Noida Institute of Engineering and Technology, Greater Noida

Microprocessor

(ACSE0405)

Unit: 4

Interfacing of I/O devices

B.Tech 4th Semester

Khushboo
Assistant Professor
ECE

Faculty Introduction

I received my Master of Technology from VIT Chennai and Bachelor of Technology from UPTU, Lucknow. I have worked as Software Developer in Cognizant Technology Solutions, Kolkata. Presently I am working as an Assistant Professor in Electronics & Communication Engineering department of Noida Institute of Engineering & Technology, Greater Noida, India. My research interest includes Antennas, IoT and Embedded System.

Evaluation Scheme

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
1	AAS0402	Engineering Mathematics-IV	3	1	0	30	20	50		100		150	4
2	AASL0401	Technical Communication	2	1	0	30	20	50		100		150	3
3	ACSE0405	Microprocessor	3	0	0	30	20	50		100		150	3
4	ACSE0403A	Operating Systems	3	0	0	30	20	50		100		150	3
5	ACSE0404	Theory of Automata and Formal Languages	3	0	0	30	20	50		100		150	3
6	ACSE0401	Design and Analysis of Algorithm	3	1	0	30	20	50		100		150	4
7	ACSE0455	Microprocessor Lab	0	0	2				25		25	50	1
8	ACSE0453A	Operating Systems Lab	0	0	2				25		25	50	1
9	ACSE0451	Design and Analysis of Algorithm Lab	0	0	2				25		25	50	1
10	ACSE0459	Mini Project using Open Technology	0	0	2				50			50	1
11	ANC0402 / ANC0401	Environmental Science*/ Cyber Security*(Non Credit)	2	0	0	30	20	50		50		100	0
12		MOOCs** (For B.Tech. Hons. Degree)											
		GRAND TOTAL										**1100**	**24**

Subject Syllabus

Course Contents / Syllabus		
UNIT-I	**8085 Microprocessor**	**8 Hours**
Introduction to Microprocessor, Microprocessor evolutionandtypes, Microprocessorarchitectureand its operation, Logic devices for interfacing,Pin diagram and internal architecture of 8085 Microprocessor,Example of an 8085 based computer, Instructionanddataflow,timerandtimingdiagram,interruptandmachinecycle, Addressingmodes.		
UNIT-II	**8085 Instructions and Programming Techniques**	**8 Hours**
Instructionsets,InstructionClassification:datatransfer operations, arithmeticoperations,logicaloperations,branchingoperations,machinecontrolandassemblerdirectives, writing assembly language programs, Programming techniques: looping, counting and indexing		
UNIT-III	**Code Conversion and BCD Arithmetic**	**8 Hours**
Counter and time delays, Illustrative program: Hexadecimal counter, zero-to-nine, (module ten) counter, generating pulse waveforms, Stack, Subroutine, Restart, Conditional call and return instructions, Advance subroutine concepts, Program: BCD-to-Binary conversion, Binary-to-BCD conversion, BCD-to-Seven segment code converter, Binary-to-ASCII and ASCII-to-Binary code conversion, BCD Addition, BCD Subtraction, Introduction to Advance instructions and Application, Multiplication		
UNIT-IV	**Interfacing of I/O devices**	**8 Hours**
Basic interfacing concepts,Memoryinterfacing,Interfacing output displays, Interfacing input devices, Memory mapped I/O,Interfacing keyboard and seven segment displays, The 8085 Interrupts, 8085 vector interrupts,8259 programmable interrupt controller,		
UNIT-V	**Programmable Peripheral IC's and 8086 Microprocessor**	**8 Hours**
Peripheral Devices: 8255 programmable peripheral interface,8253/8254 programmable timer/counter, 8237 DMA Controller, 8251 USART and RS232C.Introduction to 8086 microprocessors: Architecture of 8086 (Pin diagram, Functional block diagram, register organization), Addressing Modes		

Branch Wise Application

- **Microprocessor**, any of a type of miniature electronic device that contains the arithmetic, logic, and control circuitry necessary to perform the functions of a digital computer's central processing unit.
- This kind of integrated circuit can interpret and execute program instructions as well as handle arithmetic operations.

Course Objective

- The objective of this course is to understand basic concepts of Microprocessor based systems and able to do programming in Assembly Language of 8085. They will be able to learn and program Peripheral IC's.

Course Outcome

At the end of this course students will able to:

- Apply a basic concept of digital fundamentals to Microprocessor based personal computer system.
- Analyze a detailed s/w & h/w structure of the Microprocessor.
- Illustrate how the different peripherals (8085/8086) are interfaced with Microprocessor.
- Analyze the properties of Microprocessors (8085/8086).
- Evaluate the data transfer information through serial & parallel ports.

Program Outcomes

- **Program Outcomes** are narrow statements that describe what the students are expected to know and would be able to do upon the graduation.

- These relate to the skills, knowledge, and behavior that students acquire through the programmed.

1. Engineering knowledge
2. Problem analysis
3. Design/development of solutions
4. Conduct investigations of complex problems
5. Modern tool usage
6. The engineer and society
7. Environment and sustainability
8. Ethics
9. Individual and team work
10. Communication
11. Project management and finance
12. Life-long learning

CO-PO Mapping

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ACSE0405.1	3	1	2		1	-	-	-	-	-	-	3
ACSE0405.2	3	1	-		1	-	-	-	-	-	-	2
ACSE0405.3	3	1	3		1	-	-	-	-	-	-	2
ACSE0405.4	3	1	-		1	-	-	-	-	-	-	2
ACSE0405.5	3	1	2		1	-	-	-	-	-	-	2
Average	3	1	2.3		1	-	-	-	-	-	-	2.1

PSOs Mapping

Sr. No.	Course Outcome	PSO1	PSO2	PSO3
1	**ACSE0405.1**	3	2	1
2	**ACSE0405.2**	3	2	1
3	**ACSE0405.3**	3	2	1
4	**ACSE0405.4**	3	2	1
5	**ACSE0405.5**	3	2	2
	Average	**3**	**2**	**1.2**

