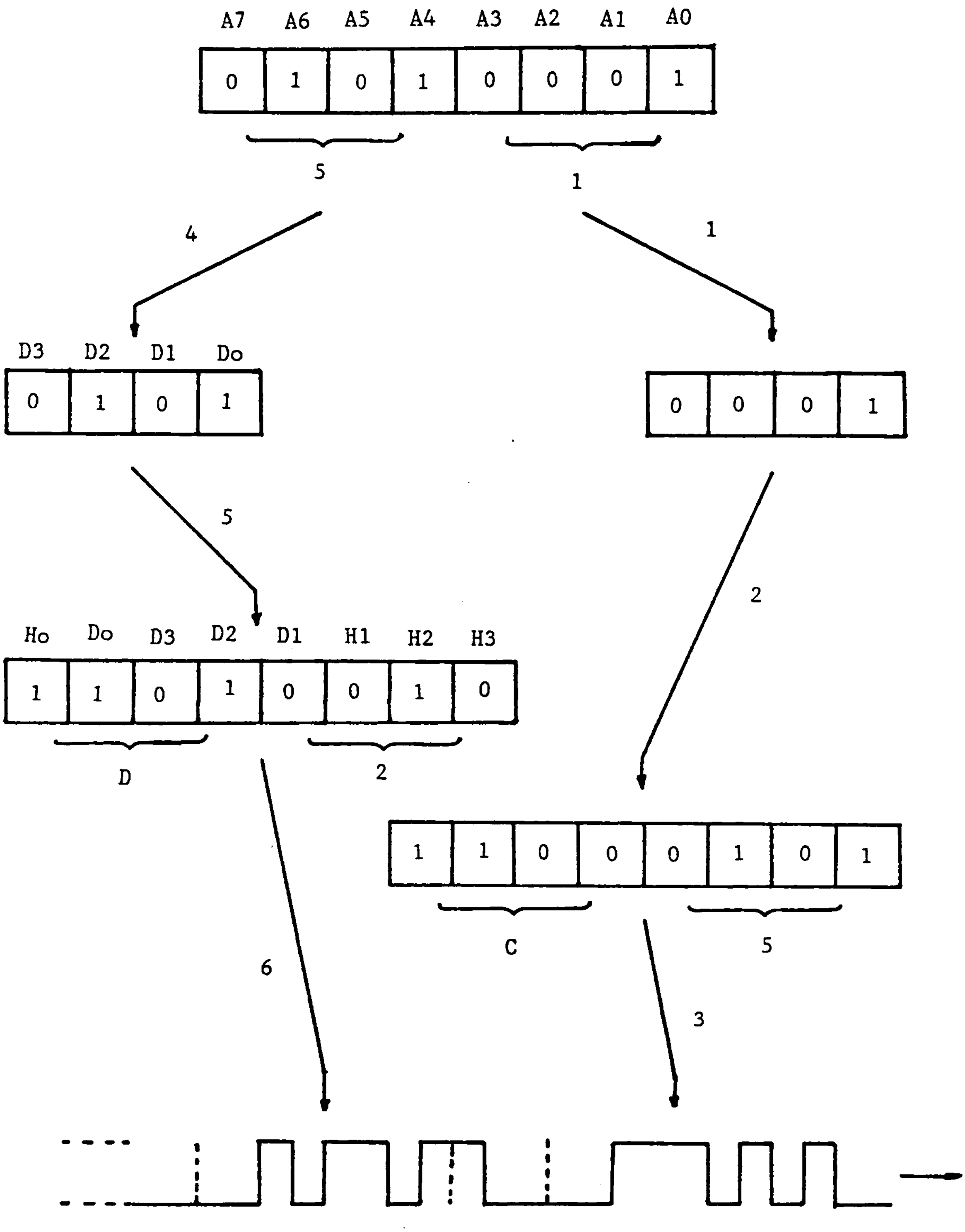


Figure 3-10. Procedure to transmit the encoded message.

