MICROPROCESSOR BASED FREQUENCY GENERATOR FOR RPM METER TESTING

PROJECT REPORT

Submitted by

V.M. KRISHNAMOORTHY
K.KAVITHA
A. PRADEEPA
T.SUDHAGAR

Guided by

Dr. K.A.PALANISWAMY, B.E.,
M.sc(Engg), Ph.D, FIE, MISTE
Professor & Head of EEE (Dept)

1998 - 99

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OFTHE DEGREE OF
BACHELOR OF ENGINEERING
IN ELECTRICAL & ELECTRONICS ENGINEERING
OF THE BHARATHIAR UNIVERSITY

Department of Electrical & Electronics Engineering
Kumaraguru College of Technology
Coimbatore - 641 006

Department of Electrical and Electronics Engineering

Kumaraguru College of Technology

Coimbatore – 641 006

Certificate

This is to certify that the report entitled

Microprocessor based Frequency Generator For RPM Meter Testing

has been submitted by

Mr. V. M. Krishnamoorthy
Ms. K. Kavitha
Ms. A. Pradeepa
Mr. T. Sudhagar

in partial fulfillment for the award of Bachelor of Engineering in Electrical and Electrical Engineering Branch of Bharathiar University, Coimbatore – 641 046 During the academic year 1998 – '99

Guide

Date : 17/3/99

Head of the Department

PROFESSOR & HEAD
Dept. of Electrical & Electronics Engg
Kumaraguru College of Tech., Coimbatore.

Certified that the candidate with University Registration No. ____________

was examined in project viva – voce on 17·03·1999·

Internal Examiner

External Examiner

PREMIER INSTRUMENTS & CONTROLS LIMITED

pricol

P.B. No. 4209, PERIANAICKENPALAYAM
COIMBATORE - 641 020. INDIA
PHONE : 0422 - 892901 to 10
FAX : 0422 - 892028.
TELEX : 0855 - 8341 PIC IN.
CABLE : DASHBOARD

March 12, 1999

TO WHOM IT MAY CONCERN:

This is to certify that the following B.E. VIII semester Electrical and Electronics branch students of Kumaraguru College of Technology, Coimbatore have completed their project work titled ' **MICROPROCESSOR BASED FREQUENCY GENERATOR FOR RPM METER TESTING'** in our organisation during the period September 1998 to March 1999.

Name of the students :

1. MR. V. M. KRISHNAMOORTHY (95EEE16)
2. MS. K. KAVITHA (95EEE13)
3. MS. A. PRADEEPA (95EEE27)
4. MR. T. SUDHAGAR (95EEE46)

During their training period, their performance and conduct were found good.

M. BALASUBRAMANIAN
MANAGER – P&S (ELECTROINCS)

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude to our guide, Dr.K.A.Palaniswamy, B.E,Msc(Engg),Ph.d,FIE,MISTE Professor and Head Of EEE Department for his valuable guidance and motivation.

We wish to express our heartfelt thanks to K. K. Padmanaban, the Principal of our Institution for providing us the encouragement to undertake this project in the industry.

We are very thankful and grateful to our industrial guide Mr.Saravana Kumar of PRICOL as his inspiring guidance and suggestions greatly helped in completion of this project work.

Last but not the least, we express our deep sentiments to Mr.Balasubramanian, Manager PRICOL, for providing the laboratory facilities and assistance to us.

SYNOPSIS

The analog mode of frequency generation is subjected to variation in frequency due to change in temperature and also the accuracy is less. These drawbacks are overcome by using microprocessor based frequency generation for RPM meter testing.

The device consists of a hardware unit and a software package which controls the hardware. The main feature of the project is to generate square waveform at different frequencies.

The hardware unit comprises of INTEL 8085 microprocessor, INTEL 8279 keyboard / interface and in-built power supply.

The software effectively controls the hardware unit and generates the square waveform for the frequencies given in the input.

The frequency of the waveform is selected through the keyboard and the signal is observed. The design details and test results are presented in the report.

CONTENTS

INTRODUCTION

1.1 Introduction :-

A frequency generator is a unit which generates square waveform. It also gives provision for the variation of frequency. The desired frequency can be selected through keyboard.

1.2 Analog Frequency Generator :-

In conventional generator commonly found in laboratories, primitive analog devices are employed to generate waveforms. Moreover the frequency variations are not implemented in an accurate manner. This is essential for many sophisticated industrial applications.

1.3 Microprocessor Based Frequency Generator :-

It has a keyboard through which one can select the frequency to be generated. It is very user-friendly and has a compact hardware unit and a software which effectively controls the hardware. It is insensitive to temperature variations.

1.4 Advantages of Microprocessor based system :-

- It has a very high accuracy in generating the exact frequency required.
- Being insensitive to temperature variations, the system gives a distortion free output of the required waveform.
- User interaction has been enhanced using attractive display.
- User surity on the output required is enhanced by the provision of the keypad and many more useful features.

1.5 Conclusion :-

Thus the construction and implementation of the microprocessor based frequency generator sees itself as a boon to users because of its flexibility and ruggedness.

HARDWARE UNIT

2.1 Introduction :-

This chapter presents a discussion on the overall block diagram of the project and the various components of the hardware used. The hardware is perfectly simple, but very effective. The components have been chosen in such a way as to precisely fit in and fulfill the requirements of the project.

2.2 General Block Diagram :-

The general block diagram shown in fig (2.1) involves three blocks.

1. Power supply unit.
2. Processor unit.
3. Keyboard unit.

Power Supply Unit :-

Fig (2.2) shows the construction unit. As far as the construction is concerned this is a real mini-power-supply. But it can deliver upto three Amps at this output voltage of 1.5 to 25 volts.

However that integrated regulator IC is a on-chip overload protected. Then adjustable pin of the regulator is connected to the ends of the potential divider R1P1. The output is calculated from ,

$$V_0 = 1.5 * (1+p1/r1 *v)$$

The value of P1 is measured between the wiper and funtion with R1 is 0 to 2.5Kohms.Capaciter C1 is a conventional capaciter while C2 and C3 improves the regulation, protection diodes D1 and D2 ensure that at swith off the potential at the output of IC is more than it's input.The value of R1 is choosen that the minimum load current through IC is about 3.5Amps.

It is essential that IC is mounted on a heat sink rated at about 1KW donot swing on the heat conducting medium.When only low output voltages are neded it makes use of mains transformer with a lower secondary voltage should be 9volts.

When a 24volts secondary is uesd and the required output voltage is 1.25volts the maximum output is 1Amps , otherwise the maximum dissipation of the LM350 is exceeded and the internal protection will switch off the regulation. When the secondary voltage is 9Volts and V0 is 1.25Volts the maximum load current amount to 2.5Amps.

PROCESSOR UNIT

The process unit is the heart of the frequency generator. It consists of the Microprocessor 8085, the latch 74SL373, The RAM memory chip, the EPROM memory chip and a 3 to 8 decoder 74LS138.The circuit of the processor block is given in Fig 2.2.

2.4.1 LATCH

The 8085 microprocssor has a multiplexed bus AD7 to AD0 used as the lower order address bus and data bus. The bus AD7 to AD0 can be demultipluxed by using a latch and the ALE signal. The bus AD7 to AD0 can be demultiplxed by using a latch and the ALE signal.The bus AD7 to AD0 is connected as the input to latch 74SL373.The ALE signal is connected to the enable EN pin of the latch.When the ALE is high ,the latch is transparent; this means that the output changes according to input data. When the ALE goes low ,the output of the latch represents lower address bus.

2.4.2 MEMORY INTERFACING

The memory is an important part of microprocessor based system. The primary function of memory interfacing is that the microprocessor should be able to read from and write into a given register of a memory chip. To perform this operation the microprocessor should,

1. Be able to select the chip by chip enable signal.

2. Identify the register by register select signal.

3. Enable appropriate buffer by control signal.

The various signals are as follows.

A15-A12	-	CHIP ENABLE
A11-A0	-	REGISTER SELECT

A12-A15 address bits are used to select the chip.The various possible chips used in this project is RAM and EPROM.Appropriate logic levels are used to select either of the two chips.For this a 3 to 8 decoder 74LS138 is used.The output of this decoder 74LS138 is connect to chip select pin of EPROM and RAM and the remaining six output of decoder remains unused.

The address lines A0-A11 serves as the register select lines. They are therefore connected to RAM and EPROM.

The control signal MEMR is used for EPROM and MEMR and MEMW both are used for RAM.Thus the data transaction to the RAM and EPROM are done by means of memory interfacing.The software ehhectively handles all the operation required for the generation of required waveform.

2.5KEYBOARD/DISPLAY UNIT

This unit consists of the 8279 programmable keyboard/display interface. The 8279 is a hardware approach interface a matrix keyboard and a multiplexed display the 8279 is a forty pin device with two major segments keyboard and display. The keyboard segment can be connected to a sixty four contact key matrix keyboard entries are debouned and stored in FIFO(first in-first out)memory.

2.6 CIRCUIT DESCRIPTION

The diagram(fig 2.3) shows the following components.

1.The 8279 programmable keyboard/display interface.

2.A matrix keyboard with 22 keys .

3. Six seven segment LED's(DS1-DS6)

4.The 74LS156 decoder with open collecter outputs.

5.Transister as current drives.

6.The 8205 decoder for decoding logic.

Lines RL0-RL7(return lines) of the 8279 are connected to the column of the matrix keyboard and the output lines (A0-A3 and B0-B3) are connected to drive LED segments through the transistors.The three scan lines are connected to the decoder ,the 74LS156 to generate eight decoder are connected as digit drivers to turn on six seven segment display LED's; two output lines are unused . In additional, the first three output lines are also used to scan the rows of the keyboard. The 8279 has four scan lines that can be decoded to generate sixteen output lines to drive sixteen displays. The data lines of the 8279 are connected to the data bus of the 8085.

2.7 CONCLUSION

Thus, this chapter gives a clear picture of one of the most important parts of this project. The role played by each of the components in the hardware has been illustrated and the overall functioning of the hardware has been also presented . The next discusses the software part of the project.

GENERAL BLOCK DIAGRAM

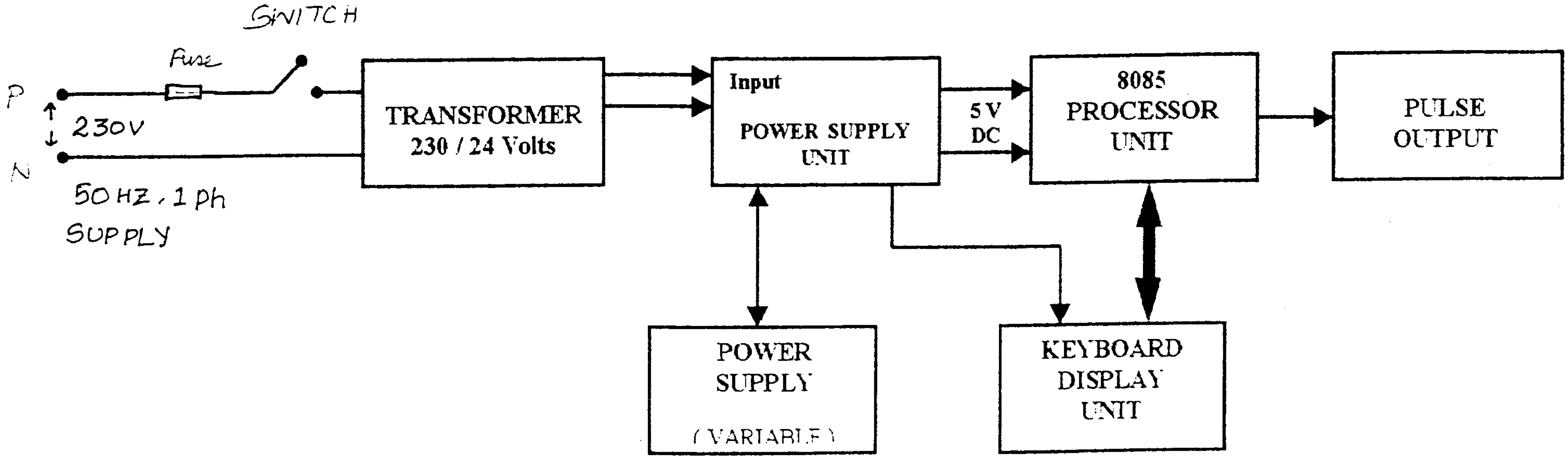
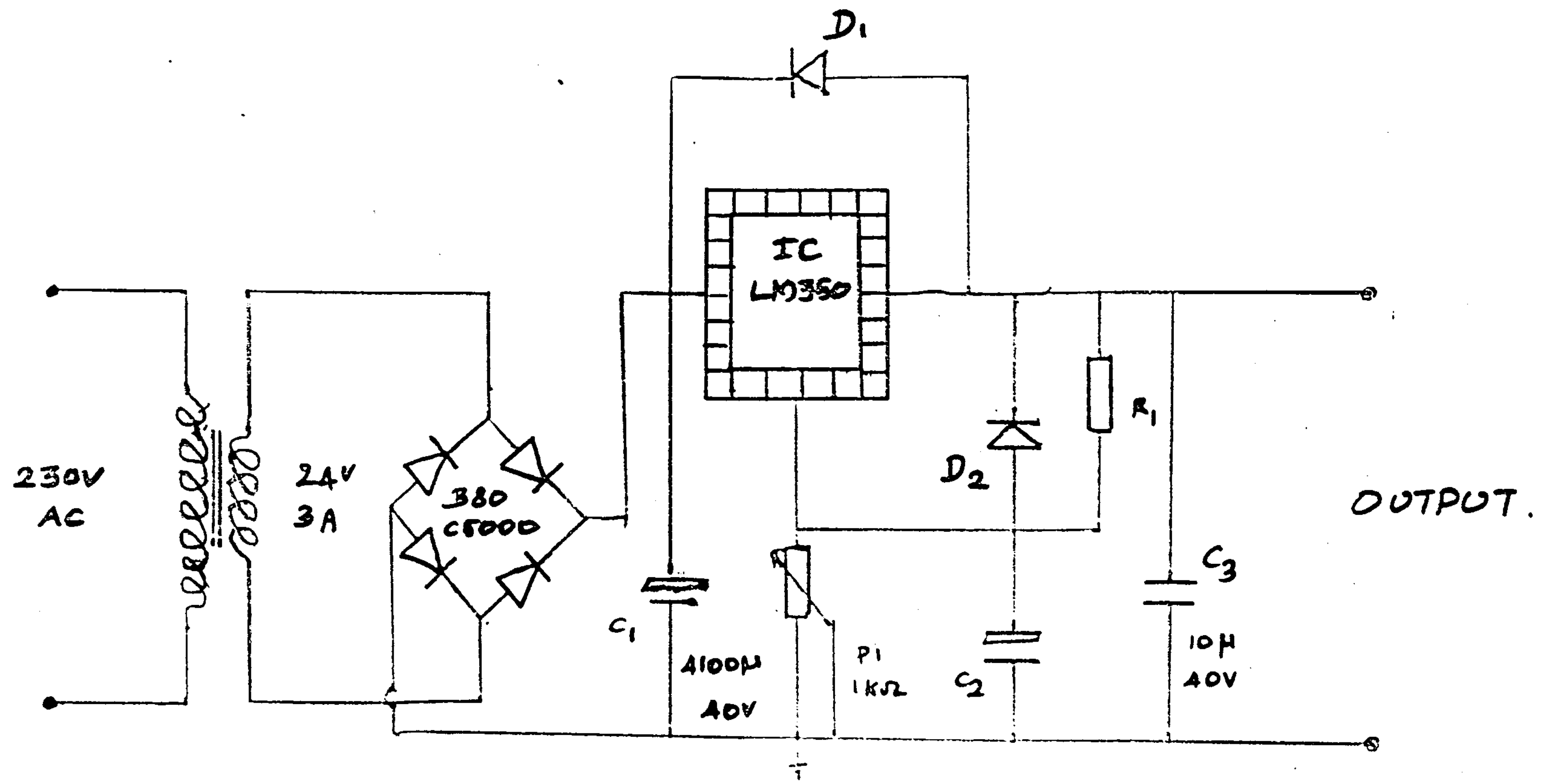