End Semester Question Paper Template

B.Tech (Semester IV Theory Examination 2021-22)			**Total Marks : 100**
Note: Attempt all sections. If require any missing data, then choose suitably. **Time: 3 hours**			
Section A			
1. Attempt all questions in brief.		2 X 10 = 20	
Q. No.	Question	Marks	CO
a. to j		2	
Section B			
2. Attempt any three of the following		3 X 10 = 30	
Q. No.	Question	Marks	CO
a to e		10	
Section C			
Question no. 3,4,5,6,7. Attempt any one of the following		1 X 10 = 10	
a		10	
b		10	

Prerequisites

- Basic knowledge of digital logic gates
- Knowledge of Boolean Algebra
- Basic concept of Flip-Flops

Unit Content

- Course Objective
- Unit Objective
- Course Outcome
- CO and PO Mapping
- Topic Objective
- Prerequisite
- Basic interfacing concepts
- Memory interfacing
- Interfacing output displays
- Interfacing input devices
- Memory mapped I/O
- Interfacing keyboard and seven segment displays
- The 8085 Interrupts
- 8085 vector interrupts
- 8259 programmable interrupt controller

Unit Objective

1. To study the Introduction of microprocessors.
2. To understand Memory and I/O devices.
3. To learn microprocessor 8085 architecture.
4. To discuss concept of Timing Diagram.
5. To acquire the knowledge of Addressing Modes.

Topic-1 Objective

Name of the Topic	Objective of the topic	Mapping with CO
Basic interfacing concepts	To understand the concept of Memory interfacing	CO4

Basic interfacing concepts

Interface is the path for communication between two components. Interfacing is of two types, memory interfacing and I/O interfacing.

IO devices can be interfaced:

- **Memory-Mapped I/O** (using addresses from memory space)
  - Device is identified by 16-bit address (Space ranges from 0000H – FFFFH

- **Standard I/O mapped or isolated I/O mapping /Peripheral Mapped I/O** has separate numbering scheme for I/O devices
  - Instructions IN/OUT are used for data transfer
  - Device is identified by 8-bit address (Space ranges from 00H – FFH)

Basic interfacing concepts

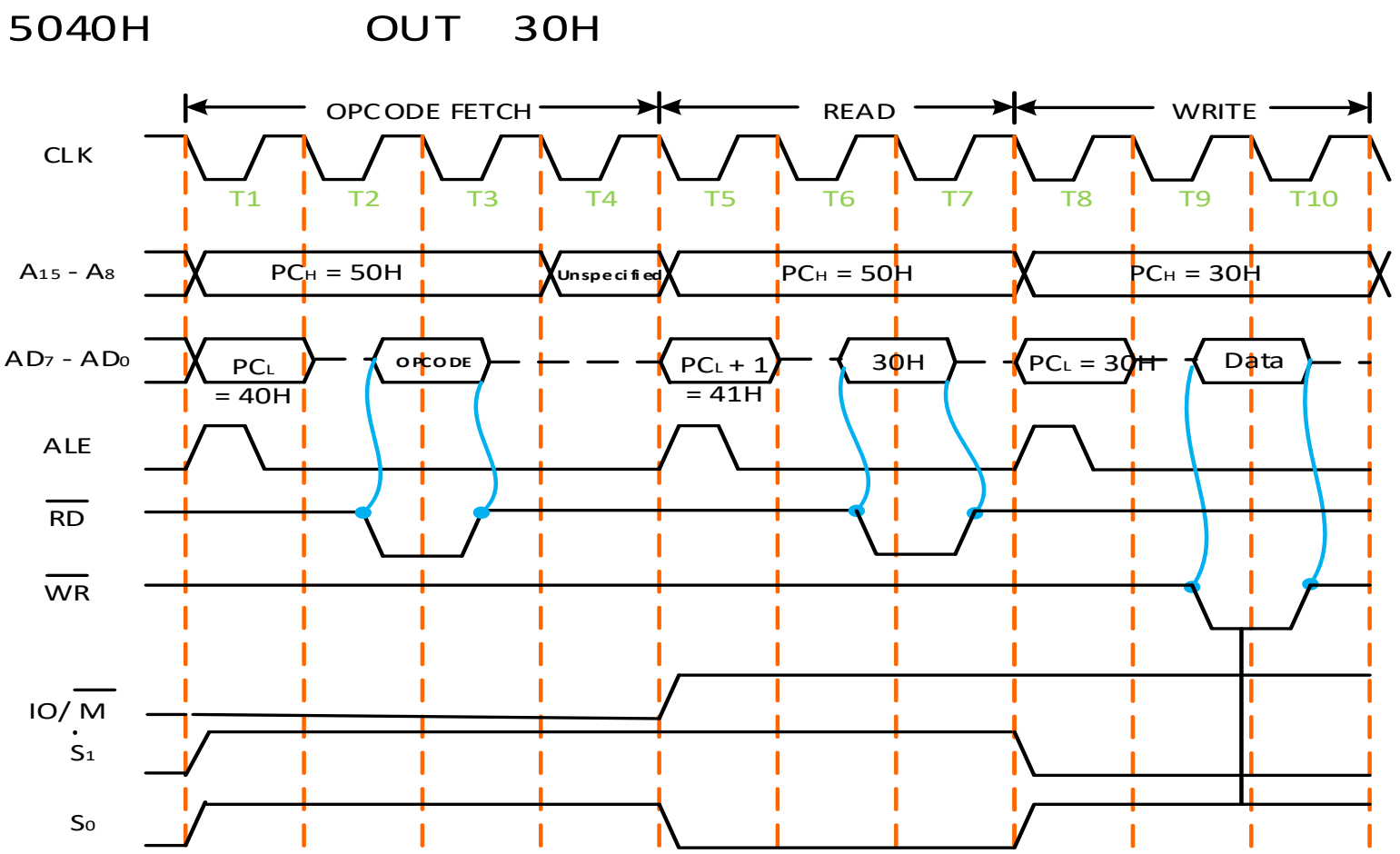
- Instruction IN, inputs data from an input device into accumulator.

