Chapter-4

Interfacing I/O Devices

4.1 BASIC INTERFACING CONCEPTS

An I/O device can be interfaced with the 8085 microprocessor either any of the following:

- Peripheral I/O or I/O mapped I/O or Port addressed I/O
- Memory mapped I/O

In peripheral I/O an 8-bit port is used to interface I/O with the processor. Intel 8085 processor uses peripheral I/O for data transfer. But, in memory mapped I/O a 16-bit memory register is used to interface I/O with the processor. Motorola 68000 processor uses memory mapped I/O for data transfer.

Peripheral I/O Instructions

- In peripheral I/O the 8085 microprocessor uses two instructions for data transfer between the processor and the I/O devices. The instructions are IN and OUT.

OUT Instruction:

An OUT instruction is used to give any output to an output device such as monitor, LED etc. The syntax of an OUT instruction is as follows:

OUT 8-bit Port Address

Mnemonic	Machine Code	Memory Address	Memory Contents
OUT 01H	D3	2050	11010011
	01	2051	00000001

4.1.2 I/O Execution

OUT INSTRUCTION

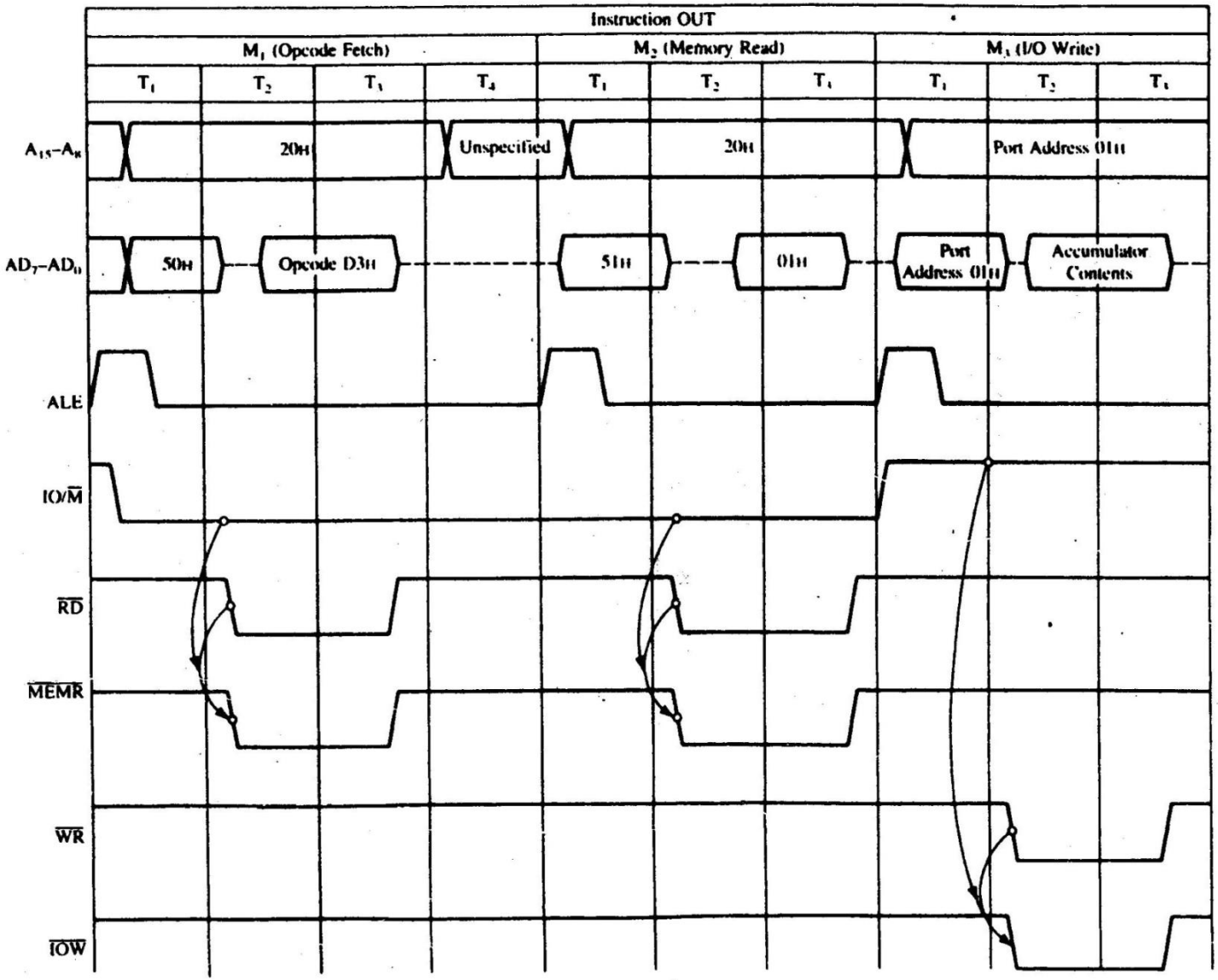


FIGURE 4.1
8085 Timing for Execution of OUT Instruction

IN Instruction: An IN instruction is used to take any input from an input device such as keyboard, switch etc. The syntax of an IN instruction is as follows:

IN 8-bit Port Address

Mnemonic	Machine Code	Memory Address	Memory Contents
IN 84H	DB	2065	11011011
	84	2066	10000100