FIG. 2·1



VARIABLE 3A POWER SUPPLY *

FIG. 2·2.

MEMORY INTERFACING.

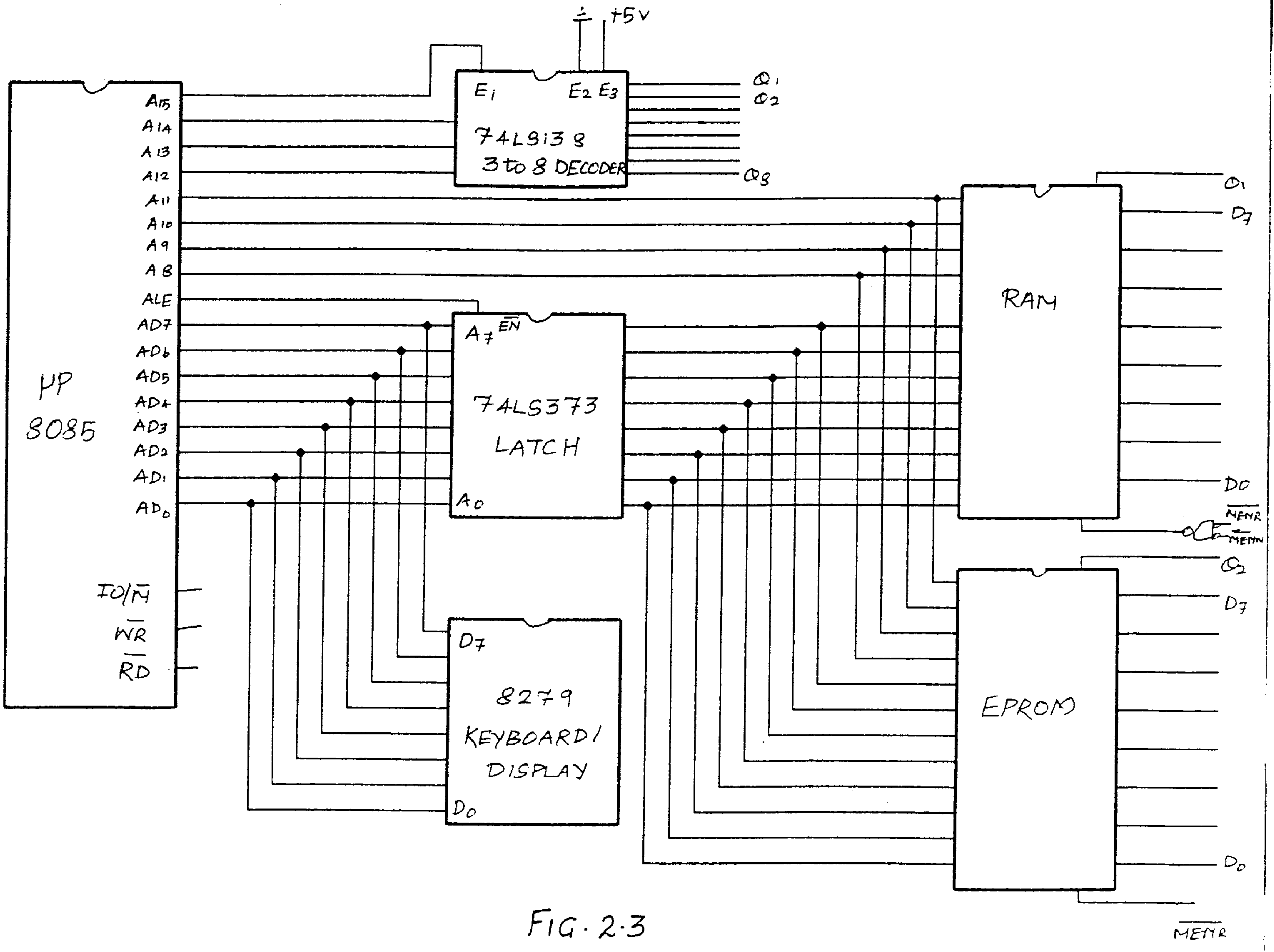
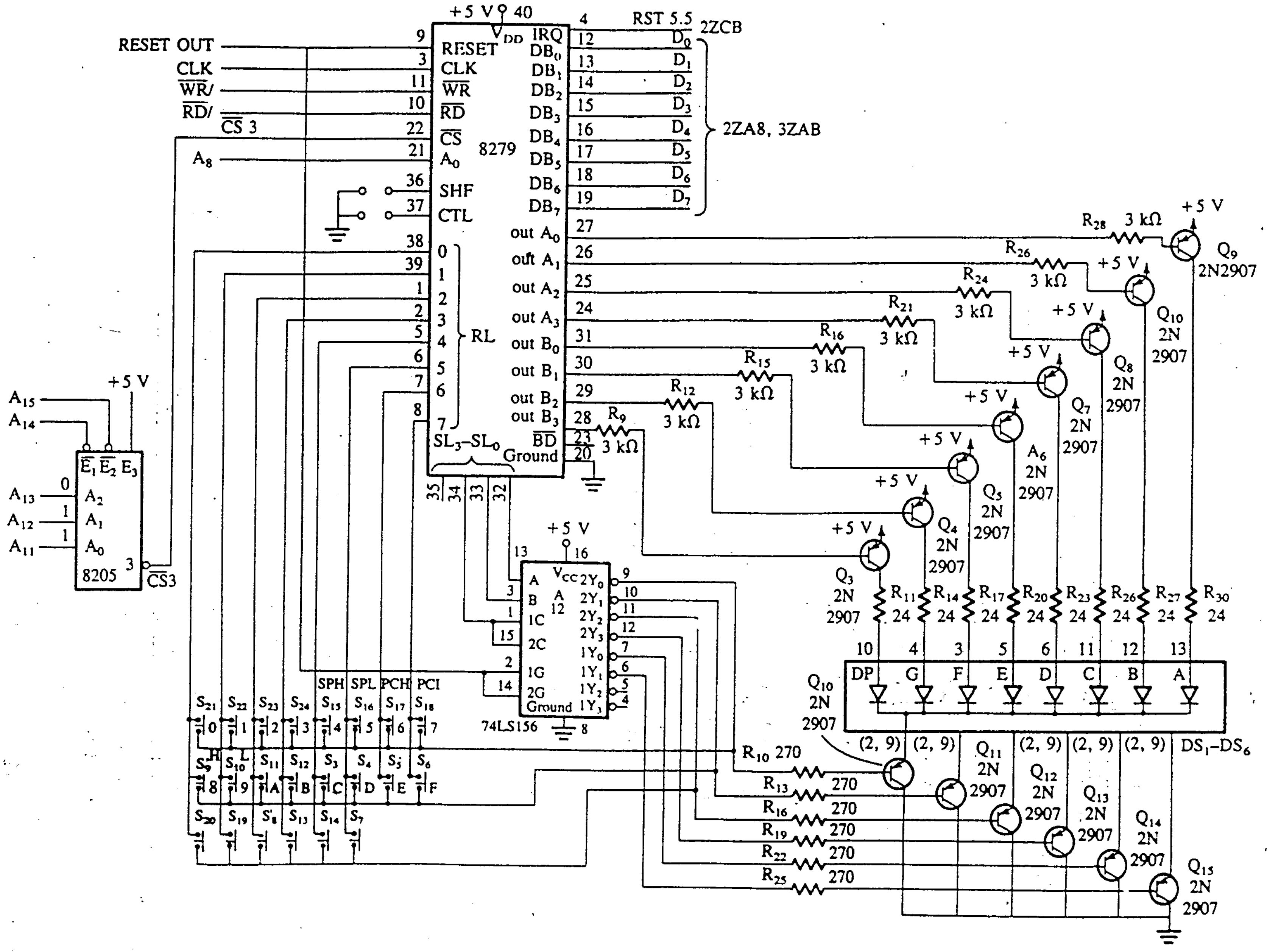


FIG. 2.3



KEYBOARD/DISPLAY FIG.2.4.

SOFTWARE PACKAGE

3.1 INTRODUCTION

In this chapter the software part of the project " Microprocessor based frequency generator" has been discussed. The software has been written in 8085 assembly code.

3.2 PROGRAMMING 8085

The process of software development in 8085 includes writing ,modifying ,testing and debugging of the user programes. The assembly language source program includes three types of statements

1. The program statements in 8085 mnemonics that are to be translated into binary code.
2. Comments that are reproduced as part of program documentation.
3. Directives to the assembler that specifies items such as starting memory locations,lable definition and required memory spaces for data.

FLOWCHART

Program assembly and execution:(fig 3.1)

As the company is interested to bring out the project in to a commercial model, the details of the software are not given in the report.

3.3 PROGRAMMING 8279

The 8279 is a complex device that can accept eight different commands to perform various functions. The initialisation commands can specify.

1. Left to Right entry and key rollover
2. Clock frequency prescaler
3. Starting address and incrementing mode of FIFO RAM
4. RAM address to read and write data and incrementing mode
5. Blanking format

3.4 GENERATION OF WAVEFORMS

The required waveform with frequency from keypad can be generated by their respective algorithms.

3.5 CONCLUSION

The assembly code is written in a microprocessor kit and tested using emulator and it is downloaded and tested along with the rest of hardware unit.

FLOW CHART.