  Example: 2065 IN 84H

- Instruction OUT, sends the content of the accumulator to output device such as LED display.

  Example: 2050 OUT 01

Basic interfacing concepts

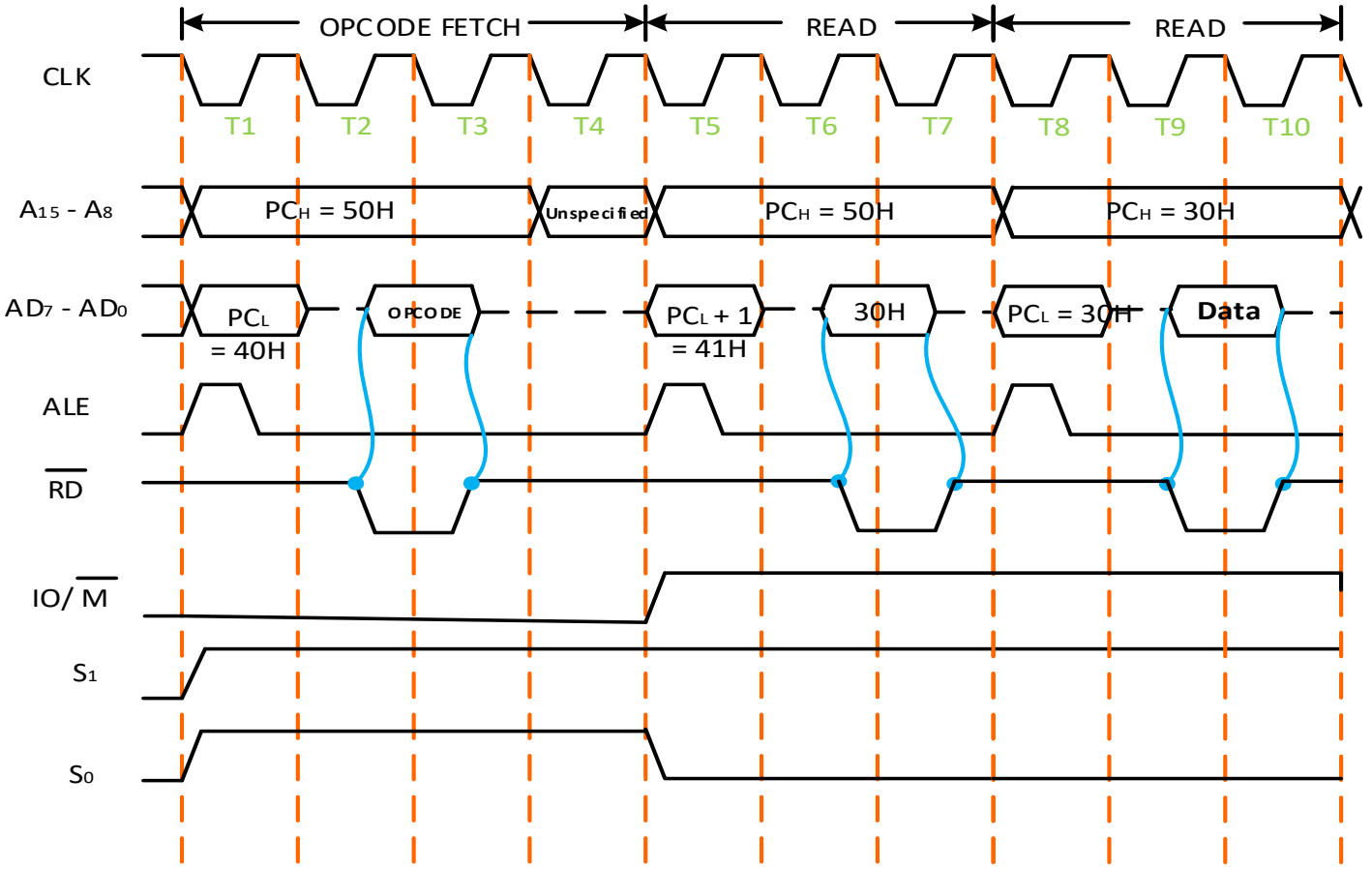


Basic interfacing concepts

- In First Machine cycle M_1(Opcode Fetch), microprocessor places the 16-bit memory address from the program counter (PC) on the address bus. At T1 50H is placed on A15-A8 and 40H is placed on AD7-AD0. ALE goes high, IO/$\overline{M}$ goes low indicates memory related operations. ALE indicates the availability of the address on AD7-AD0. At T2 microprocessor sends $\overline{RD}$ control signal which is combined with IO/$\overline{M}$ to generate $\overline{MEMR}$ signal and processor fetches the instruction code using data bus.
- M2 (memory Read), 8085 places next address 5041H on address bus and get device address 30H.
- M3 (I/O write), 8085 place device address 30H on low and high address bus both.IO/$\overline{M}$ goes high to indicate I/O operation. Information necessary for interfacing output device is available during T2 and T3 of the M3 cycle.

Basic interfacing concepts

5040H IN 30H



Basic interfacing concepts

- In First Machine cycle M_1(Opcode Fetch), microprocessor places the 16-bit memory address from the program counter (PC) on the address bus. At T1 50H is placed on A15-A8 and 40H is placed on AD7-AD0. ALE goes high, IO/$\overline{M}$ goes low indicates memory related operations. ALE indicates the availability of the address on AD7-AD0. At T2 microprocessor sends $\overline{RD}$ control signal which is combined with IO/$\overline{M}$ to generate $\overline{MEMR}$ signal and processor fetches the instruction code using data bus.
- M2 (memory Read), 8085 places next address 5041H on address bus and get device address 30H.
- M3 (I/O write), 8085 place device address 30H on low and high address bus both.IO/$\overline{M}$ goes high to indicate I/O operation. Information necessary for interfacing input device is available during T2 and T3 of the M3 cycle.