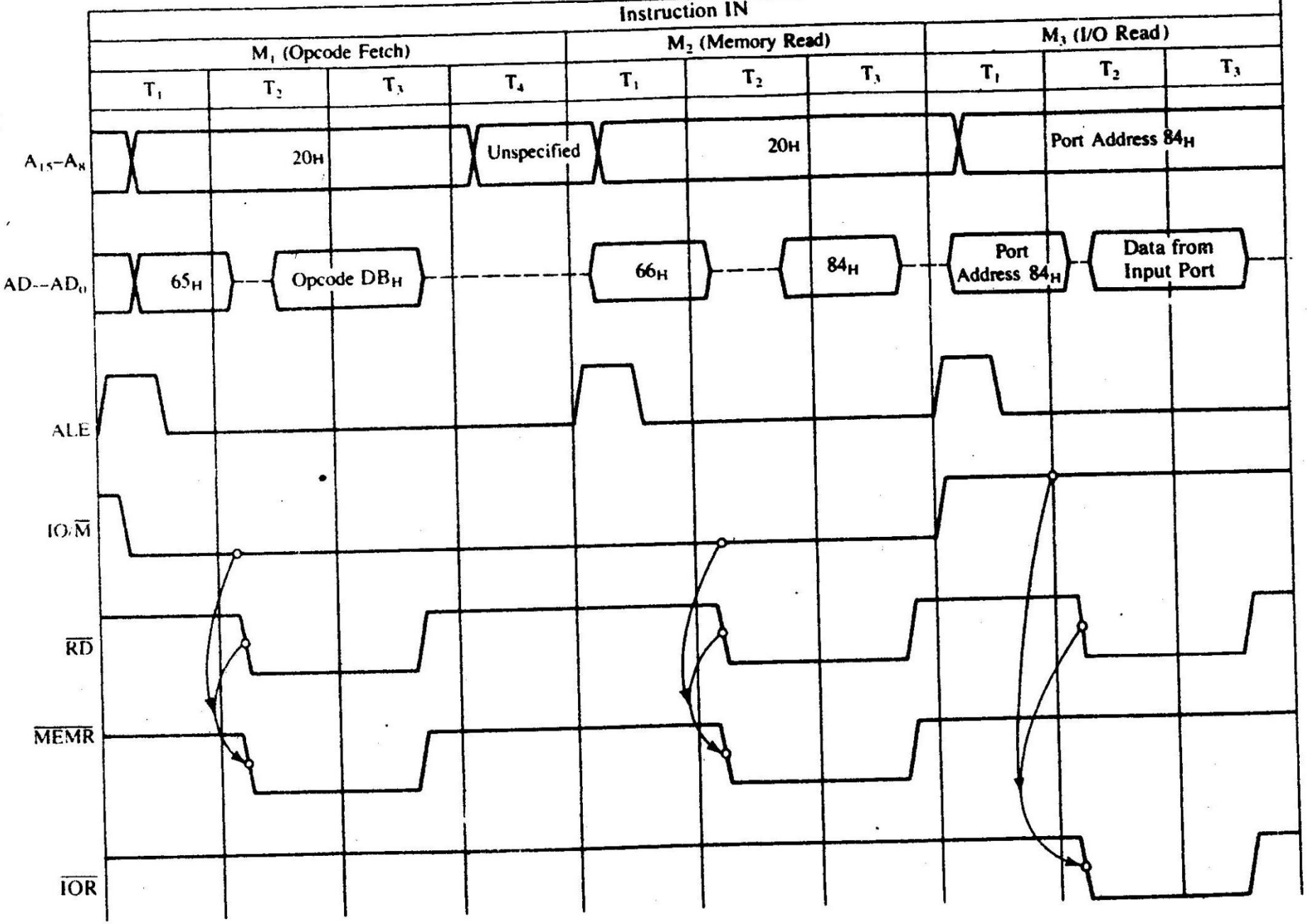


FIGURE 4.2
8085 Timing for Execution of IN Instruction

5.1.3 Device Selection and Data Transfer

Two problems regarding the I/O data transfer (M_3 cycle of OUT and IN instruction timing wave form):

(1) When should we enable the latch (or buffer for IN) to catch the information?
(2) What should be the address of that latch?

The solution of above problems are as follows:

1. Decode the address bus to generate a unique pulse called the **device address pulse** or **I/O address pulse**.
2. Combine (AND) the device address pulse with the control signal to generate a **device select pulse (I/O select pulse).**
3. Use the I/O select pulse to activate the interfacing device (I/O port).

Figure 4.3 shows a block diagram and Figure 4.4 shows an example of above solution.