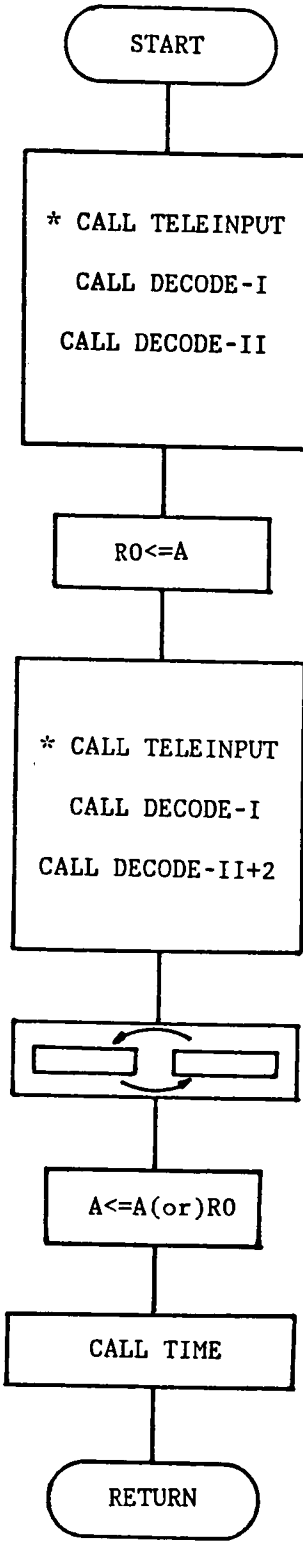


Figure 3-11. Receive Routine Flow Chart.