Data Transfer

- **For data transfer from input device to processor the following operations are performed.**
  - The input device will load the data to the port.
  - When the port receives a data, it sends message to the processor to read the data.
  - The processor will read the data from the port.
  - After a data have been read by the processor the input device will load the next data into the port.
- **For data transfer from processor to output device the following operations are performed.**
  - The processor will load the data to the port.
  - The port will send a message to the output device to read the data.
  - The output device will read the data from the port.
  - After the data have been read by the output device the processor can load the next data to the port.

Device Selection and Data Transfer

Steps are summarized as:

•Decode the address bus to generate unique pulse corresponding to device address on the bus called device address bus or I/O address pulse.

•Combine the device address pulse with the control signal to generate a device select pulse(I/O select) that is generated only when both signals are asserted.

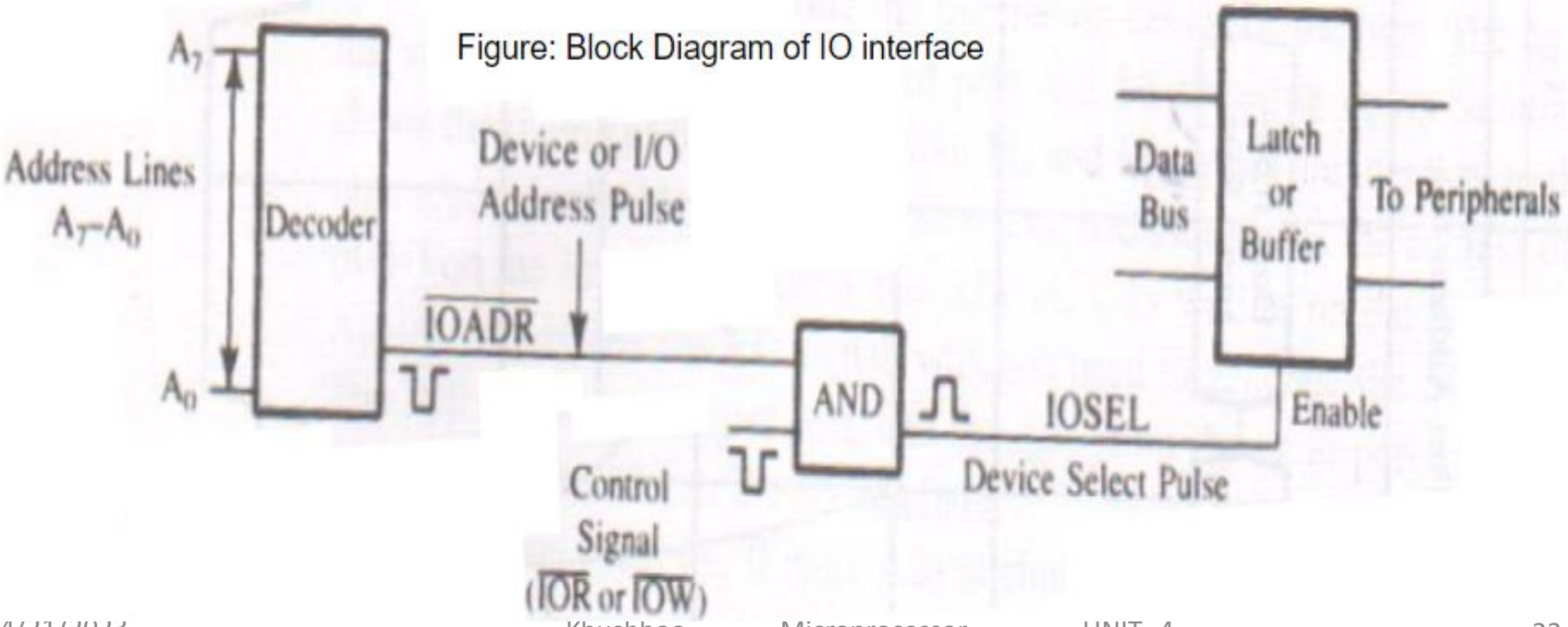


Figure: Block Diagram of IO interface

Daily Quiz

1. What is Microprocessor?
   a) A multipurpose PLD that accepts binary data as input
   b) A multipurpose PLD that accepts an integer as input
   c) A multipurpose PLD that accepts whole numbers as input
   d) A multipurpose PLD that accepts prime numbers as input
2. Which of the following is a type of microprocessor?
   a) CISC
   b) RISC
   c) EPIC
   d) All of the mentioned

Daily Quiz

3. The microprocessor of a computer can operate on any information if it is present in _____ only.
   a) Program Counter
   b) Flag
   c) Main Memory
   d) Secondary Memory
4. Which of the following technology was used by Intel to design its first 8-bit microprocessor?
   a) NMOS
   b) HMOS
   c) PMOS
   d) TTL
5. Which of the following addressing method does the instruction, MOV AX,[BX] represent?
   a) **register indirect addressing mode**
   b) direct addressing mode
   c) register addressing mode

Weekly Assignment

1. Define Microprocessor.
2. Define interface.
3. Explain Timing Diagram.
4. Define Bus.
5. Define Wire.
6. Define assert.
7. Define deassert.
8. What do you mean by data direction?

Topic Links

https://www.youtube.com/watch?v=Xl2nWDcy0To

https://slideplayer.com/slide/3944912/

https://www.bu.edu.eg/portal/uploads/Engineering,%20Shoubra/Electrical%20Engineering/3515/crs-14315/Files/Lec%201%20Intro%20to%20mp.ppt

Topic-2 Objectives

Name of the Topic	Objective of the topic	Mapping with CO
Memory interfacing	To understand the concept of memory interfacing	CO4