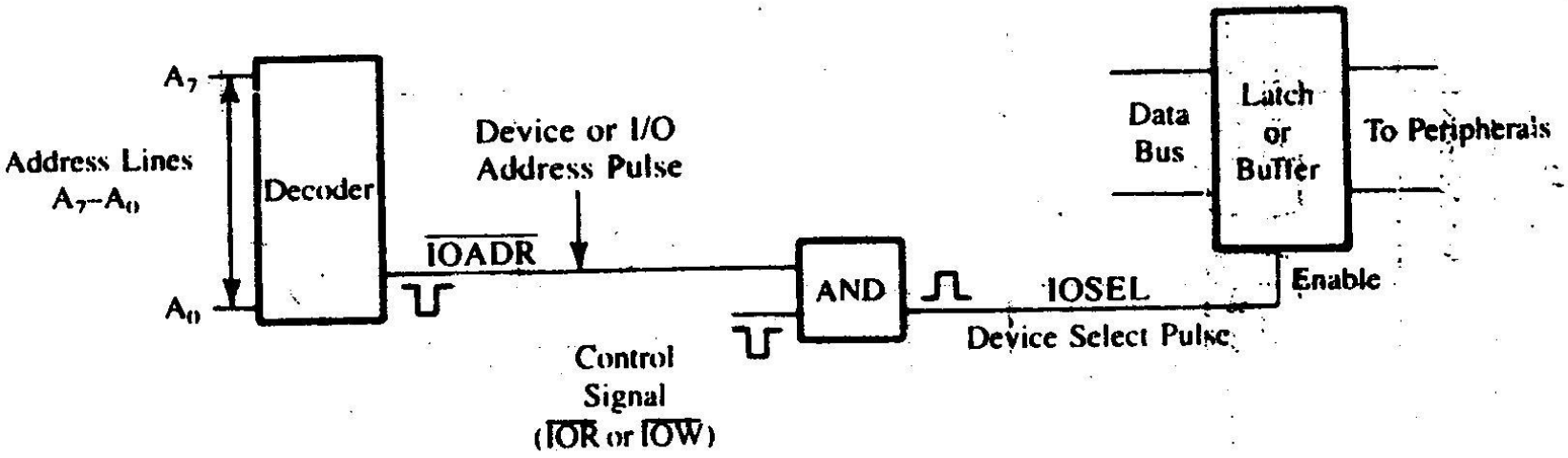


FIGURE 4.3
Block Diagram of I/O Interface

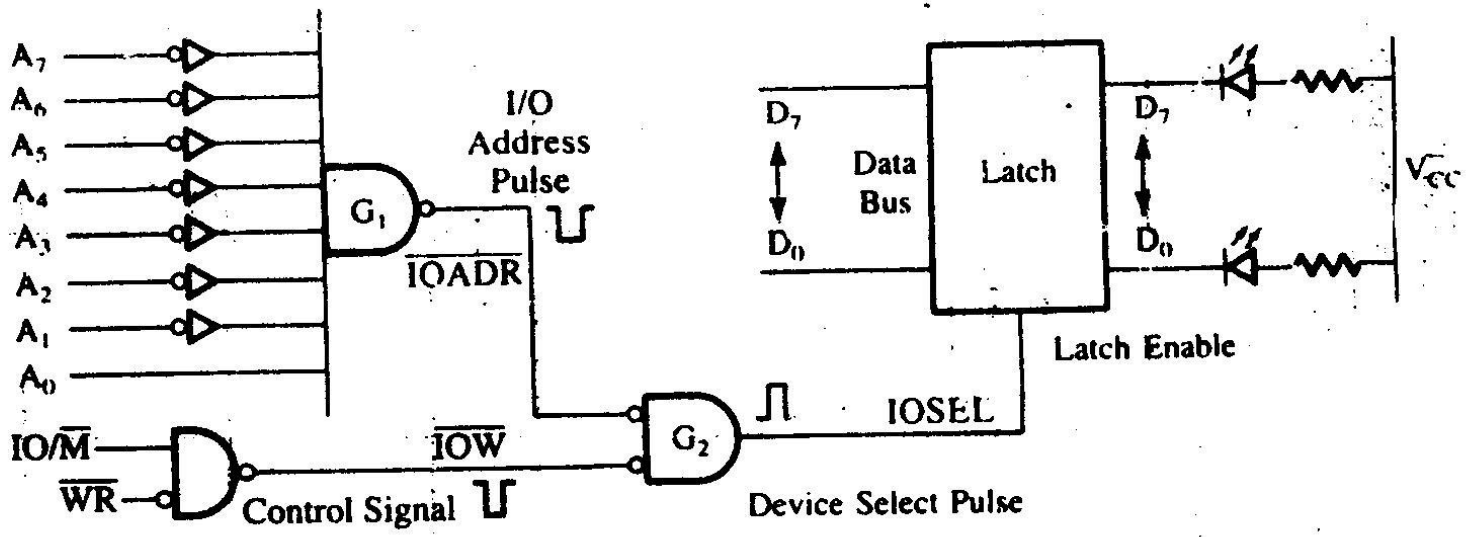


FIGURE 4.4
Decode Logic for LED Output Port

NOTE: To use this circuit with the 8085, the bus AD_7–A_0 must be demultiplexed.

4.14 Absolute Vs. Partial Decoding

- In Figure 4.4, all eight lines are decoded to generate one unique output pulse; the device will be selected only with the address, 01H. This is called **absolute decoding** and good design practice.

- However, to minimize the cost, the output port can be selected by decoding some of the address lines ,as shown in Figure 4.5; this is called **partial decoding**. As a result, the device has multiple addresses.

Figure 4.5 is similar to Figure 4.4 except that the address lines A0 and A1 are not connected, and they are replaced by IO/M and WR signals. Because the address lines A0 and A1 are at don't care logic level, they can be assumed 0 or 1. Thus this output port (latch) can be accessed by the addresses 00H, 01H, 02H, and 03H.

The partial decoding is commonly used technique in small systems. Such multiple addresses will not cause any problem, provided these addresses are not assigned to any other output ports.

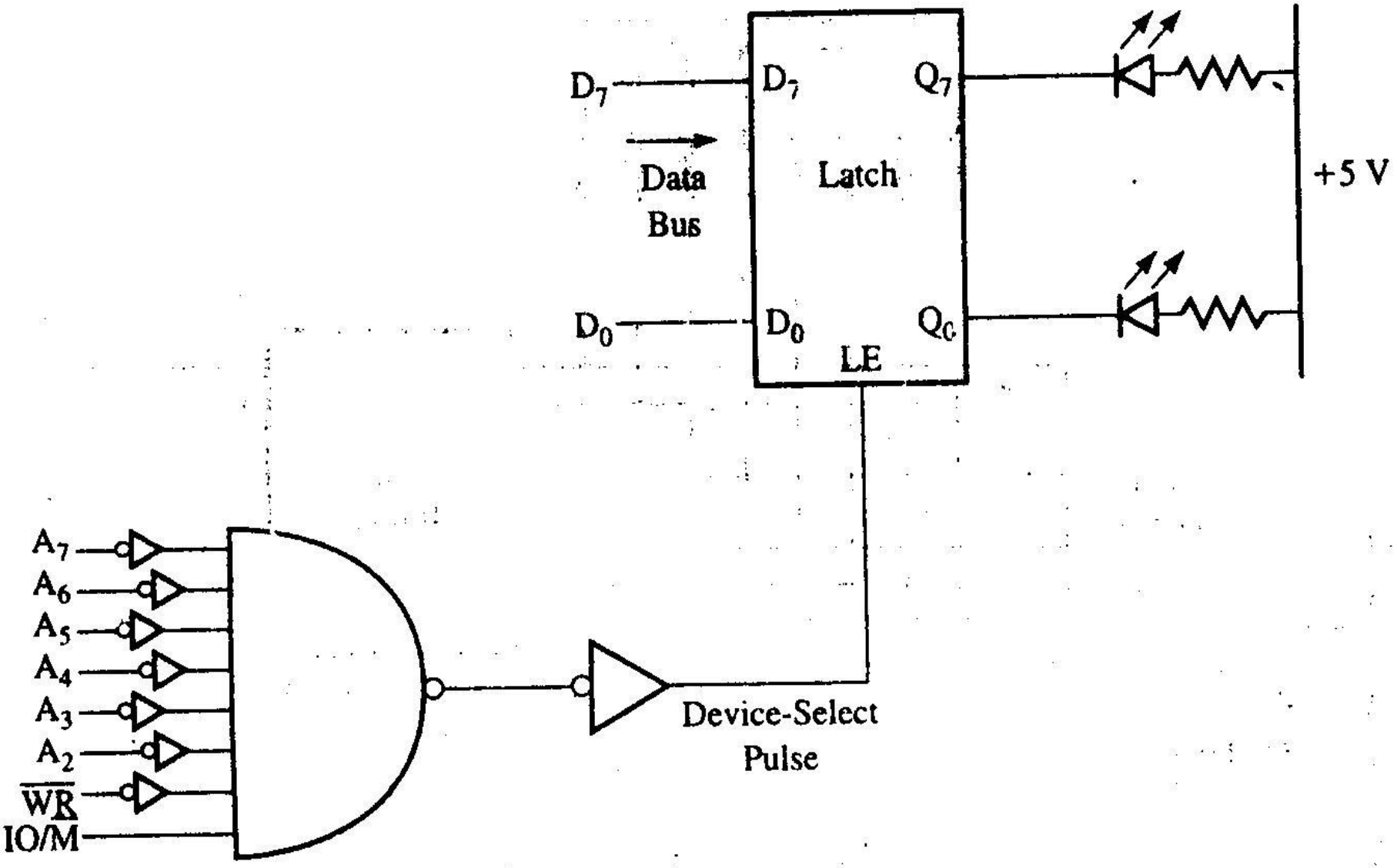


FIGURE 4.5
Partial Decoding: Output Latch with Multiple Addresses

4.15 Input Interfacing

Figure 4.6, shows an example of interfacing an 8-key input port. The circuit for the input port in Figure 4.6 differs from the output port in Figure 4.4 as follows:

1. Control signal $\overline{IOR}$ is used in place of $\overline{IOW}$.
2. The tri-state buffer is used as an interfacing port in place of the latch.
3. In Figure 4.6, data flow from the keys to the accumulator; on the other hand, in Figure 4.4, data flow from the accumulator to the LEDs.