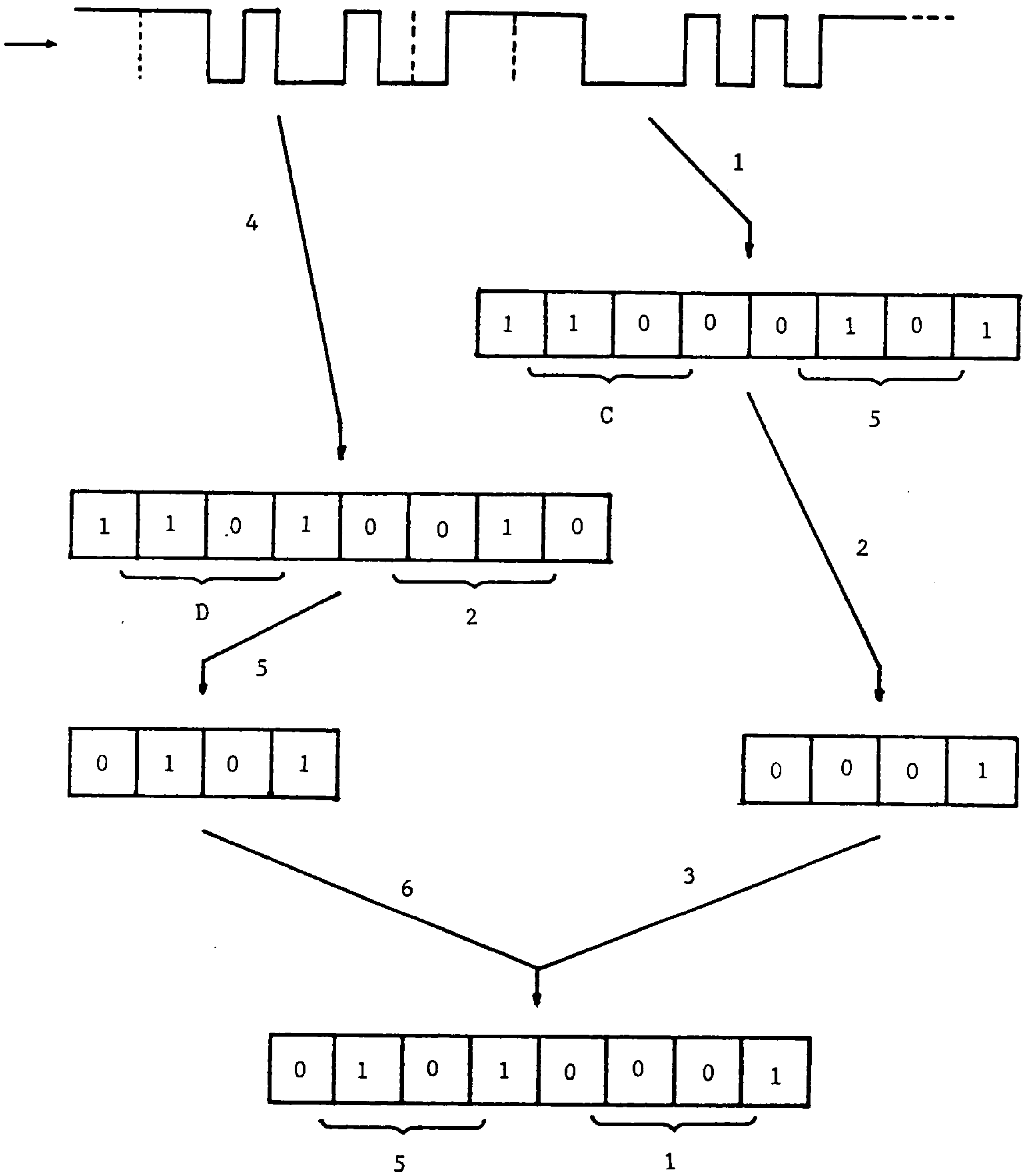


Figure 3-12. Procedure to receive the encoded message.

CHAPTER 4

CONCLUSIONS

This project illustrated the different steps which may be employed in order to design a small microcomputer alarm system using the power line as a carrier. The use of the power line reduces the installation cost and increases the flexibility of the system. However, if the power-line path includes power transformers, attenuation of the carrier frequency occurs; as a consequence the transmitted signal may not be detected. Therefore, the system may not be used in some areas of the protected environment. One solution of this problem is the use of capacity by-pass around the transformers. Further analysis is required to understand the characteristics of this powerline communication channel. Another problem appears if the distance between the transmitter and the receiver is too large.

Other important aspect is the total cost of the system; initially the cost will be moderate or high, but with the steadily increasing power of microcomputers, coupled with steadily decreasing hardware costs, the price of this system will be available at a modest cost in only a few years.

LIST OF REFERENCES

LIST OF REFERENCES

1. MCS-48 Microcomputer User's Manual, Intel Corporation, 3065 Bowers Avenue, Santa Clara, C.A 95051, July 1978.

2. IMSAI 8048 Control Computer User Manual, IMSAI Manufacturing Corporation, 14860 Wicks Boulevard, San Leandro, C.A 94577, November 1977.

3. John D. Lenk, Handbook of Data Communication, Prentice-Hall, New Jersey, 1984.

4. H.L. Capron, Brian K. Williams, Computer and Data Processing, The Benjamin/Cummings Publishing Company, California, 1982.

5. Hewlett. Packard, Optoelectronics Applications Manual, Mc Graw-Hill Book Company, 1977.

6. Peyton Z. Peebles, Jr, Communication System Principles, Addison-Wesley Publishing Company, Massachusett, 1976.

7. Richard Deasington, A Practical Guide to Computer Communications and Networking, R.J. Deasington/Ellis Horwood Ltd. England, 1982.