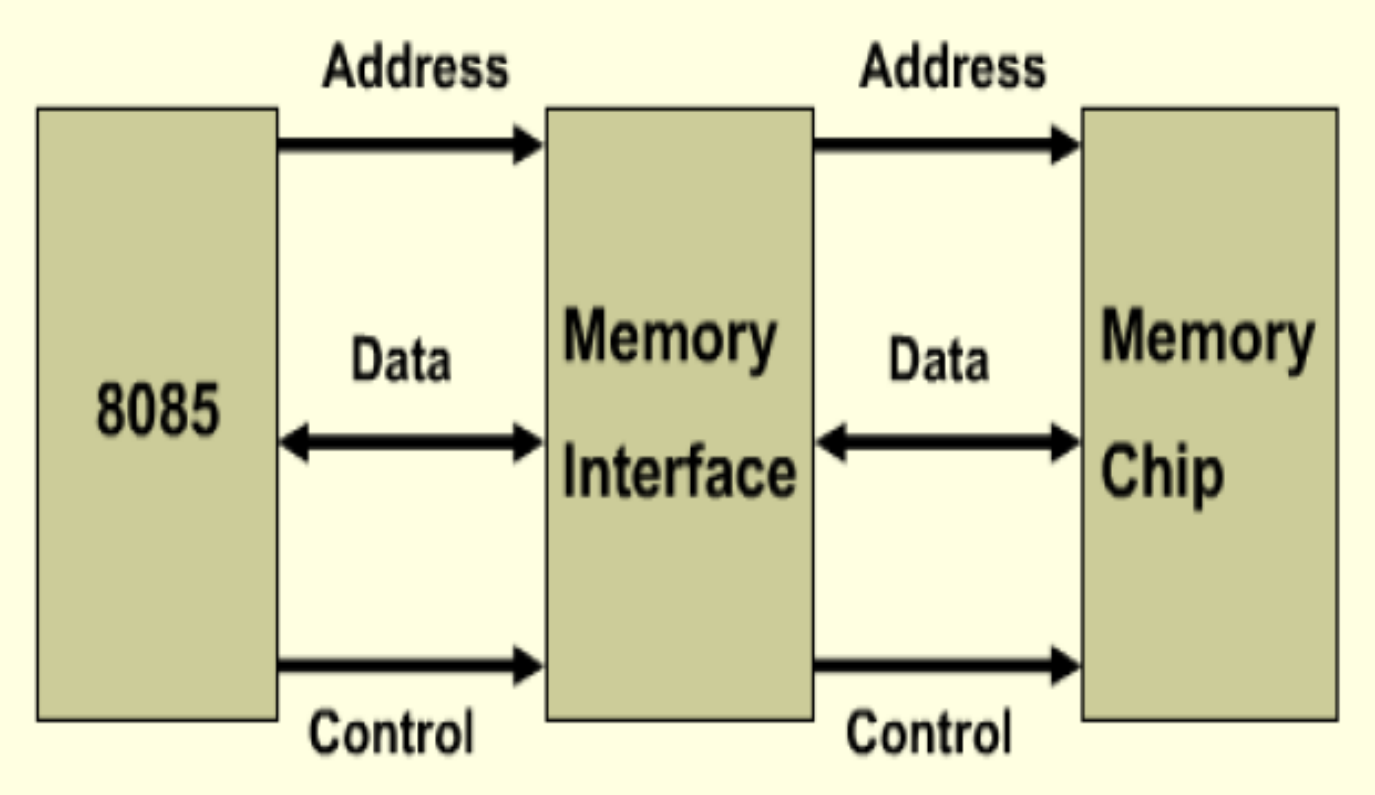
Memory interfacing

8085 interfacing with memory Chips

- Microprocessor needs to access memory quite frequently to read instructions and data stored in memory; the interface circuit enables that access.

- The interface process involves designing a circuit that will match the memory requirements with the microprocessor signal**.**

Memory interfacing



Memory interfacing

Memory Based Problem

1. Calculate the address lines required for an 8K Byte memory chip.

No of memory location = 2^n

n= no of address lines

Memory interfacing

Memory Based Problem

2. Calculate the no of chips needed to design 8K byte memory if the memory chip size is 1024*1.

3. How many bits are stored by a 256*4 memory chip? Can this chip be specified as 128 Byte memory?

4. If the memory chip size is 1024*4 bits, how many chips are required to make-up 16 K Byte memory?

5. If the memory chip size is 1024*4 bits, how many chips are required to make-up 2K Byte of memory?

Memory interfacing

Memory Based Problem

6. The memory map of a 4K byte memory chip begins at location 2000H. Specify the address of the last location on the chip. And also find no of pages in the chip.

Page size = 256 Bytes

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

Memory interfacing

Memory Based Problem

7. The memory address of last location of a 1K byte memory chip is given as FBFFH. Specify the starting address.

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

Memory interfacing

Memory Based Problem

8. The memory address of last location of an 8K byte memory chip is given as FFFFH. Specify the starting address.

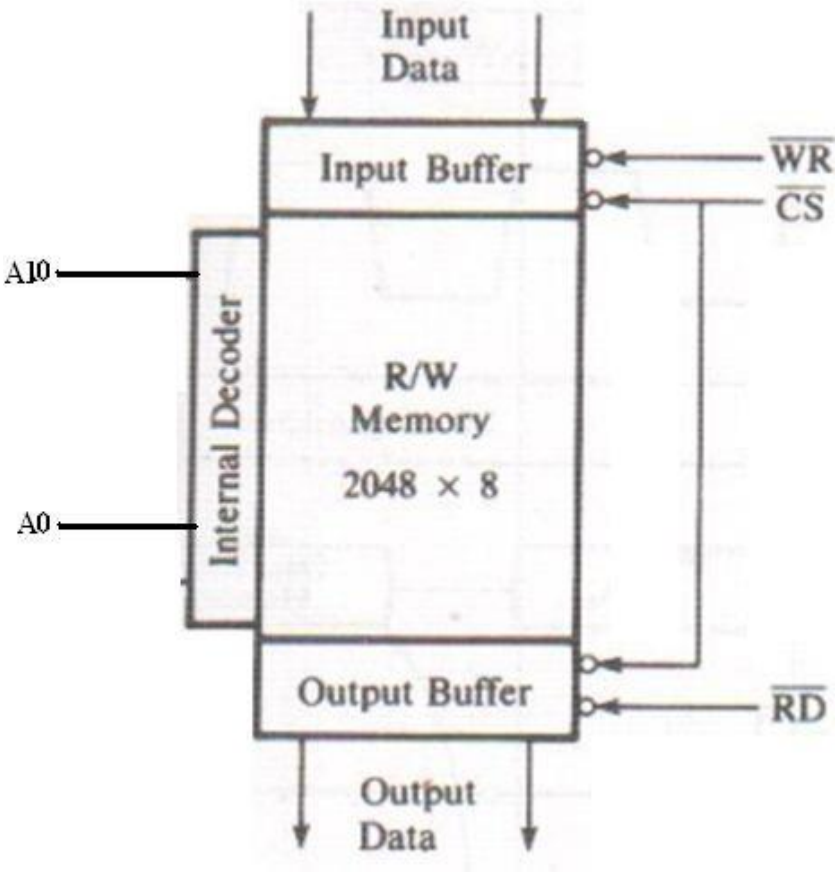
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Memory interfacing