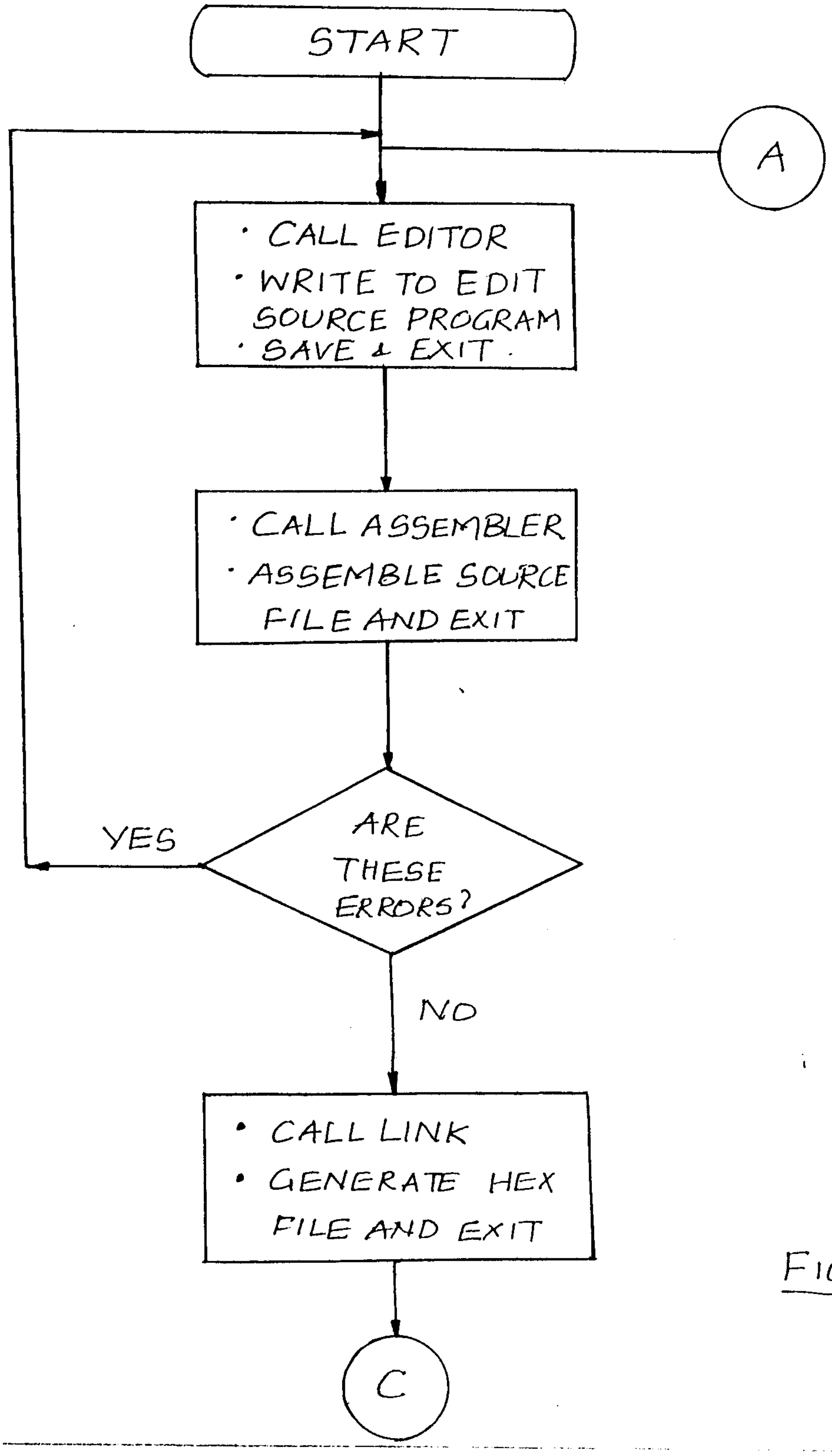
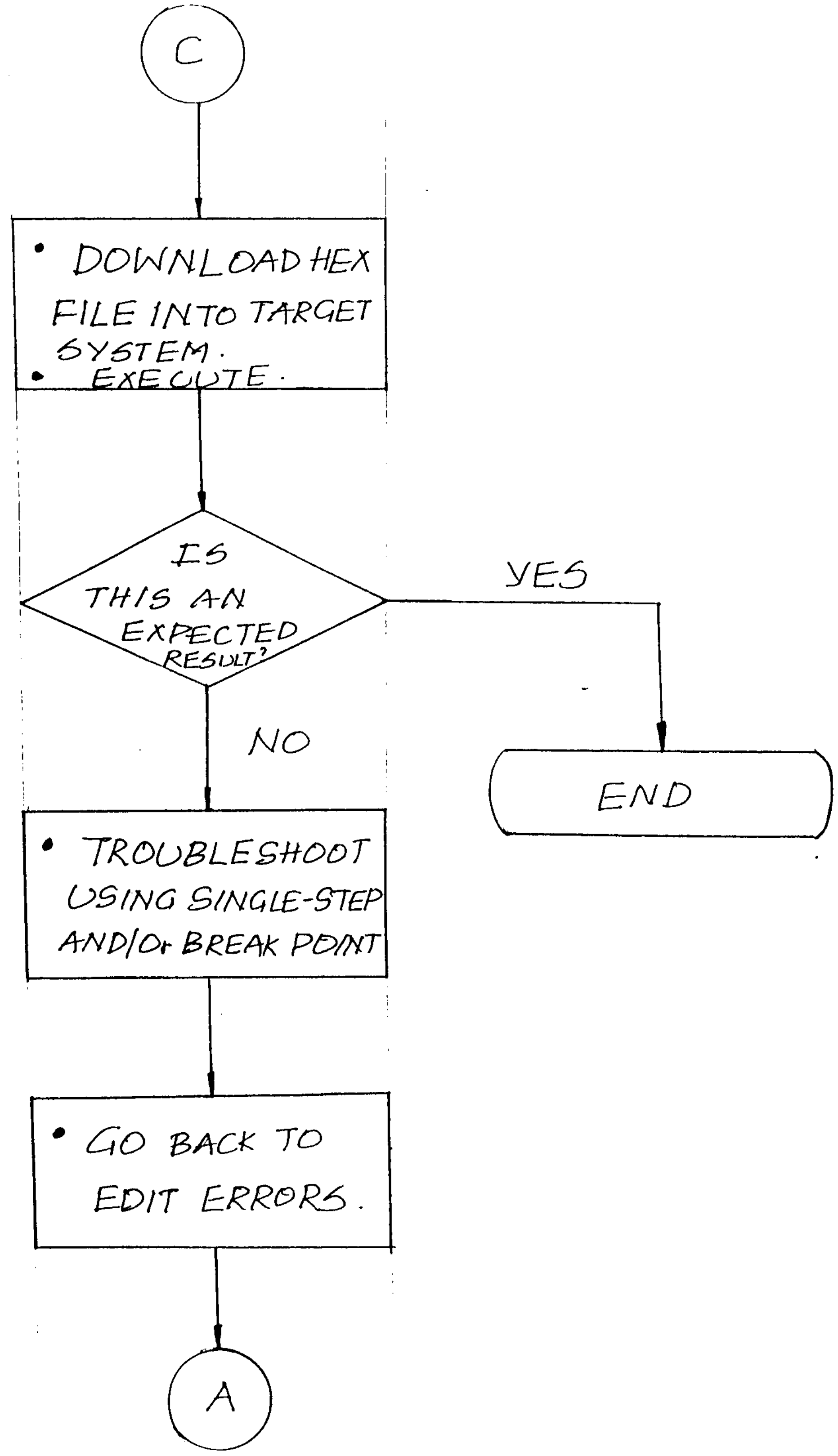

START
A
· CALL EDITOR
· WRITE TO EDIT SOURCE PROGRAM
· SAVE & EXIT.
· CALL ASSEMBLER
· ASSEMBLE SOURCE FILE AND EXIT
ARE THESE ERRORS?
YES
NO
• CALL LINK
• GENERATE HEX FILE AND EXIT
C

FIG. 3.1.


C
• DOWNLOAD HEX FILE INTO TARGET SYSTEM.
• EXECUTE.
IS THIS AN EXPECTED RESULT?
YES
NO
END
• TROUBLESHOOT USING SINGLE-STEP AND/Or BREAK POINT
• GO BACK TO EDIT ERRORS.
A

RPM METER TESTING

The testing method that is currently employed is , manual testing. In the manual testing procedure the meter is fed with a positive going pulse of amplitude of 25VOLTS and the frequencies varying between (0-466) HZ. The meter is calibrated so that it advances one step at a time for a frequency change of 33HZ . The disadvantages are completely left upto the testing staff to decide upon the accuracy of the meter.

These disadvantages are over come by using this microprocessor based frequency generation for RPM meter testing. The microprocessor generates square wave signals which is stimulated equivalent signal of engine piston movement.

Each brand of rpm meters requires a particular band of frequencies. This band of frequencies are first stored in a table created under EPROM. When this table is activated the frequencies in the band is generated sequencially and delivered to the rpm meter one by one with a time delay between them .For each frequency the rpm meter deflection has to coincide with standard deflection. Otherwise it is made to coincide with standard deflection using the adjustable knob.

This method of testing yields exact results as the frequency is generated without any error. Moreover, the user has no role in determining the accuracy of the results.

CONCLUSION

The "Microprocessor Based Frequency Generator" is very efficient when compare to the analog frequency generator. It is simple and more accurate .

The following are the suggested modification for extending the capabilities of this project.

1. Amplitude variation of waveform
2. To offset the signal
3. To provide sweep faclities
4. To extend generation to triangular and sinewave

A Microprocessor based frequency generator has been designed, fabricated and tested successfully.

•

REFRENCES

1. Ramesh S.Gaonkar, " Microprocessor Arcitecture,Programming and Applications with the 8085", Penram International Publishing ,Mumbai,Third Edition,1997.
2. Lance A.Leventhal, "Introducyion to Microprocessors: Software, Hardware,Programming",Prentice-Hall INC., New Jersey ,1978.
3. R.Somasundaram,"Microprocessor",Saraswathi Publications, Chidambaram,1996.
4. "Microprocessor Data Handbook", BPB Publications, New Delhi, 1992.
5. Mohammed Rafiquzzmann, " Microprocessor and Microcomputer based System Design ", Universal Book Stall, New Delhi, 1991.
6. Aidtya Mathur, " Introduction to Microprocessors", Tata McGraw Hill Publishing Company, New Delhi, 1989.

A.1.8085 PINOUT AND SIGNALS

The 8085 is a 8 bit general purpose microprocessor capable of addressing 64 K of memory. The device 40 pins, requires 5 volts single power supply and can operate with a 3 MHZ single phase clock.Fig A.1 shows pinout and signals of 8085.

All the signals can be classified in to six groups

1. Address bus
2. Data bus
3. Control and Status signals
4. Power supply and frequency signals
5. Extremely initiated signals
6. Serial I/O ports

A.1.1 ADDRESS BUS

The 8085 as a 8 signal lines, A15-A8.Which are unidirectional and used as the high order address bus.

A.1.2 MULTIPLEXED ADDRESS AND DATABUS

The signal lines AD7 to AD0 are bidirectional.They serve a dual purpose.They are used as low order address bus as well as the data bus.

A.1.3 CONTROL AND STATUS SIGNALS

This group of signal include two control signals,Read and Write.

Status signals are,

1.Input output /Memory,

2.S1 and S0

3.ALE-Special Signal

A.1.4 POWER SUPPLY AND CLOCK FREQUENCY

The power supply and frequency signals are as follows,

- Vcc +5V power supply.
- Vss ground reference.
- X1X2: A crystal is connected to these pins.
- CLK (OUT):Clock output.

A.1.5 EXTERNALLY INITIATED SIGNALS

The externally intiated signals include

- INTR (interrupt request)
- INTA (interrupt acknowledge)
- RST 7.5,RST 6.5,RST 5.5 (restart interrupt)
- TRAP (input)

- HOLD(input)
- HLDA(output-hold acknowledge)
- READY(input signals)

A.1.6.SERIAL INPUT-OUTPUT PORTS

The 8085 has two signals to implement the serial transmission.

- SID(serial input data)
- SOD(serial output data)

A.2 8085 ARCHITECTURE

8085 includes the ALU (Arithmetic logic unit), timing and control unit ,instruction register and decoder , register array .

A.2.1 ARITHMETIC LOGIC UNIT(ALU)

The arithmetic logic unit performs the computing functions , it includes the accumulator , the temporary register , the arithmetic and logic circuits and five flags . The temporary register is used to hold data during an arithmetic / logic operation. The flags are affected by the arithmetic and logic operation in the ALU. The flags are ,

1.S-sign flag

2.Z-zero flag

3.AC-auxillary carry flag

4.P- parity flag

5.CY-carry flag

A.2.2 TIMING AND CONTROL UNIT

This unit synchronizes all the microprocessor operations with the clock and generates the control signals necessary for communication between the microprocessor and peripherals . The control signals are similar to a synchronous pulse in an oscilloscope.

A.2.3 INSTRUCTION REGISTER AND DECODER

The instruction register and decoder are part of ALU . When an instruction is fetched from memory , it is loaded in the instruction register .The decoder decodes the instruction and establishes the sequence of events to follow .

A.2.4 REGISTER ARRAY

It has six register BCDEFH . Two additional registers called temporary register ward Z are included in the register array. These registers are used to hold 8-bit data during the execution of some instruction.