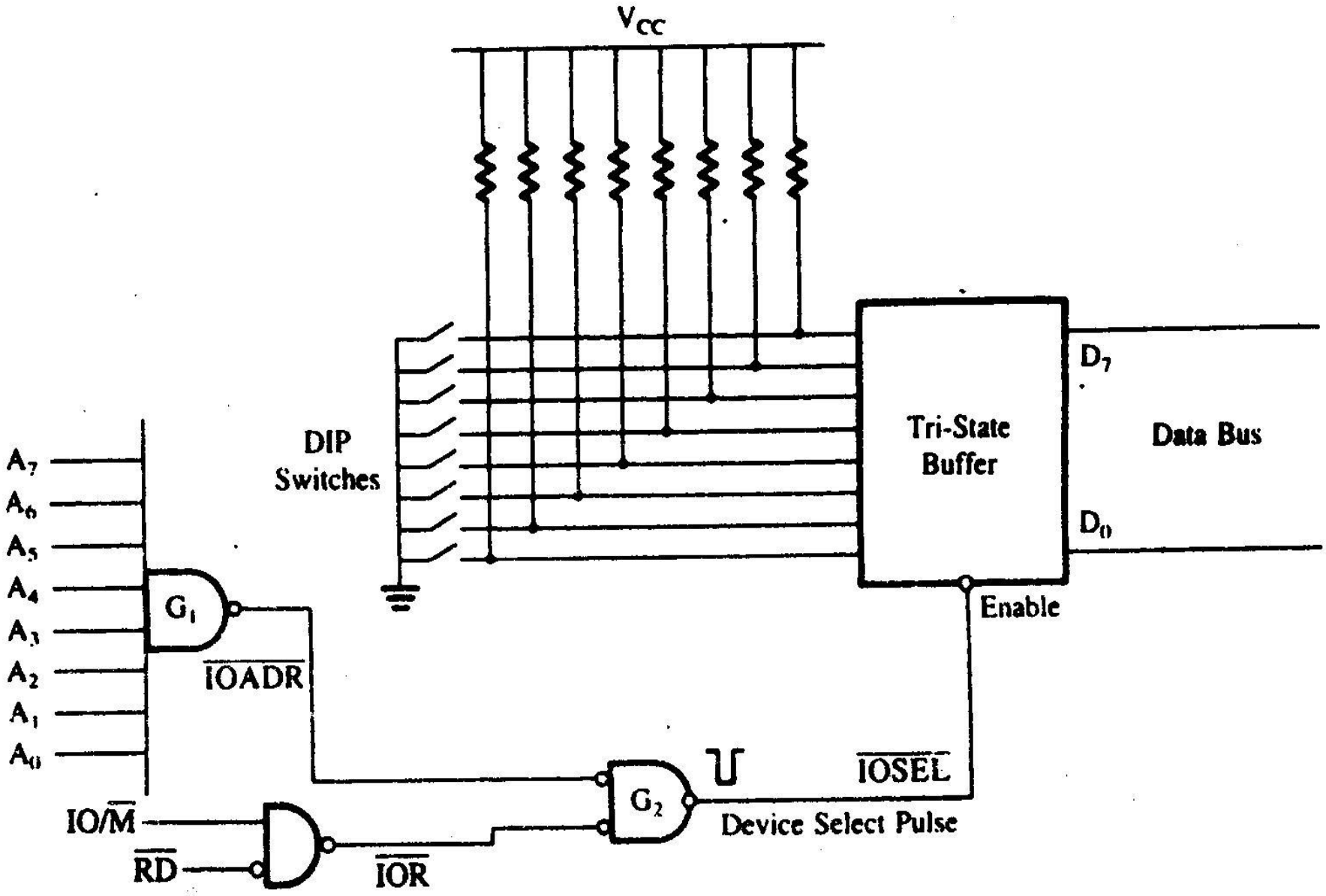


FIGURE 4.6
Decode Logic for a Dip-Switch Input Port

4.16 Interfacing I/Os Using Decoders

Figure 4.7 illustrates how a decoder is used to decode the address of input port and output port. To select or enable the tri-state buffer and latch respectively, following addresses are used to decoded by the decoder (Figure 4.7)

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0		
1	1	1	1	1	0	0	0	= F8H	(Latch)
1	1	1	1	1	0	1	0	= FAH	(Buffer)