8. Franklin F. Kuo, Computer-Communication Networks, Prentice-Hall, New Jersey, 1973.

9. Shu Lin, An Introduction to Error-Correcting Codes, Prentice-Hall, New Jersey, 1970.

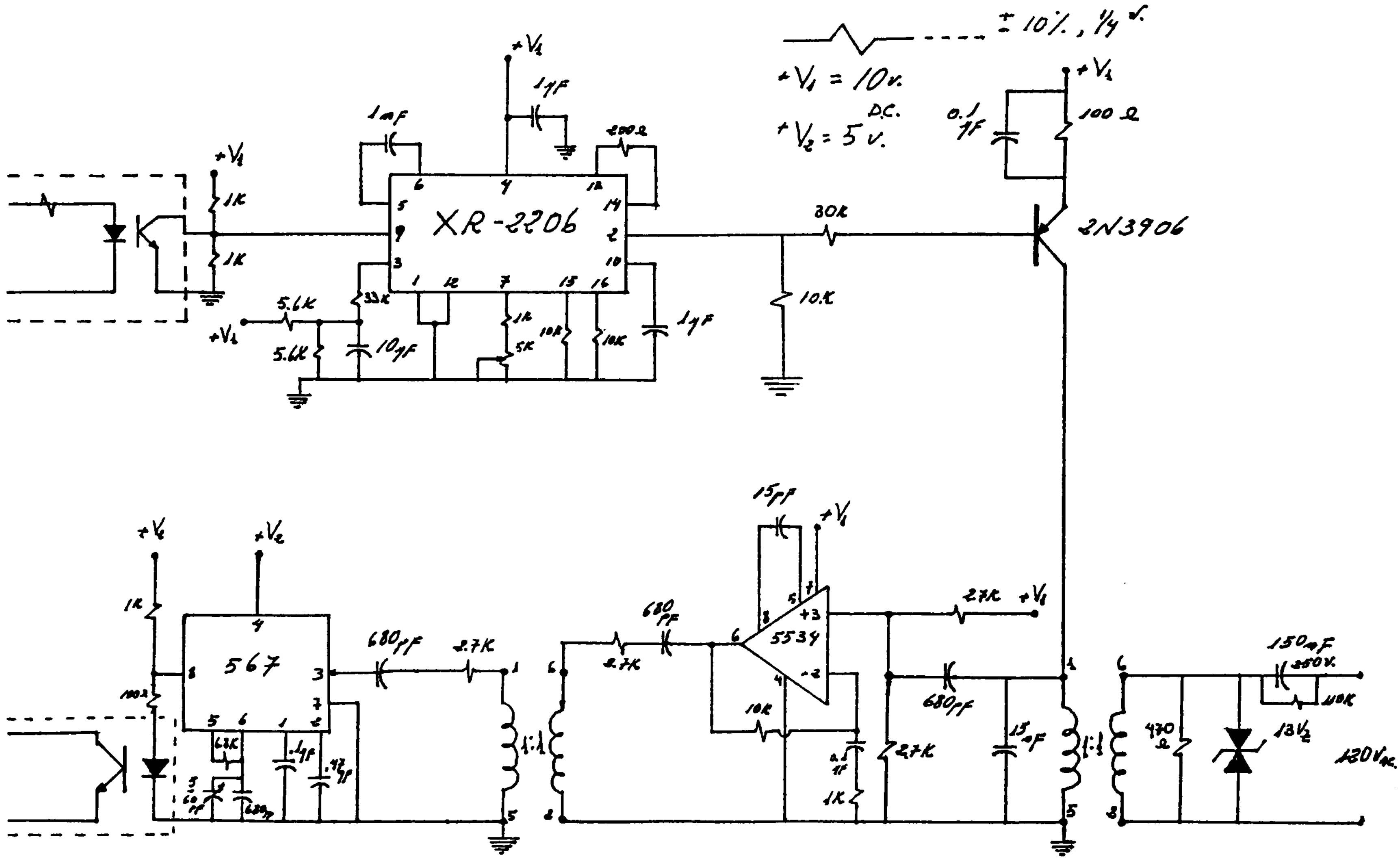
10. Roy A. Colclaser, Donal A. Neamen, Charles F. Hawkins, Electronic Circuit Analysis Principles, John Wiley & Sons, 1984.

11. George R. Davis, The Local Network Handbook, Data Communications Magazine, Mc Graw Hill, New York, New York, 1982.

12. Jack M. Nilles, Micros and Modems, Reston Publishing Company Inc, 1984.

APPENDICES

APPENDIX A



Final design of the power line modem.