8085 PINOUT.

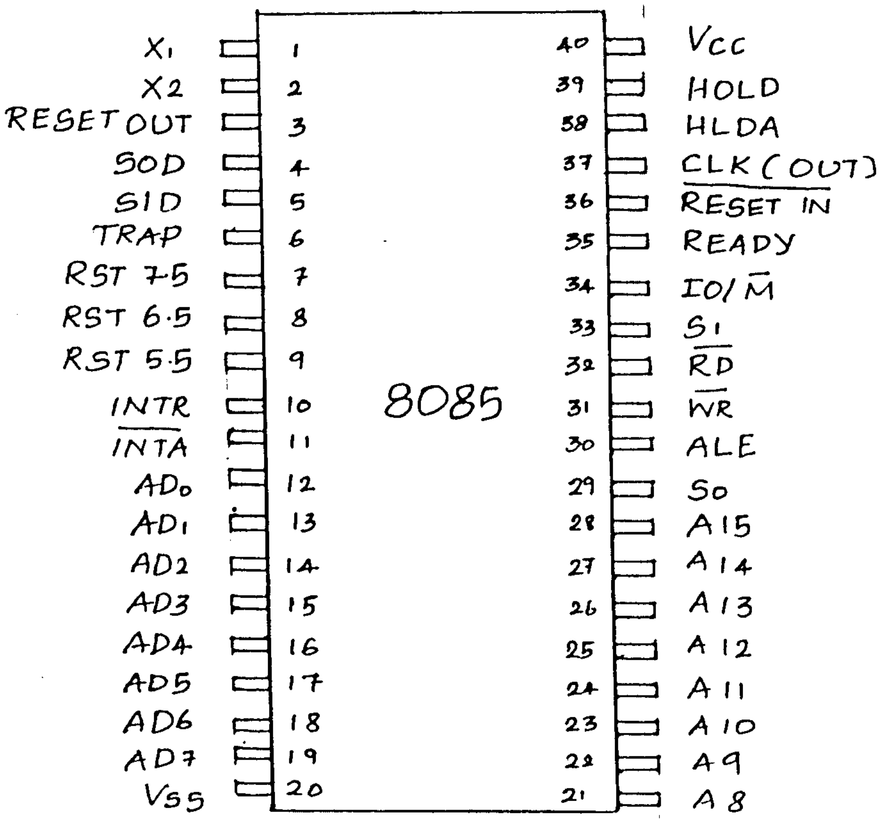
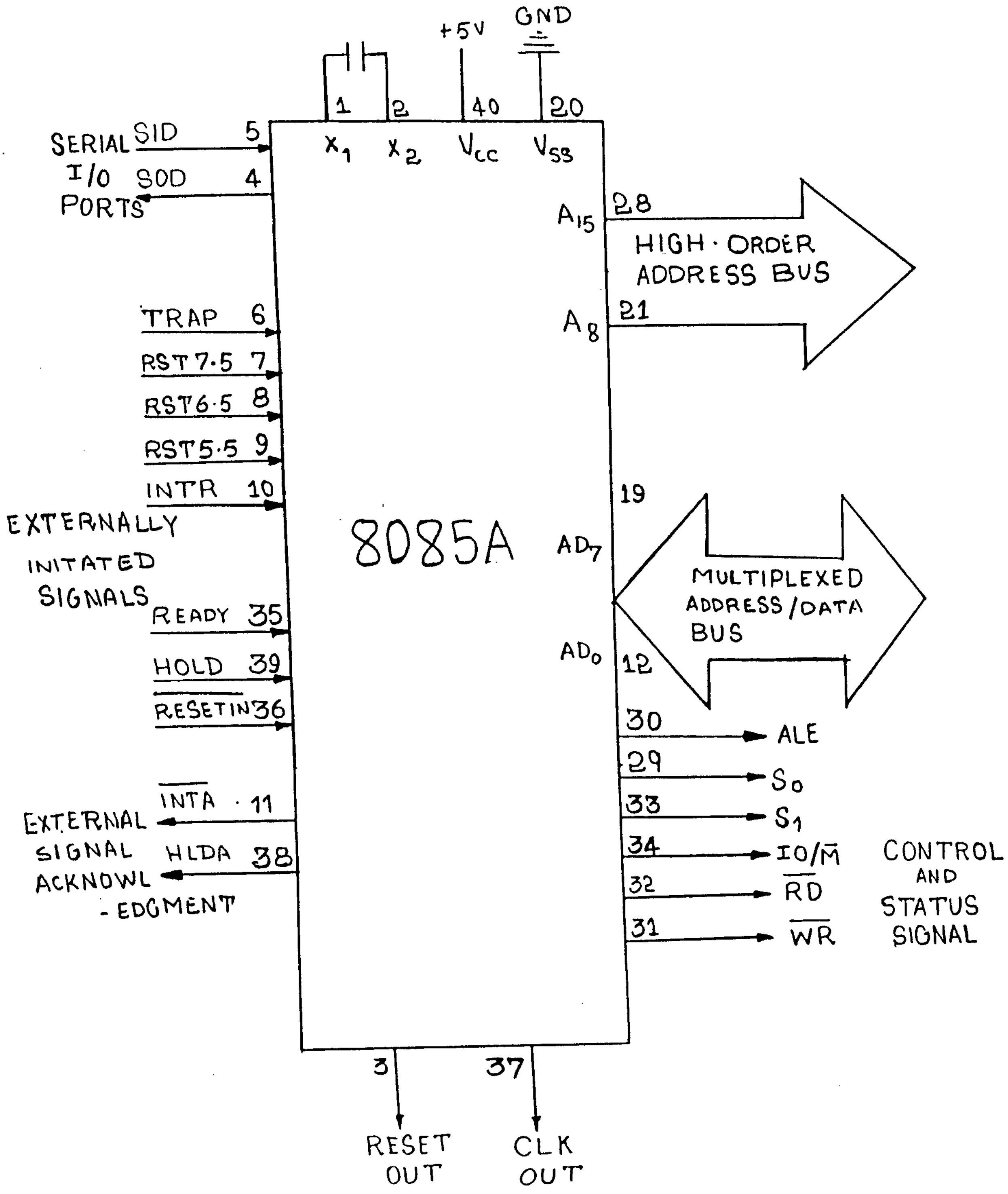


FIG. A.1.1.

8085 PINOUT AND SIGNALS



8085 Microprocessor Architecture and Memory Interfacing

3

The 8085 microprocessor is a much improved version of its predecessor, the 8080A. The 8085 includes on its chip most of the logic circuitry for performing computing tasks and for communicating with peripherals. However, eight of its bus lines are **multiplexed**; that is, they are time-shared by the low-order address and data. This chapter discusses the 8085 architecture in detail and illustrates techniques for demultiplexing the bus and generating the necessary control signals.

Later, the chapter describes a typical 8085-based microcomputer designed with general-purpose memory and I/O devices; it also illustrates the bus timing signals in executing an instruction. Then, it examines the requirements of a memory chip based on the timing signals and derives the steps necessary in interfacing memory. In addition, the chapter includes illustrations of special-purpose devices, such as the 8155 and 8755/8355, and their memory addresses in the Intel SDK-85 system.

OBJECTIVES

- □ Recognize the functions of various pins of the 8085 microprocessor.
- □ Explain the bus timings in fetching an instruction from memory.
- □ Explain how to demultiplex the AD_7–AD_0 bus using a latch.
- □ Draw a logic schematic to generate four control signals, using the 8085 $IO/\overline{M}$, $\overline{RD}$, and $\overline{WR}$ signals: (1) $\overline{MEMR}$, (2) $\overline{MEMW}$, (3) $\overline{IOR}$, and (4) $\overline{IOW}$. Explain the functions of these control signals.

3.3 MEMORY INTERFACING

Memory is an integral part of a microcomputer system, and in this chapter, our focus will be on how to interface a memory chip with the microprocessor. While executing a program, the microprocessor needs to access memory quite frequently to read instruction codes and data stored in memory; the interfacing circuit enables that access. Memory has certain signal requirements to write into and read from its registers. Similarly, the microprocessor initiates a set of signals when it wants to read from and write into memory. The interfacing process involves designing a circuit that will match the memory requirements with the microprocessor signals.

In the following sections, we will examine memory structure and its requirements and the 8085 Memory Read and Write machine cycles. Then we will derive the basic steps necessary to interface memory with the 8085. In the last chapter, we discussed a hypothetical memory chip and the concepts in addressing. In this chapter, we will illustrate memory interfacing, using memory chips such as 2732 EPROM and 6116 static R/W memory, and will discuss address decoding and memory addresses.

3.31 Memory Structure and Its Requirements

As discussed in Chapter 2, Read/Write memory (R/WM) is a group of registers to store binary information. Figure 3.11(a) shows a typical R/W memory chip; it has 2048 reg-

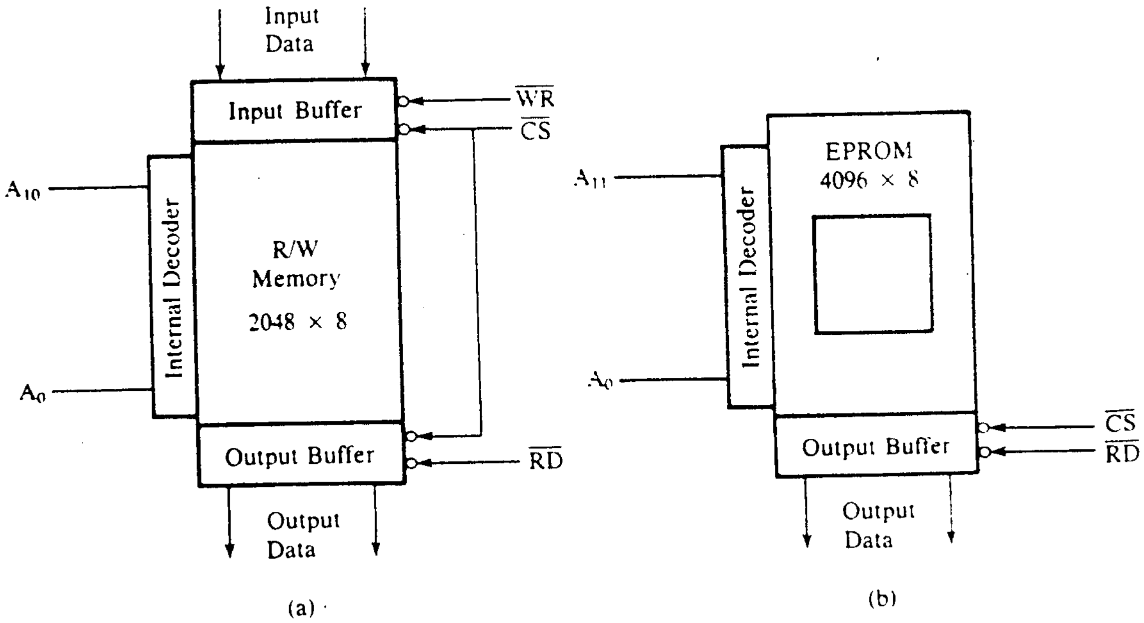


FIGURE 3.11
Typical Memory Chips: R/W Static Memory (a) and EPROM (b)

isters and each register can store eight bits indicated by eight input and eight output data lines.* The chip has 11 address lines A_{10}–A_0, one Chip Select ($\overline{CS}$), and two control lines: Read ($\overline{RD}$) to enable the output buffer and Write ($\overline{WR}$) to enable the input buffer. Figure 3.11(a) also shows the internal decoder to decode the address lines. Figure 3.11(b) shows the logic diagram of a typical EPROM (Erasable Programmable Read-Only Memory) with 4096 (4K) registers. It has 12 address lines A_{11}–A_0, one Chip Select ($\overline{CS}$), and one Read control signal. This chip must be programmed (written into) before it can be used as a read-only memory. Figure 3.11(b) also shows a quartz window on the chip that is used to expose the chip to ultraviolet rays for erasing the program. Once the chip is programmed, the window is covered with opaque tape to avoid accidental erasing. For interfacing the R/W memory, Figure 3.11(a), and the EPROM, Figure 3.11(b), the process is similar; the only difference is that the EPROM does not require the $\overline{WR}$ signal.

3.32 Basic Concepts in Memory Interfacing

The primary function of memory interfacing is that the microprocessor should be able to read from and write into a given register of a memory chip. Recall from Chapter 2 that to perform these operations, the microprocessor should

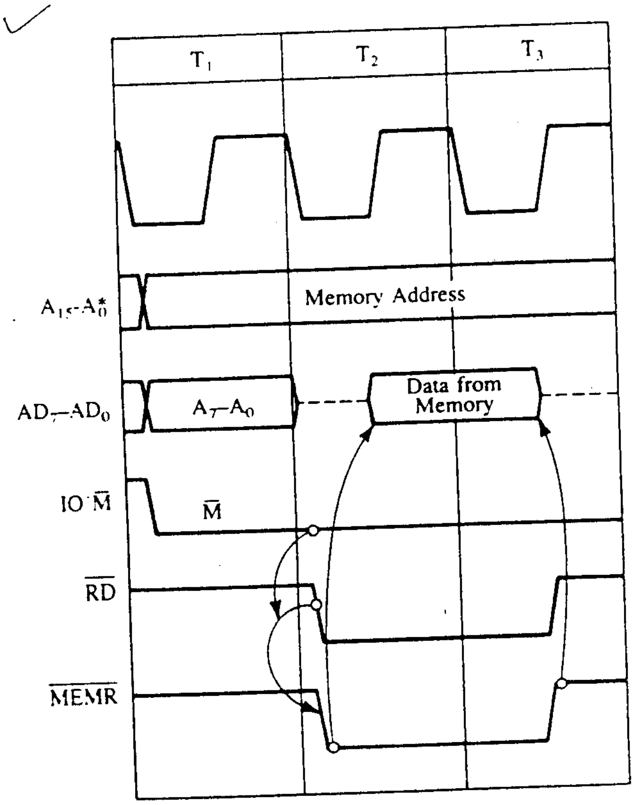
1. be able to select the chip.
2. identify the register.
3. enable the appropriate buffer.