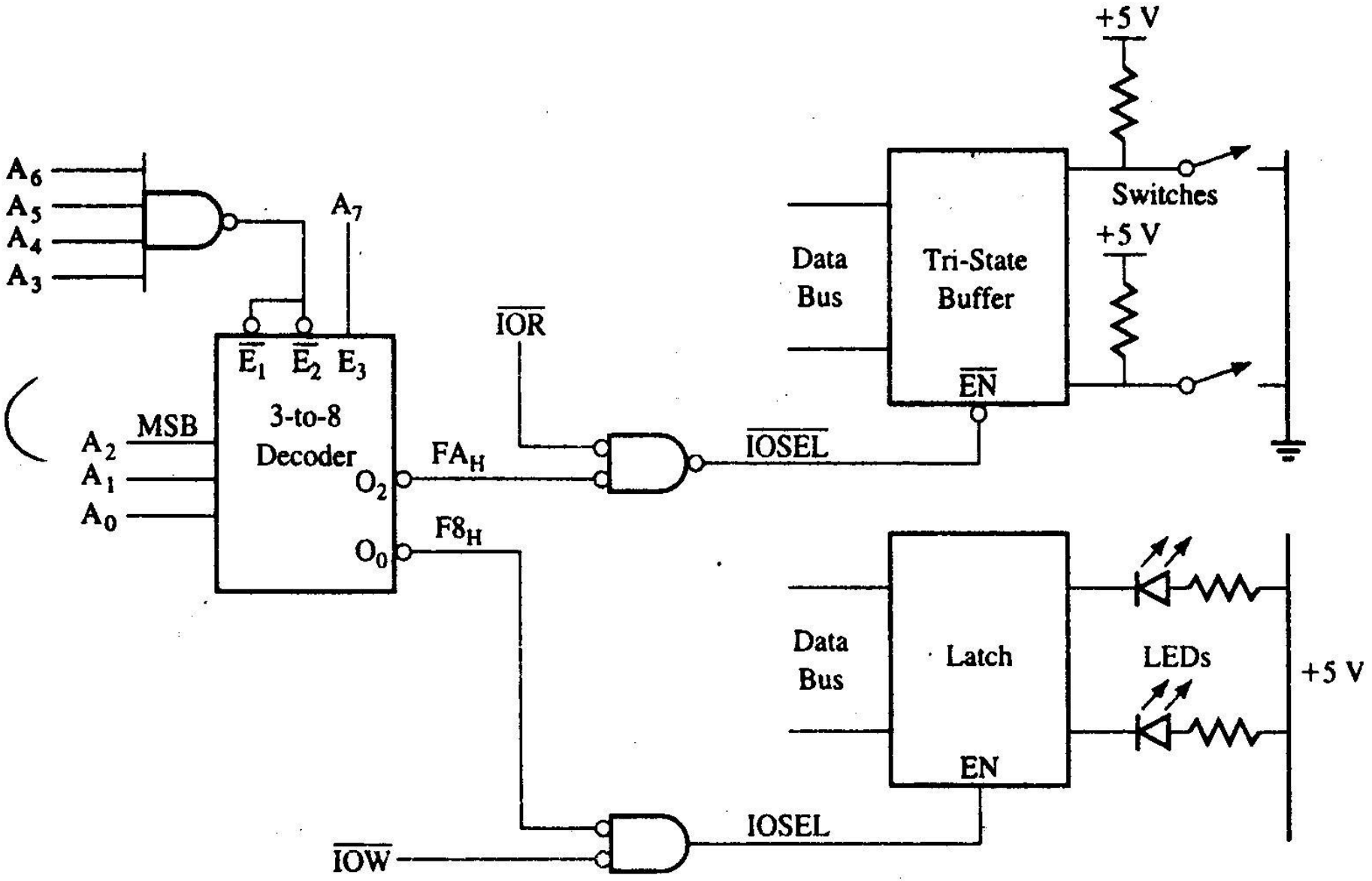


FIGURE 4.7
Address Decoding Using a 3-to-8 Decoder

4.2 INTERFACING OUTPUT DISPLAYS

4.22 Illustration: Seven-Segment LED Display as an Output Device

PROBLEM STATEMENT

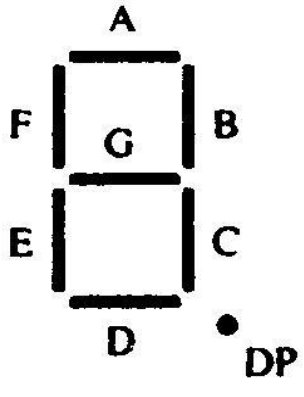
1. Design a seven-segment LED output port with the device address F5H, using a 74LS138 3-to-8 decoder, a 74LS20 4-input NAND gate, a 74LS02 NOR gate, and a common-anode seven-segment LED.
2. Given $\overline{WR}$ and IO/$\overline{M}$ signals from the 8085, generate the $\overline{IOW}$ control signal.
3. Explain the binary codes required to display 0 to F Hex digits at the seven-segment LED.
4. Write instructions to display digit 7 at the port.

HADWARE DESCRIPTION

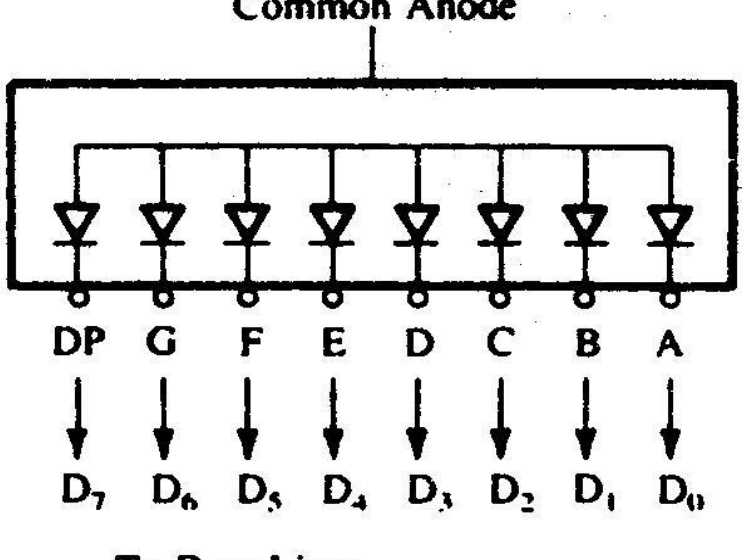
The circuit description is given below:

SEVEN-SEGMENT DISPLAY

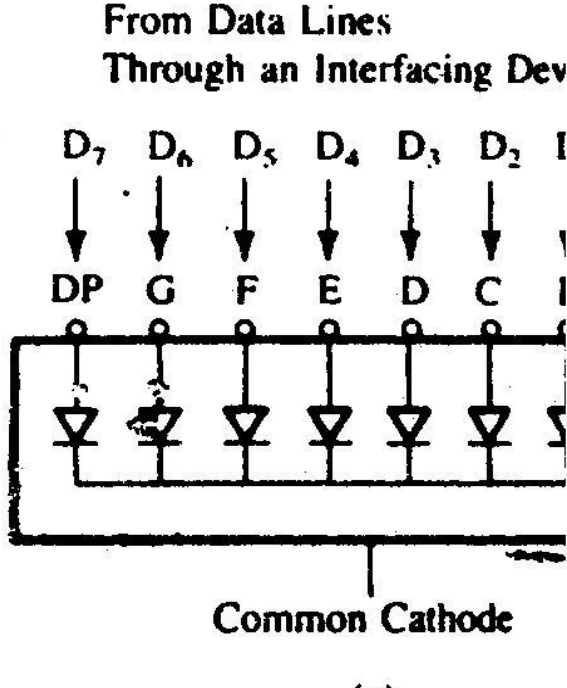
A seven-segment LED consists of seven light emitting diode and one segment for the decimal point. The LEDs are physically arranged as shown in Figure 4.9 (a). To display a number, the necessary segments are lit by sending an appropriate signals for current flow through diodes.