Memory Structure and its Requirements

Address lines ???

- 11 address lines (AD10-AD0), 1 chip select, 2 control lines to enable input and output buffer.



- R/W Memory: Group of Registers.
- 2048 registers
- Register store 8-bits
- 8 input, 8-output lines
- Internal decoder to decode address lines.

Memory interfacing

Memory Structure and its Requirements

Address Line ???

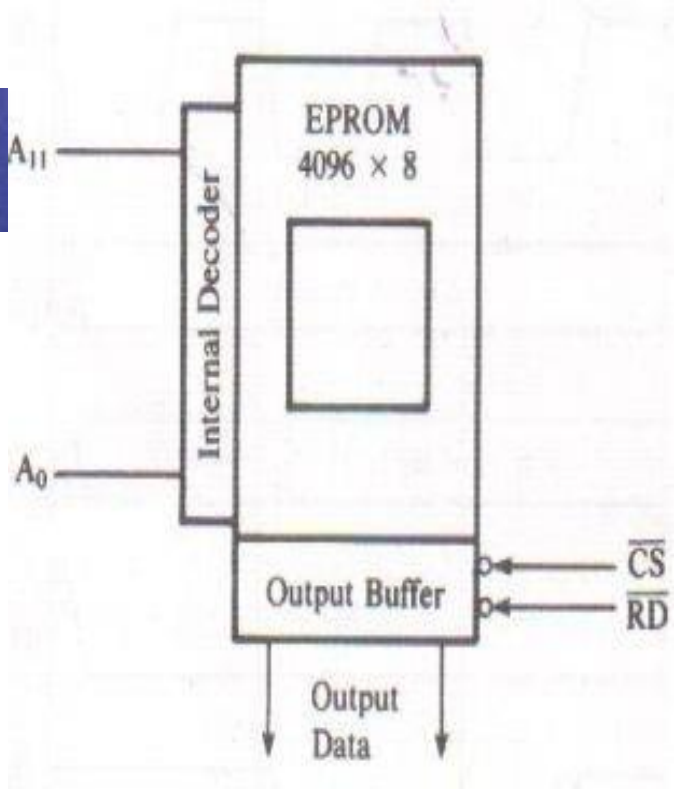
12 address lines (A11-A0), 1 chip select, 1 Read Control Signal lines to enable output buffer.

Generally EPROMS are used as program memory and RAM as data memory.

EPROM : Chip must be programmed before it can be used as ROM.

Figure Shows:

- 4096 (4K) registers.
- Register store 8-bits
- 8 input lines
- Internal decoder to decode address lines.



Memory interfacing

Basic concepts of Memory Interfacing

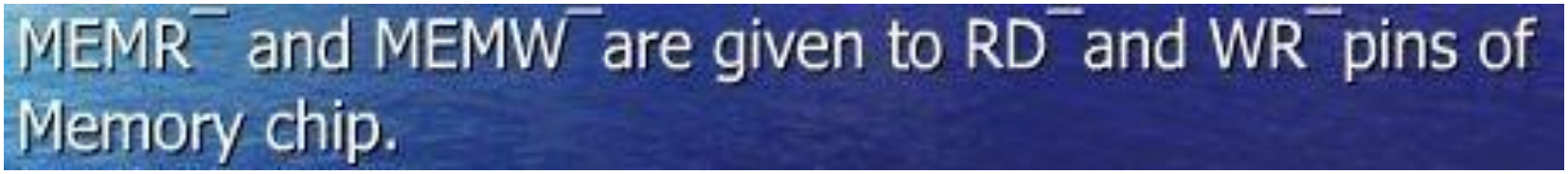
- 8085 places 16-bit address on address bus, and with this address only one register should be selected (only 11 low order address lines are required). Internal decoder of chip will identify and select the register for EPROM.
- Remaining 8085 address lines (A15-A11) should be decoded to generate chip select.
- 8085 provides two signal IO/M and RD to indicate that is memory read operation MEMR'. MEMR' control signal that can be used to enable output buffer by connecting to memory signal RD'. (Similarly signal-IO/M' and WR'– indicates memory write operation MEMW').

Memory interfacing

Basic concepts of Memory Interfacing

- 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:

  - Select the Chip
  - Identify the register
  - Enable the appropriate buffer.

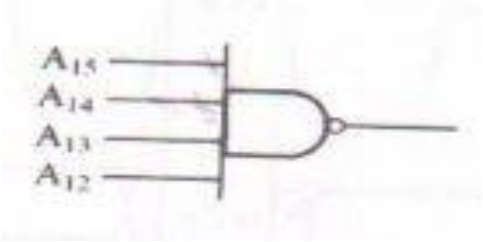


Memory interfacing

Address Decoding and Memory Addresses

- EPROM Address Lines A11-A0 are connected to memory chip.
- Remaining Address lines A15-A12 of 8085 microprocessor must be decoded.
- Two methods to decoding these lines:

a) **NAND gate:** Output of NAND gate is active and select the chip only when all address lines A15-A12 are at logic1.