Let us examine the timing diagram of the Memory Read operation (Figure 3.12) to understand how the 8085 can read from memory. Figure 3.12 is the M_2 cycle of Figure 3.10 except that the address bus is demultiplexed. We could also use the M_1 cycle to illustrate these interfacing concepts.

1. The 8085 places a 16-bit address on the address bus, and with this address only one register should be selected. For the memory chip in Figure 3.11(a), only 11 address lines are required to identify 2048 registers. Therefore, we can connect the low-order address lines A_{10}–A_0 of the 8085 address bus to the memory chip. The internal decoder of the memory chip will identify and select the register for the EPROM, Figure 3.11(a).
2. The remaining 8085 address lines (A_{15}–A_{11}) should be decoded to generate a Chip Select ($\overline{CS}$) signal unique to that combination of address logic (illustrated in Examples 3.3 and 3.4).
3. The 8085 provides two signals—IO/$\overline{M}$ and $\overline{RD}$—to indicate that it is a memory read operation. The IO/$\overline{M}$ and $\overline{RD}$ can be combined to generate the $\overline{MEMR}$ (Memory Read) control signal that can be used to enable the output buffer by connecting to the memory signal $\overline{RD}$.
4. Figure 3.12 also shows that memory places the data byte from the addressed register during T_2, and that is read by the microprocessor before the end of T_3.

*In a typical memory chip, the input and output data lines are not shown separately, but are shown as a group of eight data lines.

FIGURE 3.12
Timing of the Memory Read Cycle



*Demultiplexed address bus

To write into a register, the microprocessor performs similar steps as it reads from a register. Figure 3.13 shows the Memory Write cycle. In the Write operation, the 8085 places the address and data and asserts the IO/$\overline{M}$ signal. After allowing sufficient time for data to become stable, it asserts the Write ($\overline{WR}$) signal. The IO/$\overline{M}$ and $\overline{WR}$ signals can be combined to generate the $\overline{MEMW}$ control signal that enables the input buffer of the memory chip and stores the byte in the selected memory register.

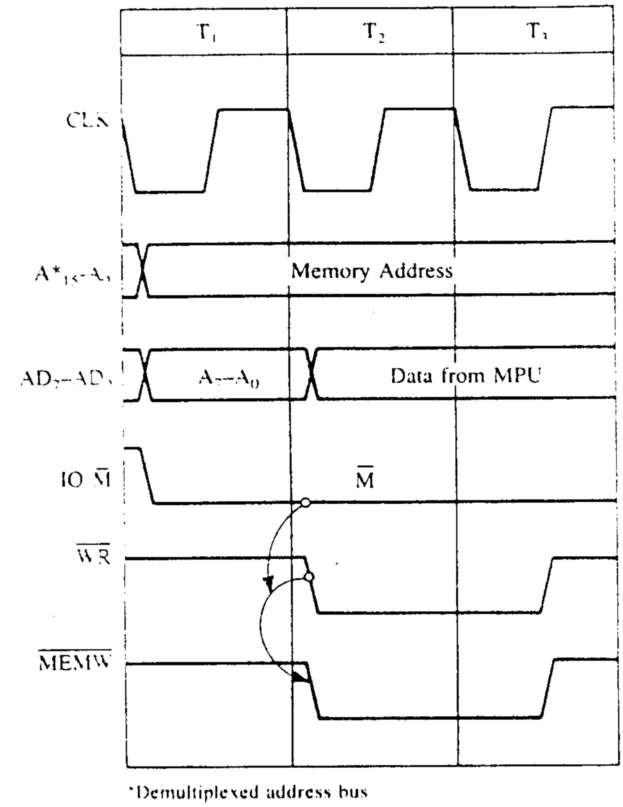
To interface memory with the microprocessor, we can summarize the above steps as follows:

1. Connect the required address lines of the address bus to the address lines of the memory chip.
2. Decode the remaining address lines of the address bus to generate the Chip Select signal, as discussed in the next section (3.33), and connect the signal to select the chip.
3. Generate control signals $\overline{MEMR}$ and $\overline{MEMW}$ by combining $\overline{RD}$ and $\overline{WR}$ signals with IO/$\overline{M}$, and use them to enable appropriate buffers.

3.33 Address Decoding

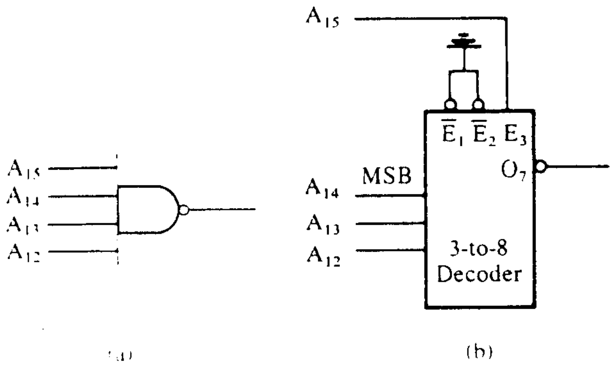
The process of address decoding should result in identifying a register for a given address. We should be able to generate a unique pulse for a given address. For example, in Figure

FIGURE 3.13
Timing of the Memory Write Cycle



3.11(b), 12 address lines (A_{11}–A_0) are connected to the memory chip, and the remaining four address lines (A_{15}–A_{12}) of the 8085 microprocessor must be decoded. Figure 3.14 shows two methods of decoding these lines: one by using a NAND gate and the other by using a 3-to-8 decoder. The output of the NAND goes active and selects the chip only when all address lines A_{15}–A_{12} are at logic 1. We can obtain the same result by using O_7 of the 3-to-8 decoder, which is capable of decoding eight different input addresses. In the decoder circuit, three input lines can have eight different logic combinations from 000 to 111; each input combination can be identified by the corresponding output line if Enable lines

FIGURE 3.14
Address Decoding Using NAND Gate (a) and 3-to-8 Decoder (b)



are active. In this circuit, the Enable lines $\overline{E}_1$ and $\overline{E}_2$ are enabled by grounding, and A_{15} must be at logic 1 to enable E_3. We will use this address decoding scheme to interface a 4K EPROM and a 2K R/W memory as illustrated in the next two examples.

3.34 Interfacing Circuit

Figure 3.15 shows an interfacing circuit using a 3-to-8 decoder to interface the 2732 EPROM memory chip. It is assumed here that the chip has already been programmed, and we will analyze the interfacing circuit in terms of the same three steps outlined previously:

□ Step 1: The 8085 address lines A_{11}–A_0 are connected to pins A_{11}–A_0 of the memory chip to address 4096 registers.

□ Step 2: The decoder is used to decode four address lines A_{15}–A_{12}. The output O_0 of the decoder is connected to Chip Enable ($\overline{CE}$). The $\overline{CE}$ is asserted only when the address on A_{15}–A_{12} is 0000; A_{15} (low) enables the decoder and the input 000 asserts the output O_0.

□ Step 3: For this EPROM, we need one control signal: Memory Read ($\overline{MEMR}$), active low. The $\overline{MEMR}$ is connected to $\overline{OE}$ to enable the output buffer; $\overline{OE}$ is the same as $\overline{RD}$ in Figure 3.11.

3.35 Address Decoding and Memory Addresses

We can obtain the address range of this memory chip by analyzing the possible logic levels on the 16 address lines. The logic levels on the address lines A_{15}–A_{12} must be 0000 to assert the Chip Enable, and the address lines A_{11}–A_0 can assume any combinations from

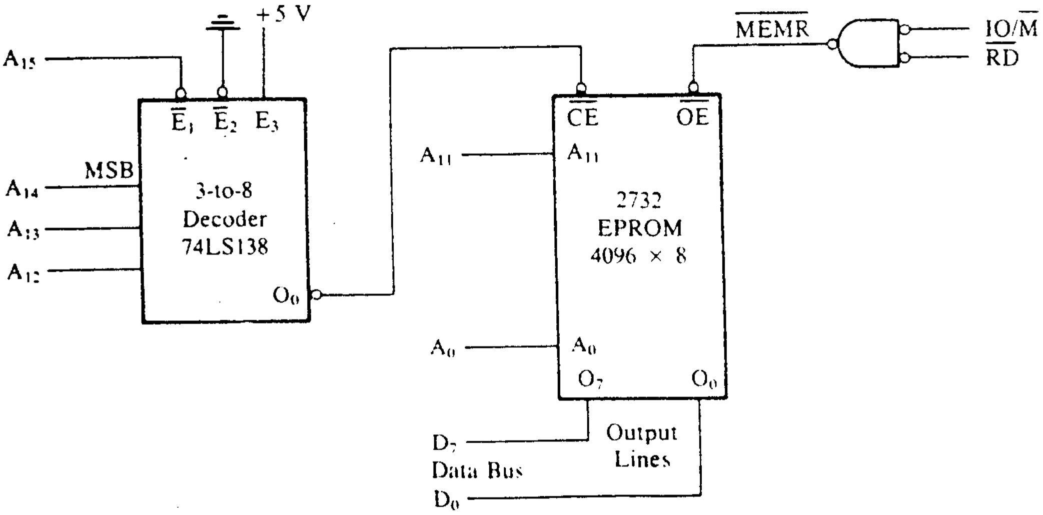


FIGURE 3.15
Interfacing the 2732 EPROM

all 0s to all 1s. Therefore, the memory address of this chip ranges from 0000H to 0FFFH, as shown below.

```
A15 A14 A13 A12   A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
 0   0   0   0     0   0   0  0  0  0  0  0  0  0  0  0  = 0000H
                   ↓                                  ↓
 0   0   0   0     1   1   1  1  1  1  1  1  1  1  1  1  = 0FFFH
└──────────────┘  └──────────────────────────────────────┘
       ↓                            ↓
  Chip Enable                 Register Select
```

We can verify the memory address range in terms of our analogy of page and line numbers, as discussed in Chapter 2, Section 2.22. The chip's 4096 bytes of memory can be viewed as 16 pages with 256 lines each. The high-order Hex digits range from 00 to 0F, indicating 16 pages—0000H to 00FFH and 0100H to 01FFH, for example.

Now, to examine how an address is decoded and how the microprocessor reads from this memory, let us assume that the 8085 places the address 0FFFH on the address bus. The address 0000 (0H) goes to the decoder, and the output line O_0 of the decoder selects the chip. The remaining address FFFH goes on the address lines of the chip, and the internal decoder of the chip decodes the address and selects the register FFFH. Thus, the address 0FFFH selects the register as shown in Figure 3.16. When the 8085 asserts the $\overline{RD}$ signal, the output buffer is enabled and the contents of the register 0FFFH are placed on the data bus for the processor to read.

Example 3.5

Analyze the interfacing circuit in Figure 3.17 and find its memory address range.

Solution

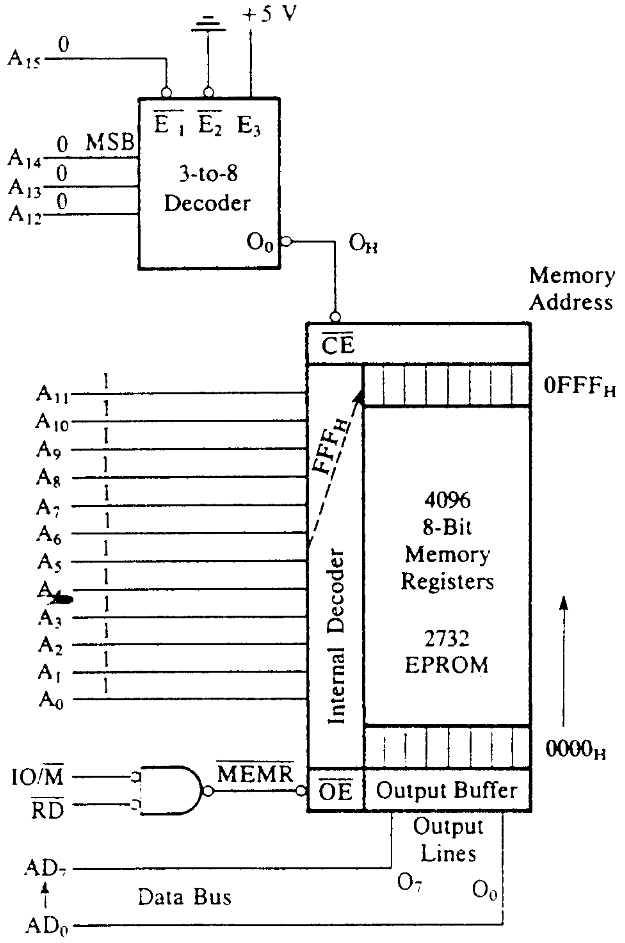
Figure 3.17 shows the interfacing of the 6116 memory chip with 2048 (2K) registers. The memory chip requires 11 address lines (A_{10}–A_0) to decode 2048 registers. The remaining address lines A_{15}–A_{11} are connected to the decoder. However, in this circuit, the decoder is enabled by the IO/$\overline{M}$ signal in addition to the address lines A_{15} and A_{14}, and the $\overline{RD}$ and $\overline{WR}$ signals of the 8085 are directly connected to the memory chip. The signals $\overline{MEMR}$ and $\overline{MEMW}$ need not be generated separately; thus, this technique saves two gates. The memory address of this chip ranges from 8800H to 8FFFH, as shown below.