APPENDIX B

SYSTEM SOFTWARE

```
LINE  ADDR   INST              SOURCE          COMMENTS
                       ;*********************************************
                       ;********        MASTER PROGRAM        ********
                       ;*********************************************
      0C00   F5                SEL MB1         ;SELECT MEMORY BANK 1
      0C01   F480              CALL CLEAR      ;CLEAR DISPLAY
      0C03   230F              MOV A,#0FH      ;
      0C05   F485              CALL DISP       ;DISPLAY OF-H
      0C07   F48D      AA:     CALL KEIP       ;STANBY STATE
      0C09   D30F              XOR A,#0FH      ;WAIT FOR THE START
      0C0B   9607              JNZ AA          ;SIGNAL
      0C0D   F480              CALL CLEAR      ;
      0C0F   2351      BB:     MOV A,#51H      ;
      0C11   F4A0      DD:     CALL TRANS      ;CALL TRANSMIT
      0C13   2300              MOV A,#00H      ;
      0C15   62                MOV T,A         ;WAIT FOR THE
      0C16   55                STRT T          ;TRANSMISSION
      0C17   0E        T3:     MOVD A,P6       ;OF THE SLAVE
      0C18   521C              JB2 T1          ;IF P62=1 GOTO T1
      0C1A   8420              JMP T2          ;
      0C1C   162B      T1:     JFT EE          ;
      0C1E   8417              JMP T3          ;
      0C20   F4B0      T2:     CALL RECEI      ;RECEIVE THE SLAVE'S
      0C22   A8                MOV R0,A        ;TRANSMISSION
      0C23   FC                MOV A,R4        ;
      0C24   960F              JNZ BB          ;DETERMINE THE STABLE
```

```
0C26    F8              MOV A,R0        ;STATE (INTRUDER OR
0C27    D341            XOR A,#41H      ;NORMAL STATE)
0C29    960F            JNZ BB          ;
0C2B    230A    EE:     MOV A,#0AH      ;
0C2D    F485            CALL DISP       ;START THE ALARM
0C2F    F48D    CC:     CALL KEIP       ;WAIT FOR THE
0C31    D30A            XOR A,#0AH      ;PASSWORD
0C33    962F            JNZ CC          ;
0C35    F480            CALL CLEAR      ;
0C37    2301            MOV A,#01H      ;INITIALIZE THE RESET
0C39    8411            JMP DD          ;CODE
```

```
ADDR   INST         SOURCE          COMMENTS
                    ;*********************************************
                    ;********        SLAVE PROGRAM        ********
                    ;*********************************************
0C00   F5                  SEL MB1         ;SELECT MEMORY BANK 1
0C01   F480                CALL CLEAR      ;CLEAR DISPLAY
0C03   2301                MOV A,#01H      ;
0C05   F485                CALL DISP       ;DISPLAY 01-H
0C07   F48D         AA:    CALL KEIP       ;STANBY STATE
0C09   D301                XOR A,#01H      ;WAIT FOR THE START
0C0B   9607                JNZ AA          ;SIGNAL
0C0D   F480                CALL CLEAR      ;
0C0F   A5           HH:    CLR F1          ;
0C10   B5                  CPL F1          ;
0C11   F4B0         FF:    CALL RECEI      ;CALL RECEIVE
0C13   4625                JNT1 BB         ;DETERMINE THE CONDITION
0C15   D301                XOR A,#01H      ;OF THE I/O-F SENSOR
0C17   C628                JZ CC           ;
0C19   761F                JF1 DD          ;
0C1B   2341         GG:    MOV A,#41H      ;
0C1D   8421                JMP EE          ;
0C1F   2351         DD:    MOV A,#51H      ;NORMAL STATE
0C21   F4A0         EE:    CALL TRANS      ;CALL TRANSMIT
0C23   8411                JMP FF          ;
0C25   A5           BB:    CLR F1          ;
0C26   841B                JMP GG          ;
```

```
0C28    2301    CC:    MOV A,#01H    ;RESET SIGNAL
0C2A    F4A0           CALL TRANS    ;CALL TRANSMIT
0C2C    840F           JMP HH        ;
```

```
LINE  ADDR    INST            SOURCE             COMMENTS
00100                 ;***********************************************
00200                 ;********        ENCODE ROUTINE          ********
00250                 ;********          TABLE 3-1             ********
00300                 ;***********************************************
00400 0F00    00      TABL1:   EN-#0             ;INITIAL MESSAGE 0
00500 0F01    C5               EN-#1             ;INITIAL MESSAGE 1
00600 0F02    8B               EN-#2             ;INITIAL MESSAGE 2
00700 0F03    4E               EN-#3             ;INITIAL MESSAGE 3
00800 0F04    17               EN-#4             ;INITIAL MESSAGE 4
00900 0F05    D2               EN-#5             ;INITIAL MESSAGE 5
01000 0F06    9C               EN-#6             ;INITIAL MESSAGE 6
01100 0F07    59               EN-#7             ;INITIAL MESSAGE 7
01200 0F08    A6               EN-#8             ;INITIAL MESSAGE 8
01300 0F09    63               EN-#9             ;INITIAL MESSAGE 9