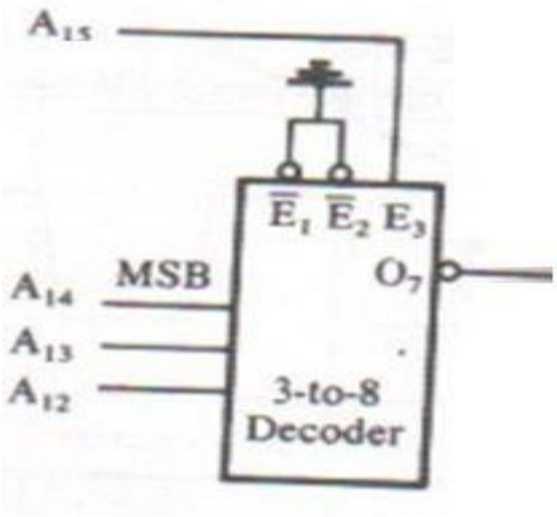
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
Chip Select				1	1	1	1	1	1	1	1	1	1	1	1	0FFFH

Memory interfacing

Address Decoding and Memory Addresses

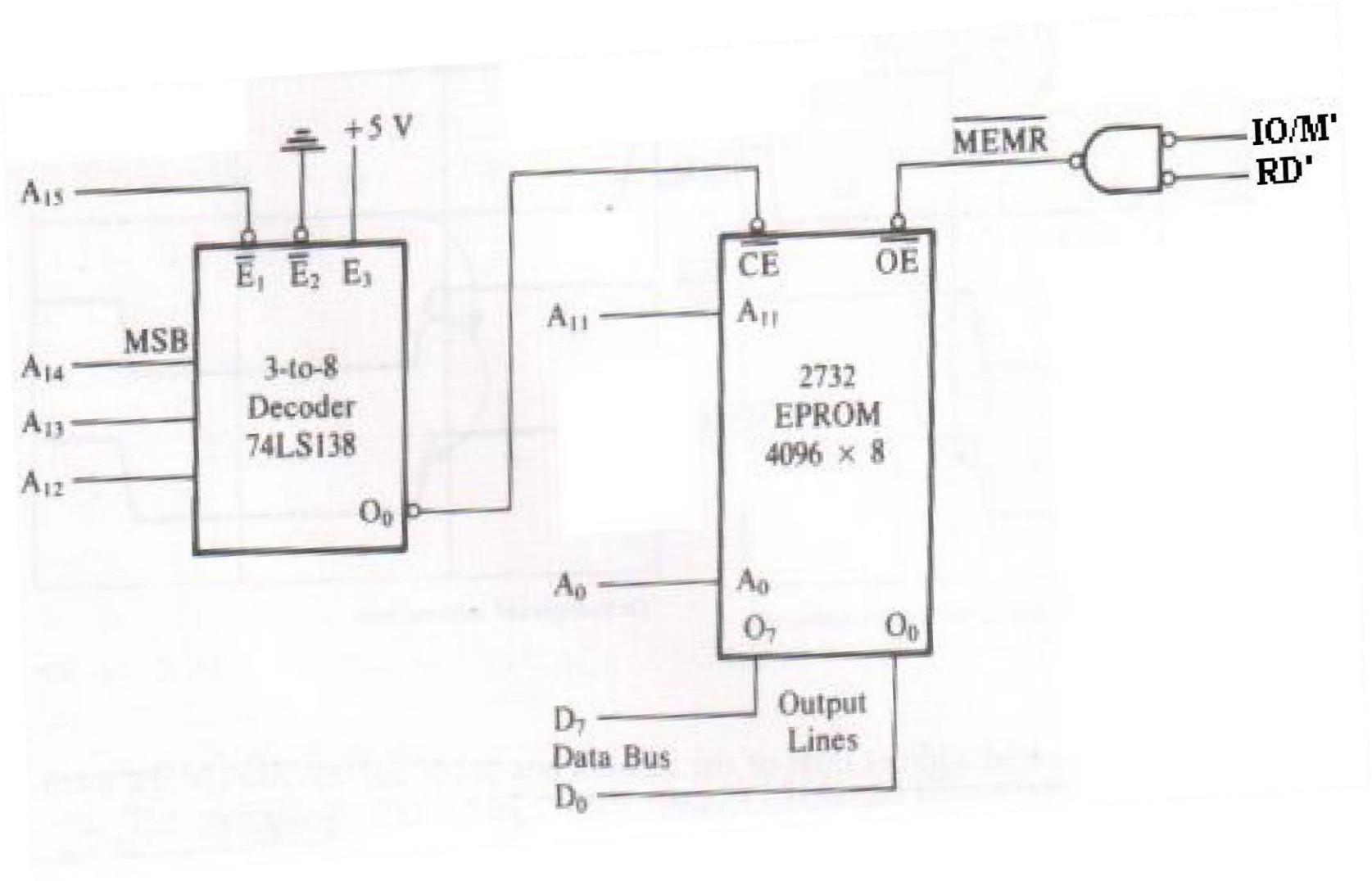
b) **using 3x8 Decoder**:

- If enable line is active eight different logic combination can be identified by output line O7.
- E1 and E2 are enable by grounding and A15 must be at logic 1.



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
Chip Select				1	1	1	1	1	1	1	1	1	1	1	1	0FFFH

Interfacing Circuit using 3x8 Decoder to interface 2732 EPROM



Interfacing Circuit using 3x8 Decoder to interface 2732 EPROM

- One control signal MEMR' is connected to OE' to enable output buffer.
- The 8085 address lines A11-A0 are connected to the pins A11-A0 of the memory chip.
- Decoder decode A15-A12 and output O0 is connected to CE' which is asserted only when A15-A12 is 0000 (A15 low enables decoder and input 000 asserts the output O0).

Memory interfacing

Address decoding and reading from memory

- Examine how 8085 places the address 0FFFH on address bus.
- The address 0000H goes to decoder.
- Output line O0 of the decoder selects the chip.
- Remaining address lines FFFH goes on address lines of the chip and the internal decoder decodes the address and selects the register FFFH.
- When RD' is asserted the output buffer is enabled and the contents of register 0FFFH are places on the data bus for the processor to read.