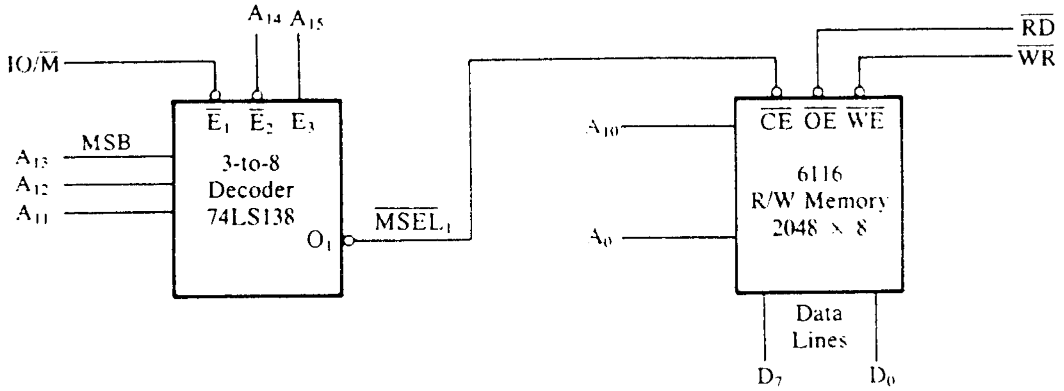
```
A15 A14 A13 A12 A11   A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
 1   0   0   0   1     0   0  0  0  0  0  0  0  0  0  0  = 8800H
                       ↓                               ↓
                       1   1  1  1  1  1  1  1  1  1  1  = 8FFFH
└──────────────────┘
         ↓
       MSEL1
```

The output line O_1 of the decoder is connected to $\overline{CE}$ of the memory chip, and it is identified as $MSEL_1$ because it is asserted only when IO/$\overline{M}$ is low.

✓

FIGURE 3.16
Address Decoding and Reading from Memory





□ The 8279 Programmable Keyboard/Display Interface
□ A matrix keyboard with 22 keys
□ Six seven-segment LEDs: DS_1–DS_6 (only one is shown; other five are identical)
□ The 74LS156 decoder with open collector outputs
□ Transistors as current drivers
□ The 8205 decoder for the decoding logic

Lines RL_0–RL_7 (Return Lines) of the 8279 are connected to the columns of the matrix keyboard, and the output lines (A_0–A_3 and B_0–B_3) are connected to drive the LED segments through the transistors. The three scan lines are connected to the decoder, the 74LS156, to generate eight decoded signals. In this circuit, six output lines of the decoder are connected as digit drivers to turn on six seven-segment LEDs; two output lines are unused. In addition, the first three output lines are also used to scan the rows of the keyboard. The 8279 has four scan lines that can be decoded to generate 16 output lines to drive 16 displays. The data lines of the 8279 are connected to the data bus of the 8085, and the IRQ (Interrupt Request) is connected to the RST 5.5 of the system.

Four signals—$\overline{RD}$, $\overline{WR}$, CLK, and RESET OUT— are connected directly from the 8085. The system has a 3.072 MHz clock; when the 8279 is reset, the clock prescaler is set to 31. This divides the clock frequency by 31 to provide the scan frequency of approximately 100 kHz. The RESET signal also sets the 8279 in the mode of 16-character display with two-key lockout keyboard.

After the initialization of the 8279, the respective codes are sent to the display RAM to display any characters. The 8279 takes over the task of displaying the characters by outputting the codes and digit strobes. To read the keyboard, the 8279 scans the columns; if a key closure is detected, it debounces the key. If a key closure is valid, it loads the key code into the FIFO, and the IRQ line goes high to interrupt the system.

DECODING LOGIC AND PORT ADDRESSES

The port addresses of the 8279 registers are determined by two signals: $\overline{CS}$ and A_0. The $\overline{CS}$ signal of the 8279 is connected to the $\overline{CS3}$ output of the 8205 decoder, and A_0 is directly connected to the address line A_8 of the 8085. For commands and status, A_0 should be high. For data transfer, A_0 should be low. By examining the decoder input lines and assuming logic 0 for the don't care address lines, one can determine that the port addresses are as follows:

$$\underbrace{\begin{matrix}A_{15} & A_{14}\\ 0 & 0\end{matrix}}_{\text{Decoder Enable}}\;\underbrace{\begin{matrix}A_{13} & A_{12} & A_{11}\\ 0 & 1 & 1\end{matrix}}_{\text{Decoder Input}}\;\underbrace{\begin{matrix}A_{10} & A_{9}\\ 0 & 0\end{matrix}}_{\text{Don't Care}}\;\underbrace{\begin{matrix}A_{8}\\ I/O\end{matrix}}_{A_0\text{ of 8279}}\;\underbrace{\begin{matrix}A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 & A_0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{matrix}}_{\text{Don't Care}} = 1900H/1800H$$

Command/Status Port: 1900H
Data Port: 1800H

Solution

In Figure 14.16, bit D_0 should be initialized as an output line and bit D_7 should be initialized as an input line. When the switch is off, D_7 remains high because it is tied to +5 V through a resistor, and when the switch is turned on, bit D_7 becomes logic 0. To turn on the motor, we should provide logic 1 to bit D_0; this will turn on the transistor driver.

Instructions

```
CNTRLB EQU 03H
PORTB  EQU 01H
START: MVI A,00000001B   ;Control word to set up bit D7 as input
                         ;  and bit D0 as output, bits D1-D6 are
                         ;  don't care
       OUT CNTRLB        ;Initialize port B
READ:  IN PORTB          ;Read the switch
       RAL               ;Check bit D7
       JC READ           ;If bit D7 is high, keep reading
       MVI A,00000001B   ;D0 = 1 to turn on transistor
       OUT PORTB         ;Turn on the motor
       HLT
```

THE 8279 PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE 14.3

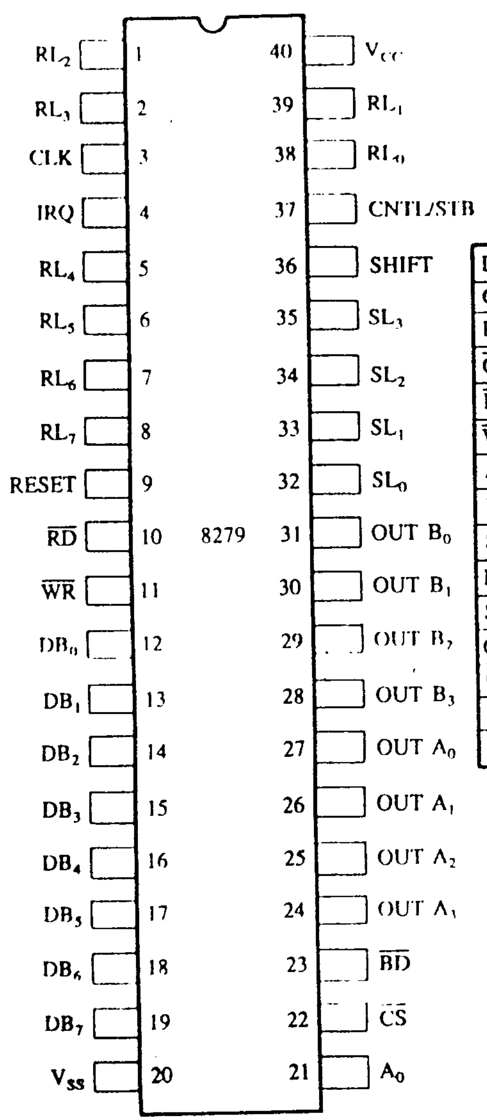
The 8279 is a hardware approach to interfacing a matrix keyboard and a multiplexed display. The software approach to interfacing a matrix keyboard and a multiplexed display of seven-segment LEDs is illustrated in Chapter 17. The disadvantage of the software approach is that the microprocessor is occupied for a considerable amount of time in checking the keyboard and refreshing the display. The 8279 relieves the processor from these two tasks. The disadvantage of using the 8279 is the cost. The trade-offs between the hardware approach and the software approach are the production cost vs. the processor time and the software development cost.

The 8279 (Figure 14.17) is a 40-pin device with two major segments: keyboard and display. The keyboard segment can be connected to a 64-contact key matrix. Keyboard entries are debounced and stored in the internal FIFO (First-In-First-Out) memory; an interrupt signal is generated with each entry. The display segment can provide a 16-character scanned display interface with such devices as LEDs. This segment has 16 × 8 R/W memory (RAM), which can be used to read/write information for display purposes. The display can be set up in either right-entry or left-entry format.

14.31 Block Diagram of the 8279

The block diagram (Figure 14.18) shows four major sections of the 8279: keyboard, scan, display, and MPU interface. The functions of these sections are described below.

Pin Configuration



Pin Names

DB_{0-7}	I/O	Data Bus (Bidirectional)
CLK	I	Clock Input
RESET	I	Reset Input
$\overline{CS}$	I	Chip Select
$\overline{RD}$	I	Read Input
$\overline{WR}$	I	Write Input
A_0	I	Buffer Address
IRQ	O	Interrupt Request Output
SL_{0-3}	O	Scan Lines
RL_{0-7}	I	Return Lines
SHIFT	I	Shift Input
CNTL/STB	I	Control Strobe Input
OUT A_{0-3}	O	Display (A) Outputs
OUT B_{0-3}	O	Display (B) Outputs
$\overline{BD}$	O	Blank Display Output

Logic Symbol

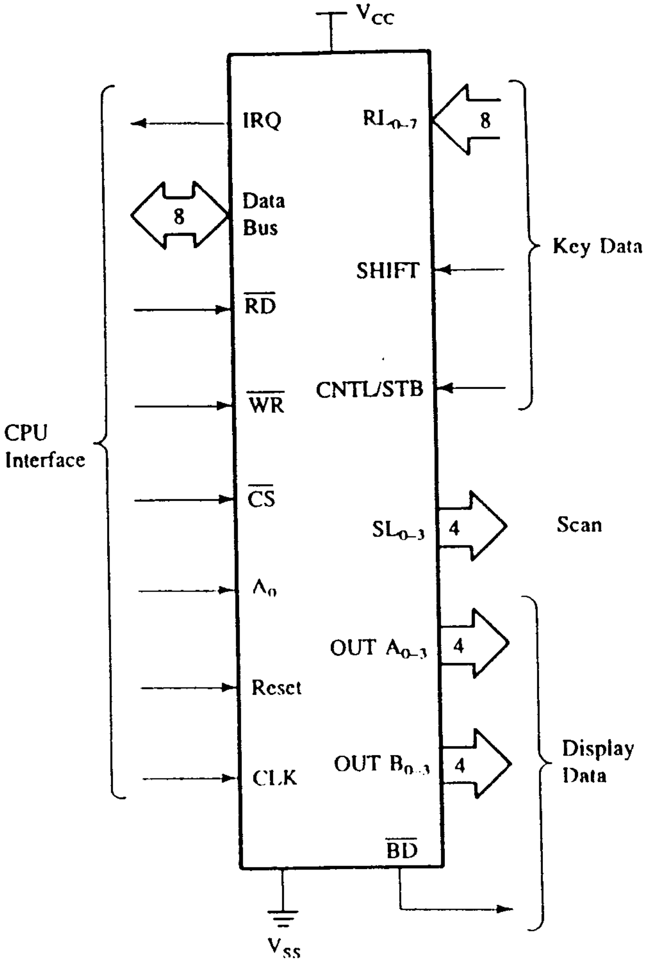


FIGURE 14.17
The 8279 Logic Pinout

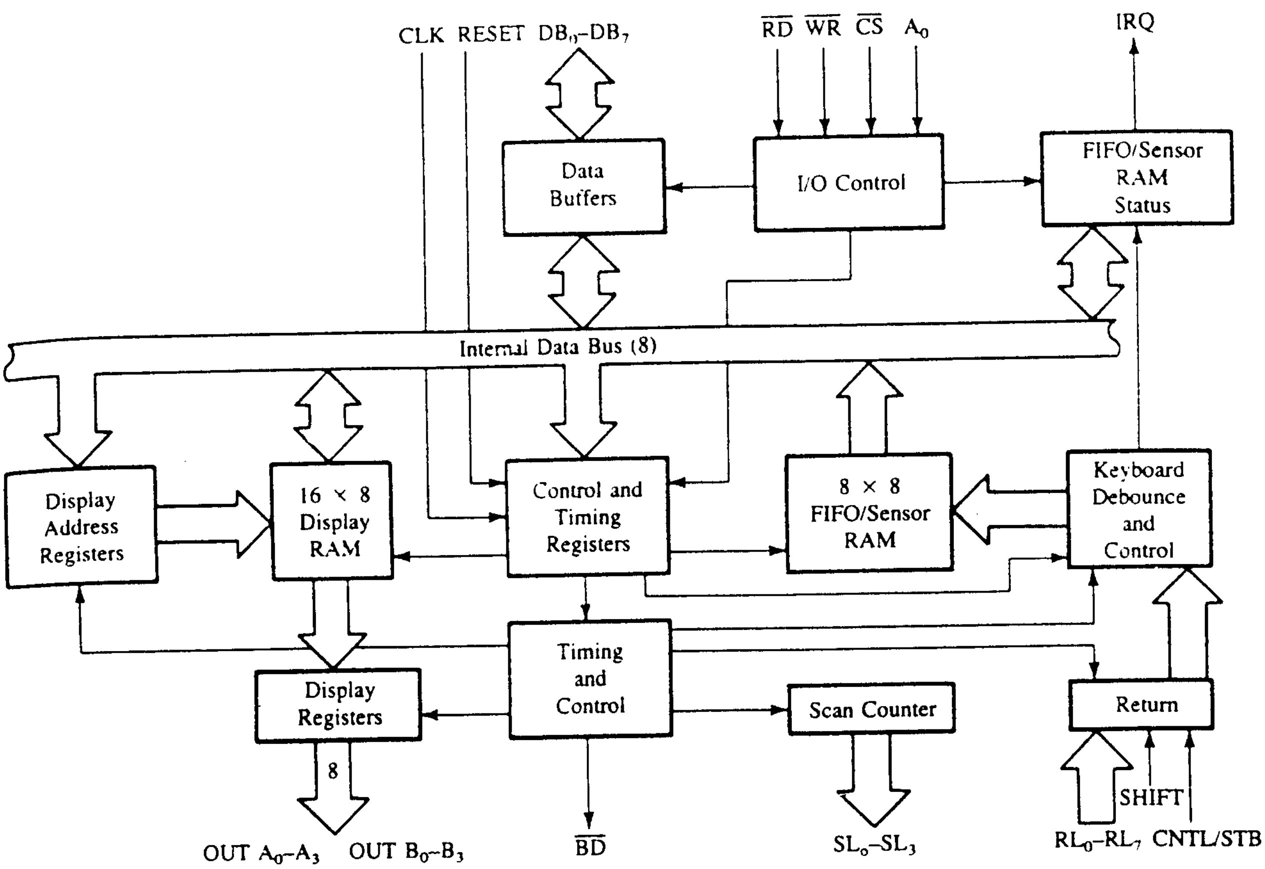


FIGURE 14.18
The 8729 Logic Block Diagram
SOURCE: Intel Corporation. *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3-218.