01400 0F0A    2D               EN-#A             ;INITIAL MESSAGE A
01500 0F0B    E8               EN-#B             ;INITIAL MESSAGE B
01600 0F0C    B1               EN-#C             ;INITIAL MESSAGE C
01700 0F0D    74               EN-#D             ;INITIAL MESSAGE D
01800 0F0E    3A               EN-#E             ;INITIAL MESSAGE E
01900 0F0F    FF               EN-#F             ;INITIAL MESSAGE F
02000 0F10    AE      ENCO:    MOVP A,@A         ;ENTRY POINT
02100 0F11    83               RET               ;RETURN
02200                 ;***********************************************
02300                 ;********        DECODE-I ROUTINE        ********
02400                 ;***********************************************
```

```
0F20    AD      DECO1:  MOV R5,A        ;ENTRY POINT
0F21    BF00            MOV R7,#00H     ;
0F23    BE09            MOV R6,#09H     ;
0F25    97              CLR C           ;
0F26    E62B    BB:     JNC AA          ;DETERMINE THE
0F28    2F              XCH A,R7        ;SYNDROME
0F29    37              CPL A           ;PARITY CHECK
0F2A    2F              XCH A,R7        ;  (P3)
0F2B    67      AA:     RRC A           ;
0F2C    EE26            DJNZ R6,BB      ;
0F2E    FF              MOV A,R7        ;
0F2F    67              RRC A           ;
0F30    BF00            MOV A,#00H      ;
0F32    F7              RLC A           ;
0F33    AF              MOV R7,A        ;
0F34    FD              MOV A,R5        ;
0F35    BF03            MOV R6,#03H     ;
0F37    77      CC:     RRA             ;DETERMINE THE
0F38    97              CLR C           ;SYNDROME
0F39    1248            JB0 DD          ;PARITY CHECKS
0F3B    524B    HH:     JB2 EE          ; P2, P1, P0
0F3D    724E    II:     JB3 FF          ;
0F3F    9251    JJ:     JB4 GG          ;
0F41    2F      KK:     XCH A,R7        ;
0F42    F7              RLC A           ;
0F43    2F              XCH A,R7        ;
```

```
0F44   EE37            DJNZ R6,CC       ;
0F46   FF              MOV A,R7         ;
0F47   83              RET              ;RETURN
0F48   A7      DD:     CPLC             ;
0F49   E43B            JMP HH           ;
0F4B   A7      EE:     CPLC             ;
0F4C   E43D            JMP II           ;
0F4E   A7      FF:     CPLC             ;
0F4F   E43F            JMP JJ           ;
0F51   A7      GG:     CPLC             ;
0F52   E441            JMP KK           ;
               ;**********************************************
               ;********        CORWORD ROUTINE        ********
               ;********          TABLE 3-3            ********
               ;**********************************************
0F54   00      INIT3:  S0               ;NEVER OCCUR
0F55   80              S1               ;COMPLEMENT A7
0F56   04              S2               ;COMPLEMENT A2
0F57   40              S3               ;COMPLEMENT A6
0F58   02              S4               ;COMPLEMENT A1
0F59   08              S5               ;COMPLEMENT A3
0F5A   10              S6               ;COMPLEMENT A4
0F5B   20              S7               ;COMPLEMENT A5
0F5C   0354    CORW:   ADD A,#INIT3     ;ENTRY POINT
0F5E   A3              MOVP A,@A        ;
0F5F   83              RET              ;RETURN
```

```
                    ;*********************************************
                    ;********       DECODE-II ROUTINE     ********
                    ;********           TABLE 3-2         ********
                    ;*********************************************
0F60    BC00    DECO2:  MOV R4,#00H     ;ENTRY POINT
0F62    7268            JB3 AA          ;
0F64    C66E            JZ BB           ;
0F66    1C              INC R4          ;
0F67    83              RET             ;RETURN
0F68    5307    AA:     ANL A,#07H      ;
0F6A    C66E            JZ BB           ;
0F6C    F45C            CALL CORW       ;CALL CORWORD
0F6E    DD      BB:     XRL A,R5        ;
0F6F    F472            CALL FIX-D      ;
0F71    83              RET             ;RETURN
                    ;*********************************************
                    ;********        FIX-D ROUTINE        ********
                    ;*********************************************
0F72    F7      FIX-D:  RLCA            ;ENTRY POINT
0F73    F7              RLCA            ;
0F74    77              RAA             ;
0F75    47              SWAP A          ;
0F76    F7              RLCA            ;
0F77    530F            ANL A,#0FH      ;
0F79    83              RET             ;RETURN
```

```
                   ;*******************************************
                   ;********      CLEAR ROUTINE        ********
                   ;*******************************************