(a)



(b)



(c)

FIGURE 4.9

Seven-Segment LED: LED Segments (a); Common-Anode LED (b); Common-Cathod

To display digit 7 at the LED in Figure 4.10, the requirements are as follows:

1. It is a common-anode seven-segment LED, and logic 0 is required to turn on a segment.
2. To display digit 7, segments A, B, and C should be turned on.
3. The binary code should be

Data Lines	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
Bits	X	1	1	1	1	0	0	0	= 78H
Segments	NC	G	F	E	D	C	B	A	

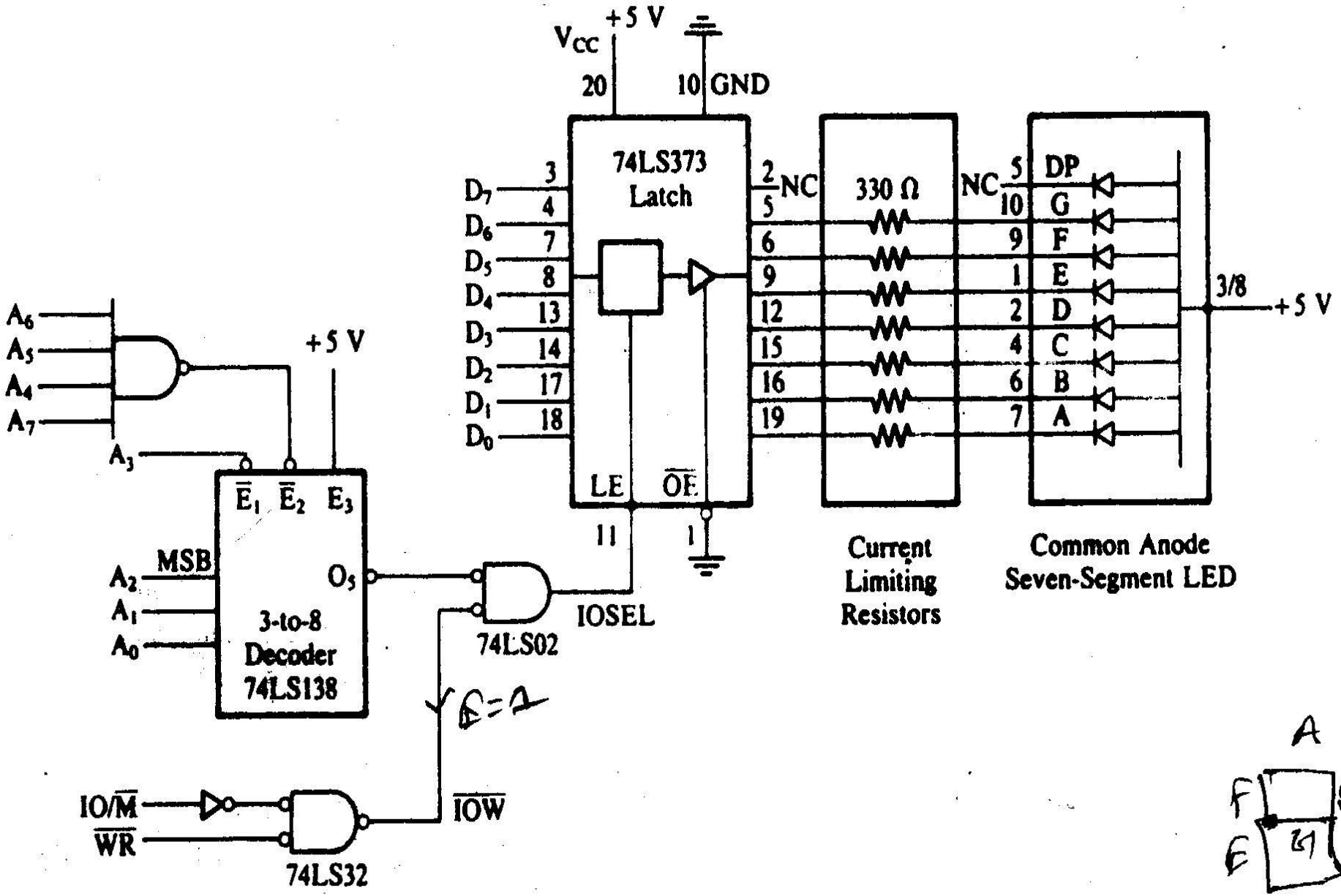


FIGURE 4.10
Interfacing Seven-Segment LED

7

To design an output port with the address F5H, the address lines A_7 – A_0 should have following logic:

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0		
1	1	1	1	0	1	0	1	=	F5H

Instructions:

The following instructions are necessary to display 7 at the output port:

```
MVI  A, 78H
OUT  F5H
HLT
```

4.3 INTERFACING INPUT DEVICES

4.31 Illustration: Data Input from DIP Switches

The circuit used for interfacing eight DIP switches, as shown in Figure 4.11. The circuit shows the 74LS138 3-to-8 decoder to decode the low-order bus and tri-state octal buffer (74LS244) to interfaces the switches to the data bus. The port can be accessed with the address 84H.

4.32 Hardware

Figure 4.11 shows the 74LS244 tri-state octal buffer used as an interfacing device. When $\overline{OE}$ signal goes low, the input data show up on the output lines (connected to the data bus).