KEYBOARD SECTION

This section has eight lines (RL_0–RL_7) that can be connected to eight columns of a keyboard, plus two additional lines: Shift and CNTL/STB (Control/Strobe). The status of the SHIFT key and the Control key can be stored along with a key closure. The keys are automatically debounced, and the keyboard can operate in two modes: two-key lockout or N-key rollover. In the two-key lockout mode, if two keys are pressed almost simultaneously, only the first key is recognized. In the N-key rollover mode, simultaneous keys are recognized and their codes are stored in the internal buffer; it can also be set up so that no key is recognized until only one key remains pressed.

The keyboard section also includes 8 × 8 FIFO (First-In–First-Out) RAM. The FIFO RAM consists of eight registers that can store eight keyboard entries; each is then read in the order of entries. The status logic keeps track of the number of entries and provides an IRQ (Interrupt Request) signal when the FIFO is not empty.

3.1 THE 8085 MPU

The term **microprocessing unit** (MPU) is similar to the term **central processing unit** (CPU) used in traditional computers. We define the MPU as a device or a group of devices (as a unit) that can communicate with peripherals, provide timing signals, direct data flow, and perform computing tasks as specified by the instructions in memory. The unit will have the necessary lines for the address bus, the data bus, and the control signals, and would require only a power supply and a crystal (or equivalent frequency-determining components) to be completely functional.

Using this description, the 8085 microprocessor can almost qualify as an MPU, but with the following two limitations.

1. The low-order address bus of the 8085 microprocessor is **multiplexed** (time-shared) with the data bus. The buses need to be demultiplexed.
2. Appropriate control signals need to be generated to interface memory and I/O with the 8085. (Intel has some specialized memory and I/O devices that do not require such control signals.)

This section shows how to demultiplex the bus and generate the control signals after describing the 8085 microprocessor and illustrates the bus timings.

3.1.1 The 8085 Microprocessor

The 8085A (commonly known as the 8085) is an 8-bit general-purpose microprocessor capable of addressing 64K of memory. The device has forty pins, requires a +5 V single power supply, and can operate with a 3-MHz single-phase clock. The 8085A-2 version can operate at the maximum frequency of 5 MHz. The 8085 is an enhanced version of its predecessor, the 8080A; its instruction set is upward-compatible with that of the 8080A, meaning that the 8085 instruction set includes all the 8080A instructions plus some additional ones. Programs written for the 8080A will be executed by the 8085, but the 8085 and the 8080A are not pin-compatible.

Figure 3.1 shows the logic pinout of the 8085 microprocessor. All the signals can be classified into six groups: (1) address bus, (2) data bus, (3) control and status signals, (4) power supply and frequency signals, (5) externally initiated signals, and (6) serial I/O ports.

TABLE 3.1
8085 Machine Cycle Status and Control Signals

Machine Cycle	Status			Control Signals
	$IO/\overline{M}$	S_1	S_0	
Opcode Fetch	0	1	1	$\overline{RD} = 0$
Memory Read	0	1	0	$\overline{RD} = 0$
Memory Write	0	0	1	$\overline{WR} = 0$
I/O Read	1	1	0	$\overline{RD} = 0$
I/O Write	1	0	1	$\overline{WR} = 0$
Interrupt Acknowledge	1	1	1	$\overline{INTA} = 0$
Halt	Z	0	0	$\overline{RD}$, $\overline{WR}$ = Z and $\overline{INTA}$ = 1
Hold	Z	X	X	
Reset	Z	X	X	

NOTE: Z = Tri-state (high impedance)
X = Unspecified

EXTERNALLY INITIATED SIGNALS, INCLUDING INTERRUPTS

The 8085 has five interrupt signals (see Table 3.2) that can be used to interrupt a program execution. One of the signals, INTR (Interrupt Request), is identical to the 8080A microprocessor interrupt signal (INT); the others are enhancements to the 8080A. The microprocessor acknowledges an interrupt request by the $\overline{INTA}$ (Interrupt Acknowledge) signal. (The interrupt process is discussed in Chapter 12.)

In addition to the interrupts, three pins—RESET, HOLD, and READY—accept the externally initiated signals as inputs. To respond to the HOLD request, it has one signal

TABLE 3.2
8085 Interrupts and Externally Initiated Signals

• INTR (Input)	Interrupt Request: This is used as a general-purpose interrupt; it is similar to the INT signal of the 8080A.
• $\overline{INTA}$ (Output)	Interrupt Acknowledge: This is used to acknowledge an interrupt.
• RST 7.5 (Inputs) RST 6.5 RST 5.5	Restart Interrupts: These are vectored interrupts and transfer the program control to specific memory locations. They have higher priorities than the INTR interrupt. Among these three, the priority order is 7.5, 6.5, and 5.5.
• TRAP (Input)	This is a nonmaskable interrupt and has the highest priority.
• HOLD (Input)	This signal indicates that a peripheral such as a DMA (Direct Memory Access) controller is requesting the use of the address and data buses.
• HLDA (Output)	Hold Acknowledge: This signal acknowledges the HOLD request.
• READY (Input)	This signal is used to delay the microprocessor Read or Write cycles until a slow-responding peripheral is ready to send or accept data. When this signal goes low, the microprocessor waits for an integral number of clock cycles until [illegible]

ADDRESS BUS

The 8085 has eight signal lines, $A_{15}-A_8$, which are unidirectional and used as the high-order address bus.

MULTIPLEXED ADDRESS/DATA BUS

The signal lines AD_7-AD_0 are bidirectional: they serve a dual purpose. They are used as the low-order address bus as well as the data bus. In executing an instruction, during the earlier part of the cycle, these lines are used as the low-order address bus. During the later part of the cycle, these lines are used as the data bus. (This is also known as multiplexing the bus.) However, the low-order address bus can be separated from these signals by using a latch.

CONTROL AND STATUS SIGNALS

This group of signals includes two control signals ($\overline{RD}$ and $\overline{WR}$), three status signals ($IO/\overline{M}$, S_1 and S_0) to identify the nature of the operation, and one special signal (ALE) to indicate the beginning of the operation. These signals are as follows:

□ ALE—Address Latch Enable: This is a positive going pulse generated every time the 8085 begins an operation (machine cycle): it indicates that the bits on AD_7-AD_0 are address bits. This signal is used primarily to latch the low-order address from the multiplexed bus and generate a separate set of eight address lines, A_7-A_0.

□ $\overline{RD}$—Read: This is a Read control signal (active low). This signal indicates that the selected I/O or memory device is to be read and data are available on the data bus.

□ $\overline{WR}$—Write: This is a Write control signal (active low). This signal indicates that the data on the data bus are to be written into a selected memory or I/O location.

□ $IO/\overline{M}$: This is a status signal used to differentiate between I/O and memory operations. When it is high, it indicates an I/O operation; when it is low, it indicates a memory operation. This signal is combined with $\overline{RD}$ (Read) and $\overline{WR}$ (Write) to generate I/O and memory control signals.

□ S_1 and S_0: These status signals, similar to $IO/\overline{M}$, can identify various operations, but they are rarely used in small systems. (All the operations and their associated status signals are listed in Table 3.1 for reference.)

POWER SUPPLY AND CLOCK FREQUENCY

The power supply and frequency signals are as follows:

□ V_{CC}: +5 V power supply.

□ V_{SS}: Ground Reference.

□ X_1, X_2: A crystal (or RC, LC network) is connected at these two pins. The frequency is internally divided by two; therefore, to operate a system at 3 MHz, the crystal should have a frequency of 6 MHz.

□ CLK (OUT)—Clock Output: This signal can be used as the system clock for other devices.

3.13 Demultiplexing the Bus AD_7–AD_0

The need for demultiplexing the bus AD_7–AD_0 becomes easier to understand after examining Figure 3.3. This figure shows that the address on the high-order bus (20H) remains on the bus for three clock periods. However, the low-order address (05H) is lost after the first clock period. This address needs to be latched and used for identifying the memory address. If the bus AD_7–AD_0 is used to identify the memory location (2005H), the address will change to 204FH after the first clock period.

Figure 3.4 shows a schematic that uses a latch and the ALE signal to demultiplex the bus. The bus AD_7–AD_0 is connected as the input to the latch 74LS373. The ALE signal is connected to the Enable (G) pin of the latch, and the Output control ($\overline{OC}$) signal of the latch is grounded.

Figure 3.3 shows that the ALE goes high during T_1. When the ALE is high, the latch is transparent; this means that the output changes according to input data. During T_1, the

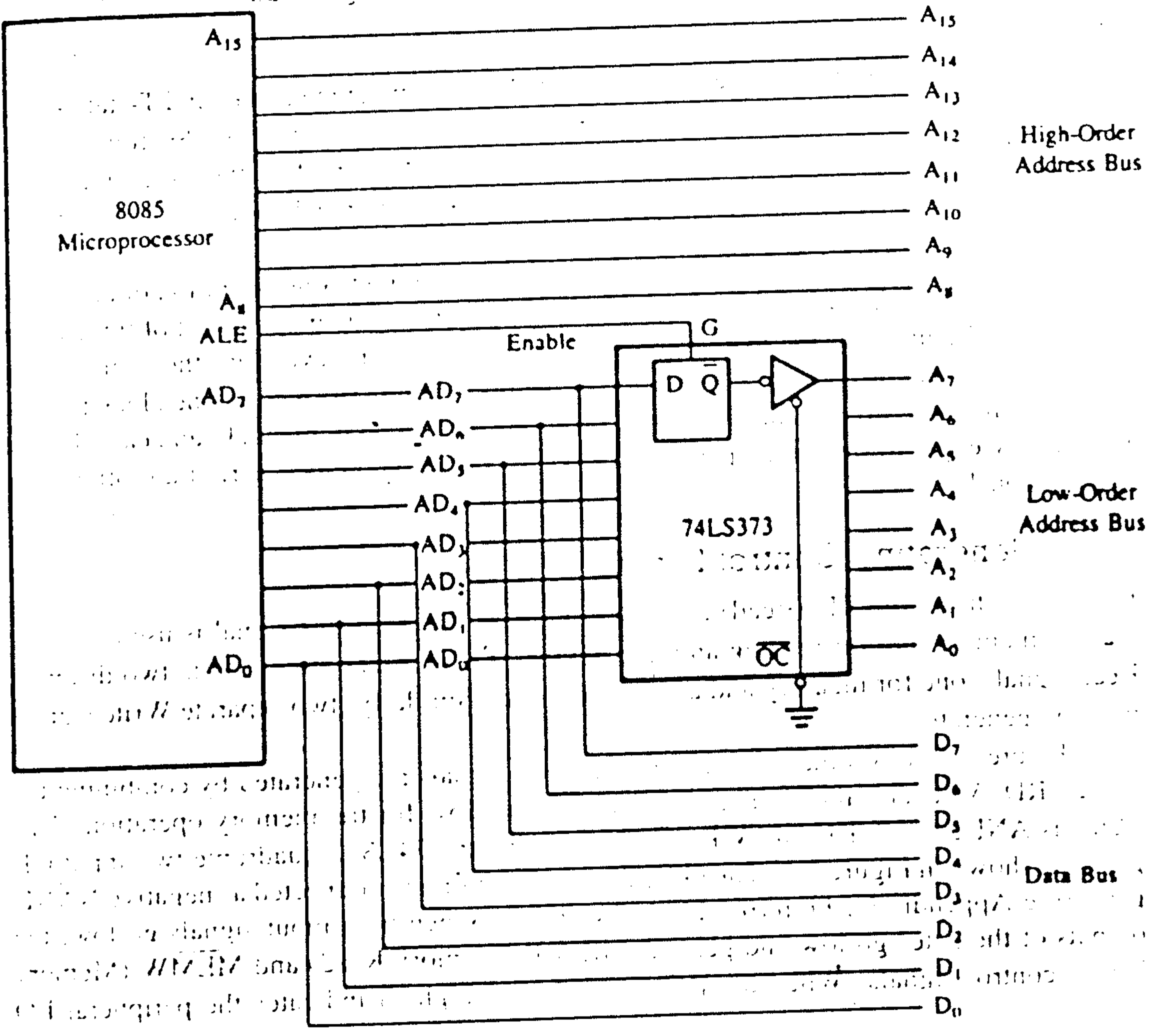


FIGURE 3.4
Schematic of Latching Low-Order Address Bus

output of the latch is 05H. When the ALE goes low, the data byte 05H is latched until the next ALE, and the output of the latch represents the low-order address bus A_7–A_0 after the latching operation.