0F80    E5         CLEAR:  SEL MB0         ;ENTRY POINT
0F81    5413               CALL *CLEAR     ;CALL MONITOR ROUTINE
0F83    F5                 SELL MB1        ;
0F84    83                 RET             ;RETURN
                   ;*******************************************
                   ;********     DISPLAY ROUTINE       ********
                   ;*******************************************
0F85    E5         DISP:   SEL MB0         ;ENTRY POINT
0F86    5462               CALL *DGS       ;CALL MONITOR ROUTINE
0F88    541F               CALL *OUTDSP    ;CALL MONITOR ROUTINE
0F8A    F5                 SEL MB1         ;
0F8B    FF                 MOV A,R7        ;
0F8C    83                 RET             ;RETURN
                   ;*******************************************
                   ;********    KEYINPUT ROUTINE       ********
                   ;*******************************************
0F8D    E5         KEIP:   SEL MB0         ;ENTRY POINT
0F8E    5434               CALL *KEYINP    ;CALL MONITOR ROUTINE
0F90    F5                 MOV R7,A        ;
0F91    AF                 SEL MB1         ;
0F92    83                 RET             ;RETURN
```

```
                         ;*********************************************
                         ;********      TELEOUT ROUTINE        ********
                         ;*********************************************
0F93    55      TELO:    STRT T            ;ENTRY POINT
0F94    E5               SEL MB0           ;
0F95    744B             CALL *TELEOUT     ;CALL MONITOR ROUTINE
0F97    F5               SEL MB1           ;
0F98    83               RET               ;RETURN
                         ;*********************************************
                         ;********      TELEINP ROUTINE        ********
                         ;*********************************************
0F99    55      TEINP:   STRT TIMER        ;ENTRY POINT
0F9A    E5               SEL MB0           ;
0F9B    7469             CALL *TELEIN      ;CALL MONITOR ROUTINE
0F9D    F5               SEL MB1           ;
0F9E    83               RET               ;RETURN
                         ,*********************************************
                         ;********      TRANSMIT ROUTINE       ********
                         ;********       FIGURE 3-8(A)         ********
                         ;*********************************************
0FA0    A8      TRANS:   MOV R0,A          ;ENTRY POINT
0FA1    B90F             MOV R1,#0FH       ;
0FA3    59               ANL A,R1          ;STEP 1
0FA4    F410             CALL ENCO         ;CALL ENCODE (STEP 2)
0FA6    F493             CALL TELO         ;CALL TELEOUT (STEP 3)
```

```
14800 0FA8    F8                MOV A,R0          ;
14900 0FA9    47                SWAP A            ;
15000 0FAA    59                ANL A,R1          ;STEP 4
15100 0FAB    F410              CALL ENCO         ;CALL ENCODE (STEP 5)
15200 0FAD    F493              CALL TELO         ;CALL TELEOUT (STEP 6)
15300 0FAF    83                RET               ;RETURN
15400                   ;********************************************
15500                   ;********        RECEIVE ROUTINE        ********
15550                   ;********         FIGURE 3-8(B)         ********
15600                   ;********************************************
15700 0FB0    F499      RECEI:  CALL TEINP        ;ENTRY POINT (STEP 1)
15800 0FB2    F420              CALL DECO1        ;CALL DECODE-I (STEP 2)
15900 0FB4    F460              CALL DECO2        ;CALL DECODE-II (STEP 2)
16000 0FB6    A8                MOV R0,A          ;STEP 3
16100 0FB7    F499              CALL TEINP        ;CALL TELEINP (STEP 4)
16200 0FB9    F420              CALL DECO1        ;CALL DECODE-I (STEP 5)
16300 0FBB    F462              CALL DECO2+2      ;CALL DECODE-II+2 (ST 5)
16400 0FBD    47                SWAP A            ;STEP 6
16500 0FBE    48                ORL A,R0          ;STEP 6
16600 0FBF    F4C2              CALL TIME         ;
16700 0FC1    83                RET               ;RETURN
16800                   ;********************************************
16900                   ;********        TIME ROUTINE           ********
17000                   ;********************************************
17100 0FC2    A8        TIME:   MOV R0,A          ;ENTRY POINT
17200 0FC3    23BC              MOV A,#BCH        ;
```

```
0FC5    62              MOV T,A         ;WAIT FOR THE
0FC6    55              STRT T          ;END
0FC7    16CB    BB:     JTF AA          ;OF THE TRANSMISSION
0FC9    E4C7            JMP BB          ;
0FCB    F8      AA:     MOV A,R0        ;
0FCC    83              RET             ;RETURN
```