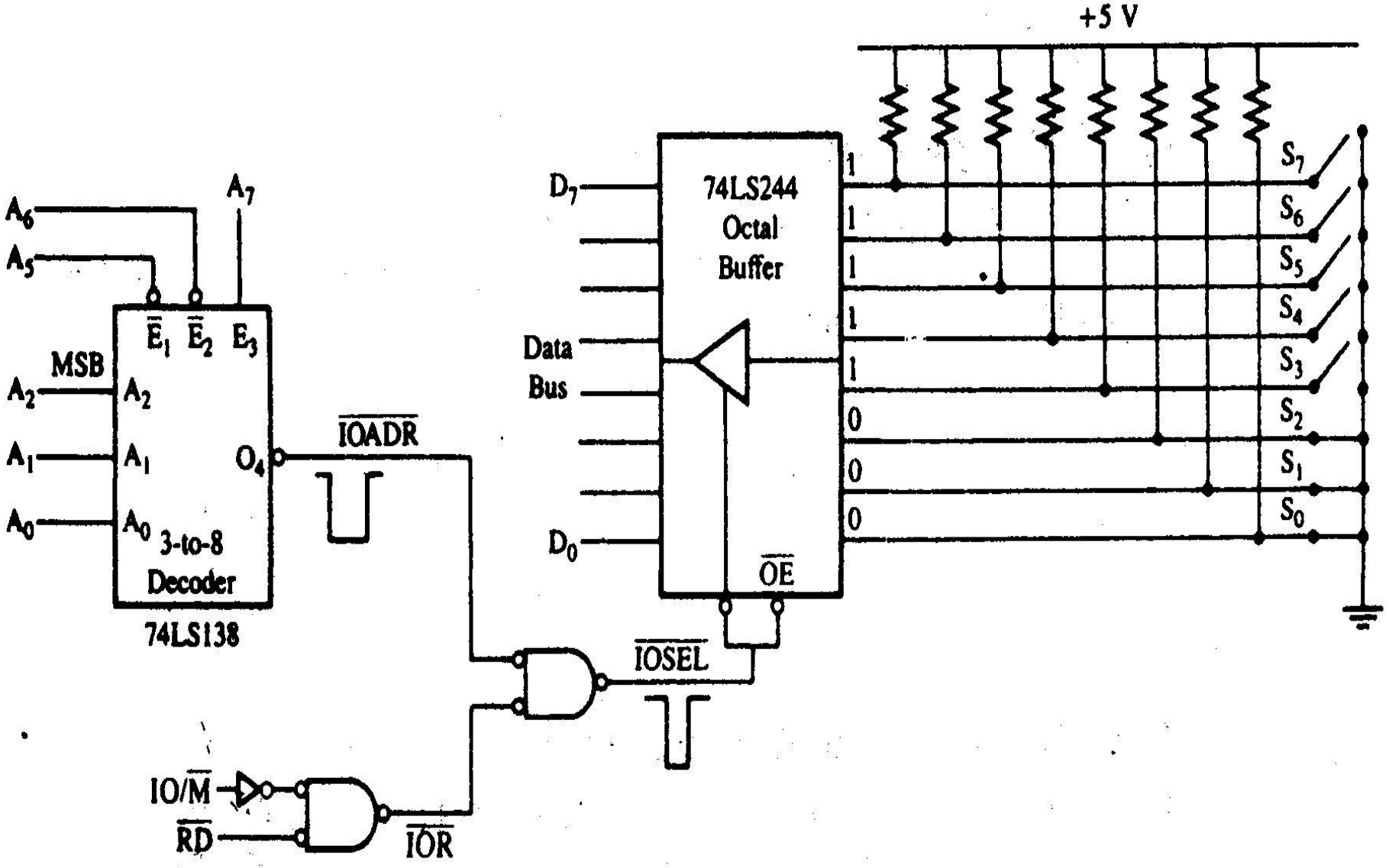


FIGURE 4.11
Interfacing DIP Switches

4.33 Interfacing Circuit

Figure 4.11 shows that the address lines A_4 and A_3 are left in the don't care state. The output line $\overline{OE}$ goes low when the address bus has the following address (assume the don't care lines are at logic 0):

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1	0	0	0	0	1	0	0	= 84H

4.34 Multiple Port Address

In Figure 4.11, the address lines A_4 and A_3 are not used by the decoding circuit; the logic levels on these lines can 0 or 1. Therefore, this input port can be accessed by four different addresses, as shown below.

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1	0	0	0	0	1	0	0	= 84H
			0	1				= 8CH
			1	0				= 94H
			1	1				= 9CH

4.35 Instructions to Read Input Port

To read data from the input port shown in Figure 4.11, the instructions are as follows:

```
IN 84H
HLT
```

When the instruction IN 84H is executed, during M_3 cycle, the 8085 places the address 84H on the low-order (as well as high-order bus), asserts the $\overline{RD}$ control signal, and reads the switch positions.

4.4 MEMORY- MAPPED I/O

In memory-mapped I/O, the input and output devices are assigned and identified by 16-bit addresses. To transfer data between the MPU and I/O devices, memory-related instructions (such as LDA, STA etc.) and memory control signals $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ are used.

To understand memory-mapped I/O technique, the following example is considered:

Memory Address	**Machine Code**	**Mnemonics**
2050	32	STA 8000H
2051	00	
2052	80	

The STA is a three-byte instructions.

In memory-mapped I/O technique both the memory and I/O devices uses the common domain of addressing. That is an input device or an output device can be connected to the memory location.

4.41 Execution of Memory-Related Data Transfer Instructions

To execute the STA instruction, 8085 microprocessor requires four machine cycles (M_1 to M_4) and thirteen T-states (Figure 4.12). The machine cycle M_4 for the STA instruction is similar to the machine cycle M_3 for the OUT instruction.

Device selection and data transfer in memory-mapped I/O require three steps that are similar to those required in peripheral I/O:

1. Decode the address bus to generate the device address pulse.
2. AND the control signal with the device address pulse to generate the device select pulse.
3. Use the device select pulse to enable the I/O port.