Intel has circumvented the problem of demultiplexing the low-order bus by designing special devices such as the 8155 (256 bytes of R/W memory + I/Os) and 8355 (2K ROM + I/Os), which are compatible with the 8085 multiplexed bus. These devices internally demultiplex the bus using the ALE signal (see Figures 3.18 and 3.19).

After carefully examining Figure 3.3, we can make the following observations:

1. The machine code 4FH (0100 1000) is a one-byte instruction that copies the contents of the accumulator into register C.
2. The 8085 microprocessor requires one external operation—fetching a machine code* from memory location 2005H.
3. The entire operation—fetching, decoding, and executing—requires four clock periods.

Now we can define three terms—instruction cycle, machine cycle, and T-state—and use these terms later for examining timings of various 8085 operations (Section 3.2).

Instruction cycle is defined as the time required to complete the execution of an instruction. The 8085 instruction cycle consists of one to six machine cycles or one to six operations.

Machine cycle is defined as the time required to complete one operation of accessing memory, I/O, or acknowledging an external request. This cycle may consist of three to six T-states. In Figure 3.3, the instruction cycle and the machine cycle are the same.

T-state is defined as one subdivision of the operation performed in one clock period. These subdivisions are internal states synchronized with the system clock, and each T-state is precisely equal to one clock period. The terms T-state and clock period are often used synonymously.

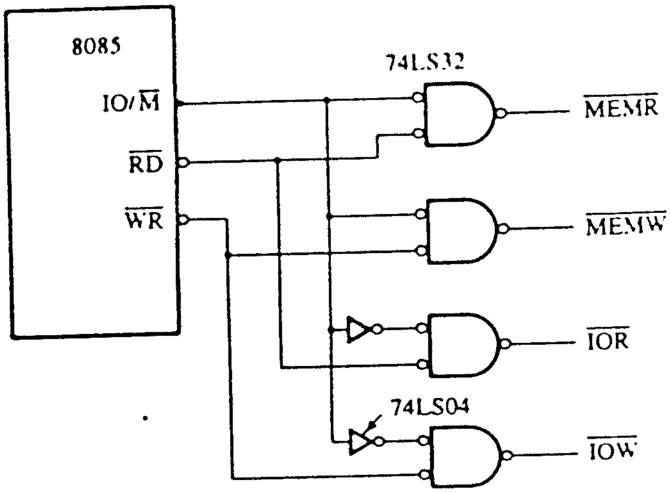
3.14 Generating Control Signals

Figure 3.3 shows the $\overline{RD}$ (Read) as a control signal. Because this signal is used both for reading memory and for reading an input device, it is necessary to generate two different Read signals: one for memory and another for input. Similarly, two separate Write signals must be generated.

Figure 3.5 shows that four different control signals are generated by combining the signals $\overline{RD}$, $\overline{WR}$, and IO/$\overline{M}$. The signal IO/$\overline{M}$ goes low for the memory operation. This signal is ANDed with $\overline{RD}$ and $\overline{WR}$ signals by using the 74LS32 quadruple two-input OR gates, as shown in Figure 3.5. The OR gates are functionally connected as negative NAND gates (see Appendix C, Preferred Logic Symbols). When both input signals go low, the outputs of the gates go low and generate $\overline{MEMR}$ (Memory Read) and $\overline{MEMW}$ (Memory Write) control signals. When the IO/$\overline{M}$ signal goes high, it indicates the peripheral I/O

*This code is an operation code (opcode) that instructs the microprocessor to perform the specified task. The term opcode is explained later.

FIGURE 3.5
Schematic to Generate Read/Write Control Signals for Memory and I/O



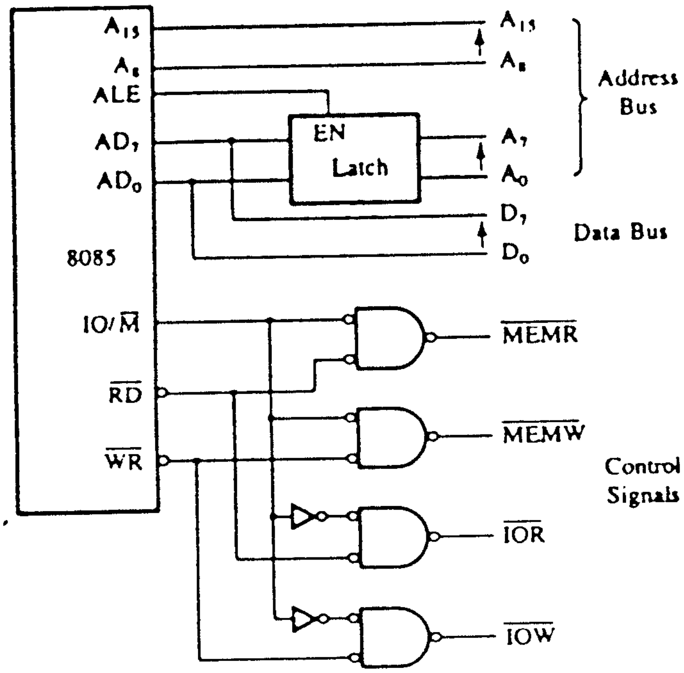
operation. Figure 3.5 shows that this signal is complemented using the Hex inverter 74LS04 and ANDed with the $\overline{RD}$ and $\overline{WR}$ signals to generate $\overline{IOR}$ (I/O Read) and $\overline{IOW}$ (I/O Write) control signals.

To demultiplex the bus and to generate the necessary control signals, the 8085 microprocessor requires a latch and logic gates to build the MPU, as shown in Figure 3.6. This MPU can be interfaced with any memory or I/O.

3.15 A Detailed Look at the 8085 MPU and Its Architecture

Figure 3.7 shows the internal architecture of the 8085 beyond the programmable registers we discussed previously. It includes the ALU (Arithmetic/Logic Unit), Timing and Con-

FIGURE 3.6
8085 Demultiplexed Address and Data Bus with Control Signals



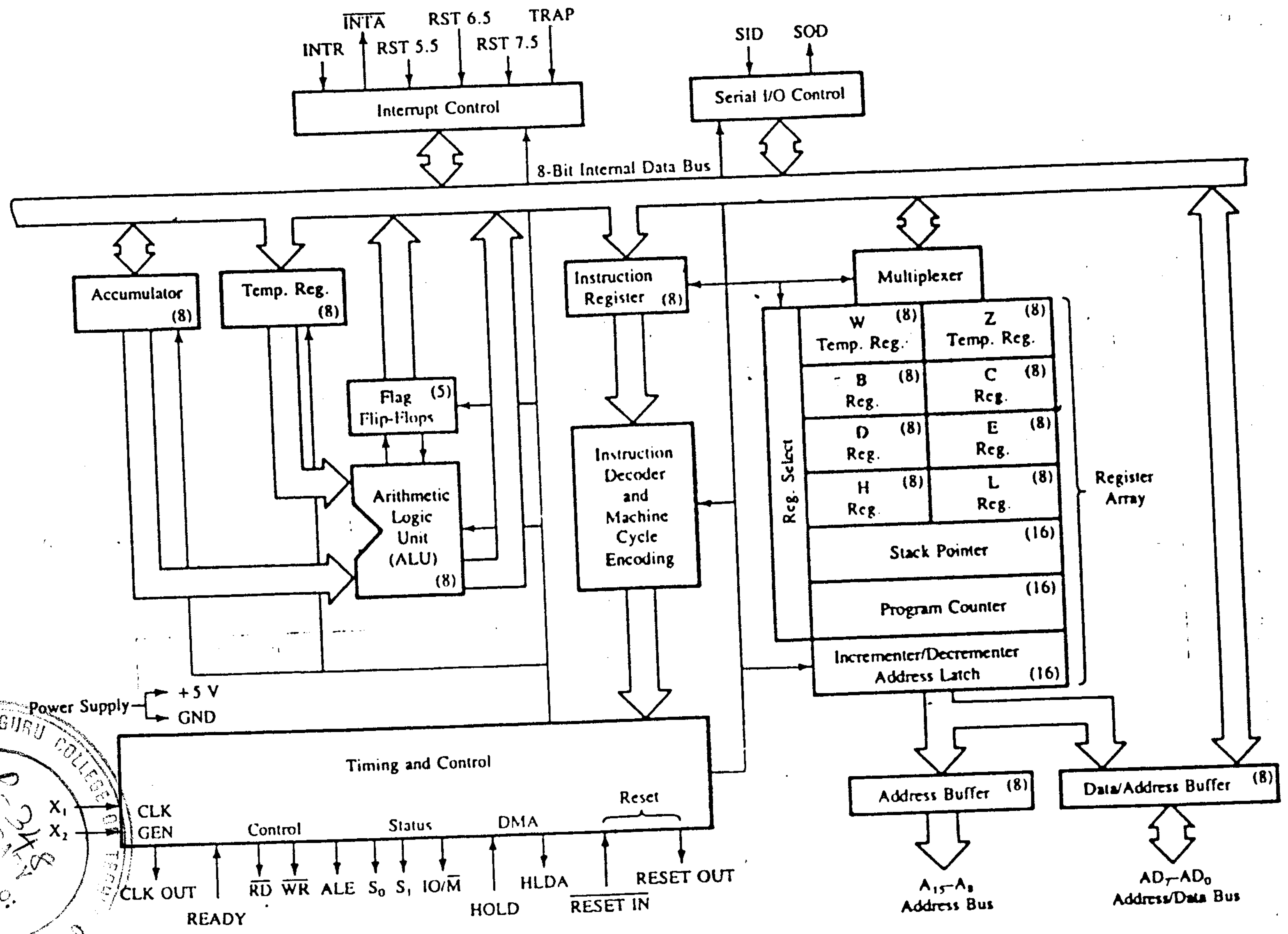


FIGURE 3.7

The 8085A Microprocessor: Functional Block Diagram

NOTE: The 8085A microprocessor is commonly known as the 8085.

SOURCE: Intel Corporation, *Embedded Microprocessors* (Santa Clara, Calif.: Author, 1994), p. 1-11 trol Unit, Instruction Register and Decoder, Register Array, Interrupt Control, and Serial I/O Control. We will discuss the first four units below; the last two will be discussed later in the book.

THE ALU

The arithmetic/logic unit performs the computing functions; it includes the accumulator, the temporary register, the arithmetic and logic circuits, and five flags. The temporary register is used to hold data during an arithmetic/logic operation. The result is stored in the accumulator, and the flags (flip-flops) are set or reset according to the result of the operation.

The flags are affected by the arithmetic and logic operations in the ALU. In most of these operations, the result is stored in the accumulator. Therefore, the flags generally reflect data conditions in the accumulator—with some exceptions. The descriptions and conditions of the flags are as follows:

□ **S—Sign flag:** After the execution of an arithmetic or logic operation, if bit D_7 of the result (usually in the accumulator) is 1, the Sign flag is set. This flag is used with signed numbers. In a given byte, if D_7 is 1, the number will be viewed as a negative number; if it is 0, the number will be considered positive. In arithmetic operations with signed numbers, bit D_7 is reserved for indicating the sign, and the remaining seven bits are used to represent the magnitude of a number. (See Appendix A2 for a discussion of signed numbers.)

□ **Z—Zero flag:** The Zero flag is set if the ALU operation results in 0, and the flag is reset if the result is not 0. This flag is modified by the results in the accumulator as well as in the other registers.

□ **AC—Auxiliary Carry flag:** In an arithmetic operation, when a carry is generated by digit D_3 and passed on to digit D_4, the AC flag is set. The flag is used only internally for BCD (binary-coded decimal) operations and is not available for the programmer to change the sequence of a program with a jump instruction.

□ **P—Parity flag:** After an arithmetic or logical operation, if the result has an even number of 1s, the flag is set. If it has odd number of 1s, the flag is reset. (For example, the data byte 0000 0011 has even parity even if the magnitude of the number is odd.)

□ **CY—Carry flag:** If an arithmetic operation results in a carry, the Carry flag is set; otherwise it is reset. The Carry flag also serves as a borrow flag for subtraction.

The bit positions reserved for these flags in the flag register are as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z		AC		P		CY

Among the five flags, the AC flag is used internally for BCD arithmetic; the instruction set does not include any conditional jump instructions based on the AC flag. Of the remaining four flags, the Z and CY flags are those most commonly used.