TELETYPE MONITOR ROUTINES

```
ADDR   INST      SOURCE                      COMMENTS
                 ;**********************************************
                 ;******* TELETYPE OUTPUT DRIVER 110 BAUD *******
                 ;**********************************************
034B   AE        TELEOUT: MOV R6,A           ;SAVE
034C   BA0B               MOV R2,#11         ;BIT COUNT
034E   97                 CLR C              ;FOR START BIT
034F   1653      TELE1:   JTF TELE2          ;
0351   644F               JMP TELE1          ;WAIT FOR TIMEOUT
0353   E65A      TELE2:   JNC BITOUT1        ;OUTPUT 0 BIT
0355   230B               MOV A,#0BH         ;OUTPUT INVERTED 1 BIT
0357   9F                 ANLD P7,A          ;PORT 7 BIT 2
0358   645D               JMP BITOUT3        ;
035A   2304      BITOUT1: MOV A,#4           ;
035C   8F                 ORLD P7,A          ;
035D   97        BITOUT3: CLR C              ;
035E   A7                 CLP C              ;FOR STOP BITS
035F   FE                 MOV A,R6           ;
0360   67                 RRC A              ;
0361   AE                 MOV R6,A           ;
0362   23BD               MOV A,#-67         ;FOR 9040 USEC DELAY
0364   7449               CALL TIMEND        ;SET TIMER AND START
0366   EA4F               DJNZ R2,TELE1      ;GET NEXT BIT IF ANY
0368   83                 RET                ;RETURN
```

```
                    ;*********************************************
                    ;******* TELETYPE INPUT ROUTINE 110 BAUD *******
                    ;*********************************************
0369    BA08        TELEIN: MOV R2,#8           ;8 BITS/BYTE
036B    0E          T1:     MOVD A,P6           ;READ LINE
036B    526B                JB2 T1              ;
036E    23DE                MOV A,#-34          ;HALF BIT TIME
0370    7445                CALL TIMEOUT+2      ;START CLOCK
0372    23BD                MOV A,#-67          ;
0374    7445                CALL TIMEOUT+2      ;
0376    0E                  MOVD A,P6           ;READ LINE
0377    526B                JB2 T1              ;FALSE START BITE
0379    23BD        T3:     MOV A,#-67          ;
037B    7445                CALL TIMEOUT+2      ;
037D    0E                  MOVD A,P6           ;
037E    67                  RRC A               ;
037F    67                  RRC A               ;
0380    67                  RRC A               ;BIT IN CARRY
0381    FE                  MOV A,R6            ;
0382    67                  RRC A               ;MERGE NEW BIT
0383    AE                  MOV R6,A            ;SAVE
0384    EA79                DJNZ R2,T3          ;GET NEXT BIT
0386    7443                CALL TIMEOUT        ;WAIT FOR STOP BIT
0388    FE                  MOV A,R6            ;RETRIEVE BYTE
```

```
0389   83         RET                  ;RETURN
                  ;************************************************
                  ;********       AUXILIAR ROUTINES       ********
                  ;************************************************
0343   23FE  TIMEOUT: MOV A,#-2        ;
0345   1649           JTF TIMEND       ;
0347   6445           JMP TIMEOUT+2    ;
0349   44F4  TIMEND:  JMP TEND1        ;
                  ;
                  ;***********************************************
                  ;
02F4   62    TEND1:   MOV T,A          ;ENTRY POINT
02F5   00             NOP              ;
02F6   00             NOP              ;
02F7   00             NOP              ;
02F8   00             NOP              ;
02F9   55             STRT T           ;
02FA   2308           MOV A,#8         ;
02FC   83             RET              ;RETURN
                  ;
                  ;***********************************************
```

VITA

Marcos Jesus Dominguez Santana is a Venezuelan although he was born in the Canary Islands, Spain on September 7, 1956. He attended elementary school in Guacara, Carabobo, Venezuela and was graduated from the high school of Enrique Delgado Palacios in July 1973. In January of 1974, he entered the University of Carabobo, Carabobo, Venezuela and in July 1980 he received a bachelor of Science Degree in Electrical Engineering. After graduation, he worked for one and a half years in the company RO-MI a subsidiary of the Allen Bradley Company in Venezuela. In January 1982 he came to the United States under a scholarship from the Venezuelan Goverment represented by the Fundacion Gran Mariscal de Ayacucho. In January 1983 he was accepted at the University of Tennessee, Knoxville, where he began his study toward a Master's Degree. His Master's Degree was awarded in March 1985.

His graduate work has a special interest in the area of Microcomputer Alarm Systems.