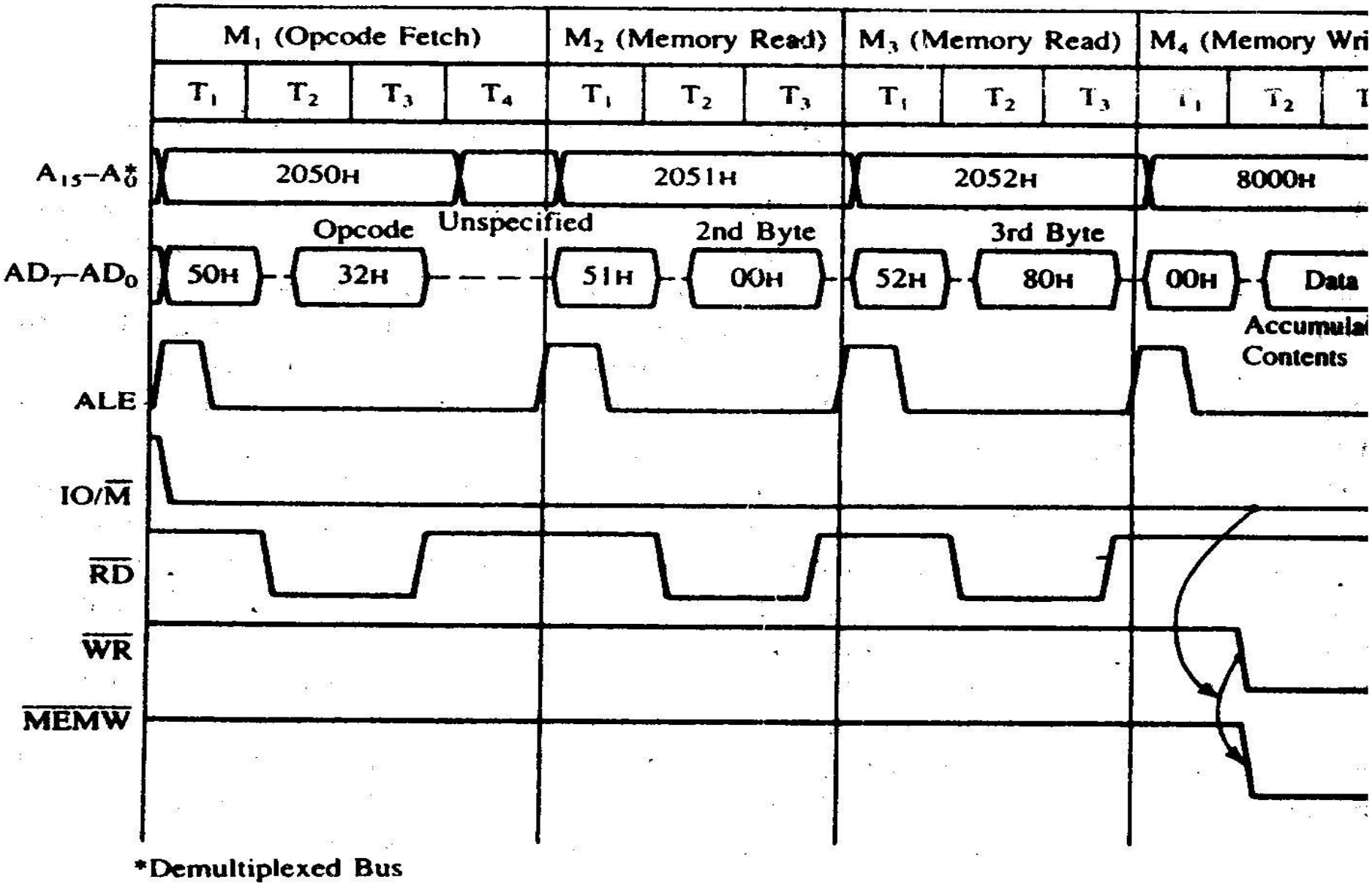


FIGURE 4.12
Timing for Execution of the Instruction: STA 8000H

4.42 Illustration: Safety Control System Using Memory-Mapped I/O Technique

Figure 4.13 shows a schematic of interfacing I/O devices using the memory-mapped I/O technique. The circuit includes one input port with eight DIP switches and one output port to control various processes and gates, which are turned on/off by the microprocessor according to the corresponding switch position.

OUTPUT PORT AND ITS ADDRESS

The various process control devices are connected to the data bus through the latch 74LS373 and solid state relays. When LE is high, the data enter the latch and when LE goes low, data are latched. The latched data are available on the output lines enabled by $\overline{OE}$. To assert the I/O select pulse, the output port address should be FFF8H, as shown below:

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	= FFF8H

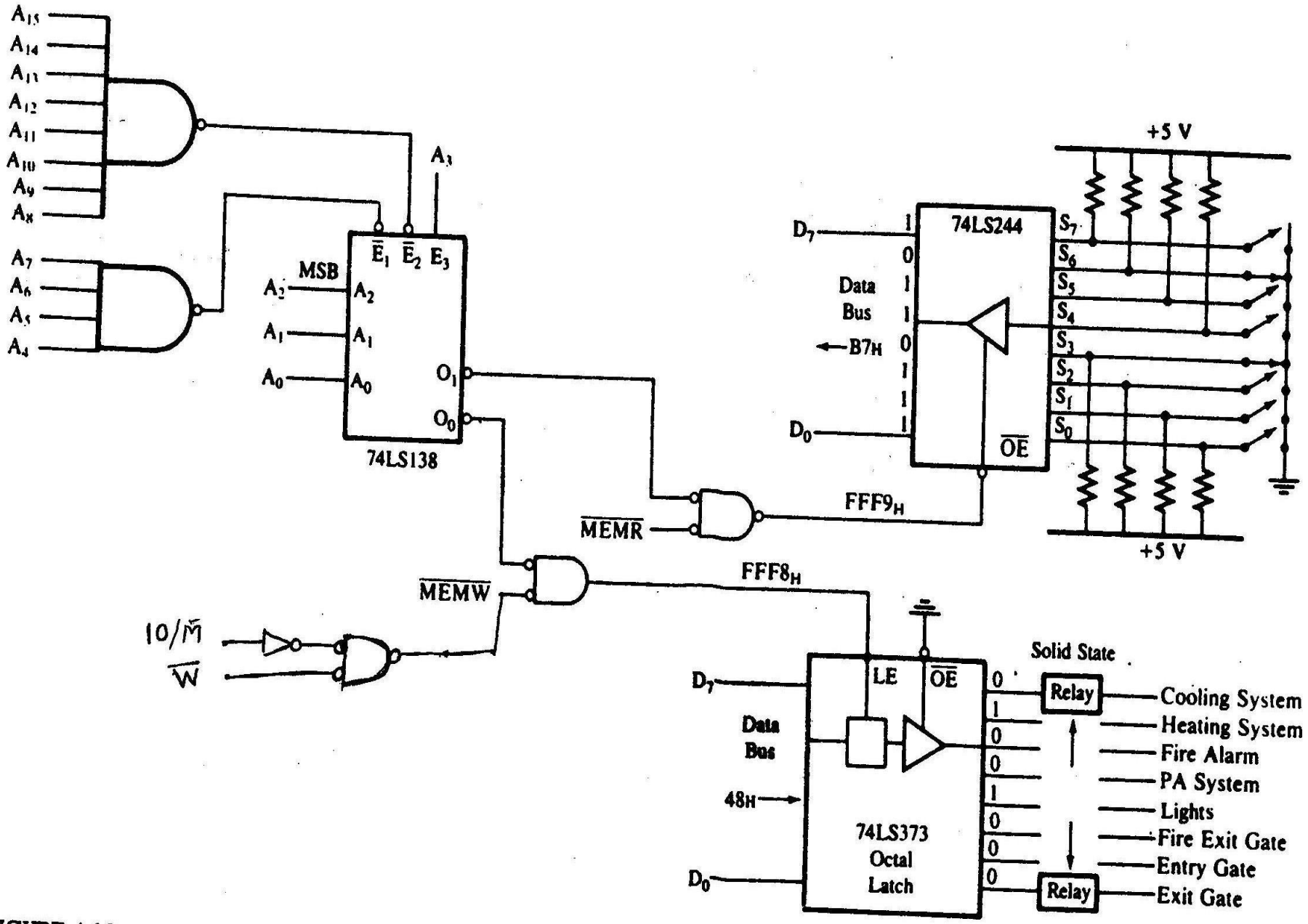


FIGURE 4.13
Memory-Mapped I/O Interfacing

INPUT PORT AND ITS ADDRESS

The DIP switches are interfaced with the 8085 using tri-state buffer 74LS244. The switch positions can be read by enabling the signal $\overline{OE}$. To read the input port, the port address should be

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	= FFF9H

Instruction:

To control the processes according to switch positions, the microprocessor should read the bit pattern at the input port and send that bit pattern to the output port. The following instructions can accomplish this task:

```
READ:  LDA  FFF9H
       CMA
       STA  FFF8H
       JMP  READ
```