Address Decoding for SRAM

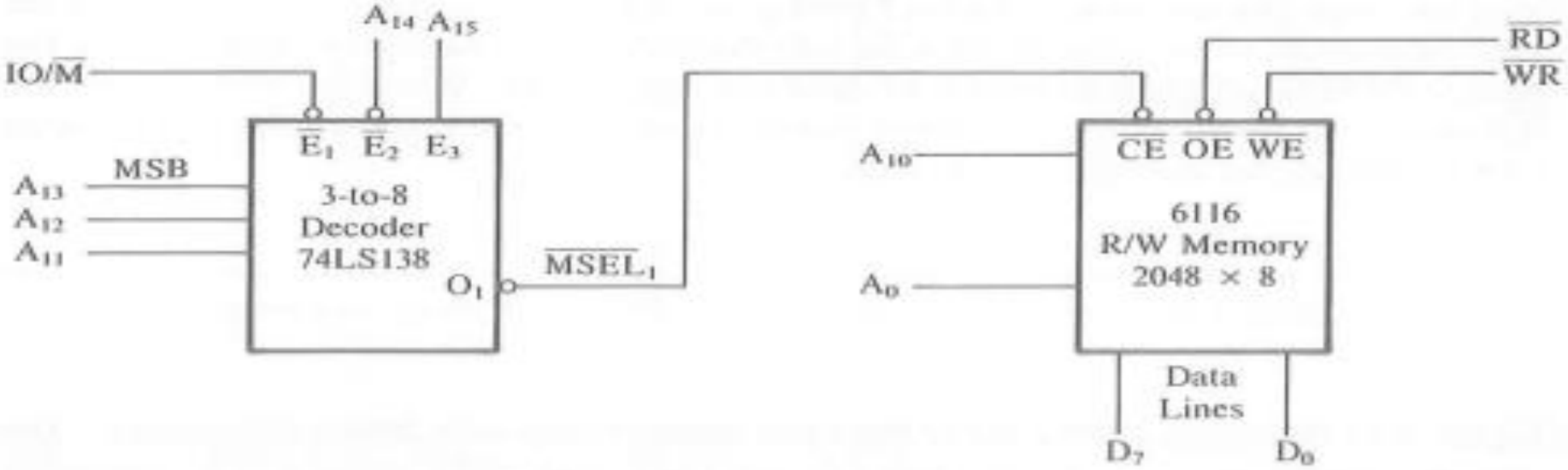
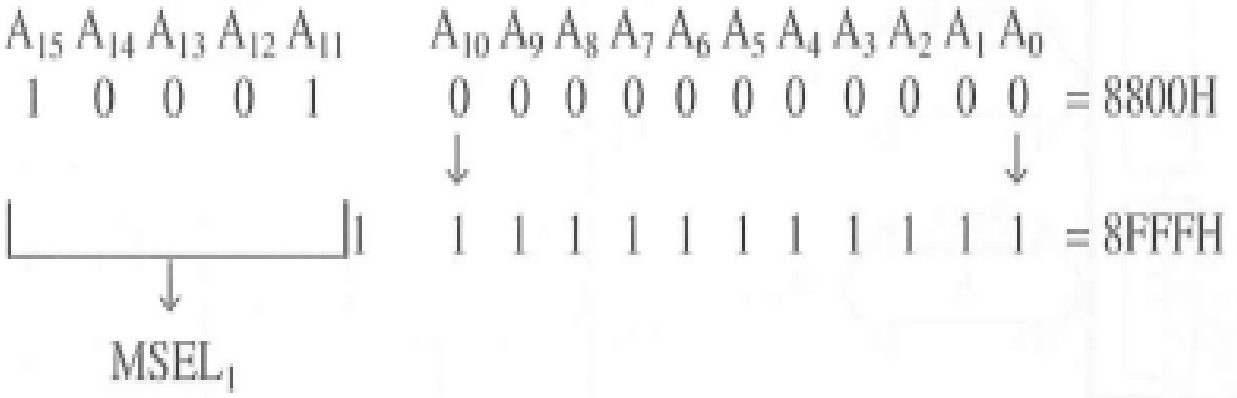


FIGURE 4.17
Interfacing R/W Memory



Daily Quiz

- Which of the following is a type of microprocessor?
  a) CISC
  b) RISC
  c) EPIC
  d) **All of the mentioned**
- Which of the following technology was used by Intel to design its first 8-bit microprocessor?
  a) NMOS
  b) HMOS
  c) **PMOS**
  d) TTL
- Which of the following is true about microprocessors?
  a) It has an internal memory
  b) It has interfacing circuits
  c) **It contains ALU, CU, and registers**
  d) It uses Harvard architecture

Daily Quiz

- What is Microprocessor?
  a) **A multipurpose PLD that accepts binary data as input**
  b) A multipurpose PLD that accepts an integer as input
  c) A multipurpose PLD that accepts whole numbers as input
  d) A multipurpose PLD that accepts prime numbers as input

- Which of the following is not true about 8085 microprocessor?
  a) It is an 8-bit microprocessor
  b) It is a 40 pin DIP chip
  c) **It is manufactured using PMOS technology**
  d) It has 16 address lines

Weekly Assignment

1. What is Memory Interfacing?
2. Explain 8085 interfacing with memory Chips
3. Explain Timing Diagram for Memory Read Machine Cycle.
4. Explain Timing Diagram for Memory Write Machine Cycle.
5. Explain Memory structure and its requirement.

Topic Links

https://www.youtube.com/watch?v=cylTOKGdFJI

https://www.youtube.com/watch?v=mU7wpeThu7A

Topic-3 Objective

Name of the Topic	Objective of the topic	Mapping with CO
Interfacing output displays	To understand the concept of output displays	CO4

Interfacing output displays

5.2.1 Illustration: LED Display for Binary Data

PROBLEM STATEMENT

1. Analyze the interfacing circuit in Figure 5.8(a), identify the address of the output port, and explain the circuit operation.
2. Explain similarities between (a) and (b) in Figure 5.8.
3. Write instructions to display binary data at the port.

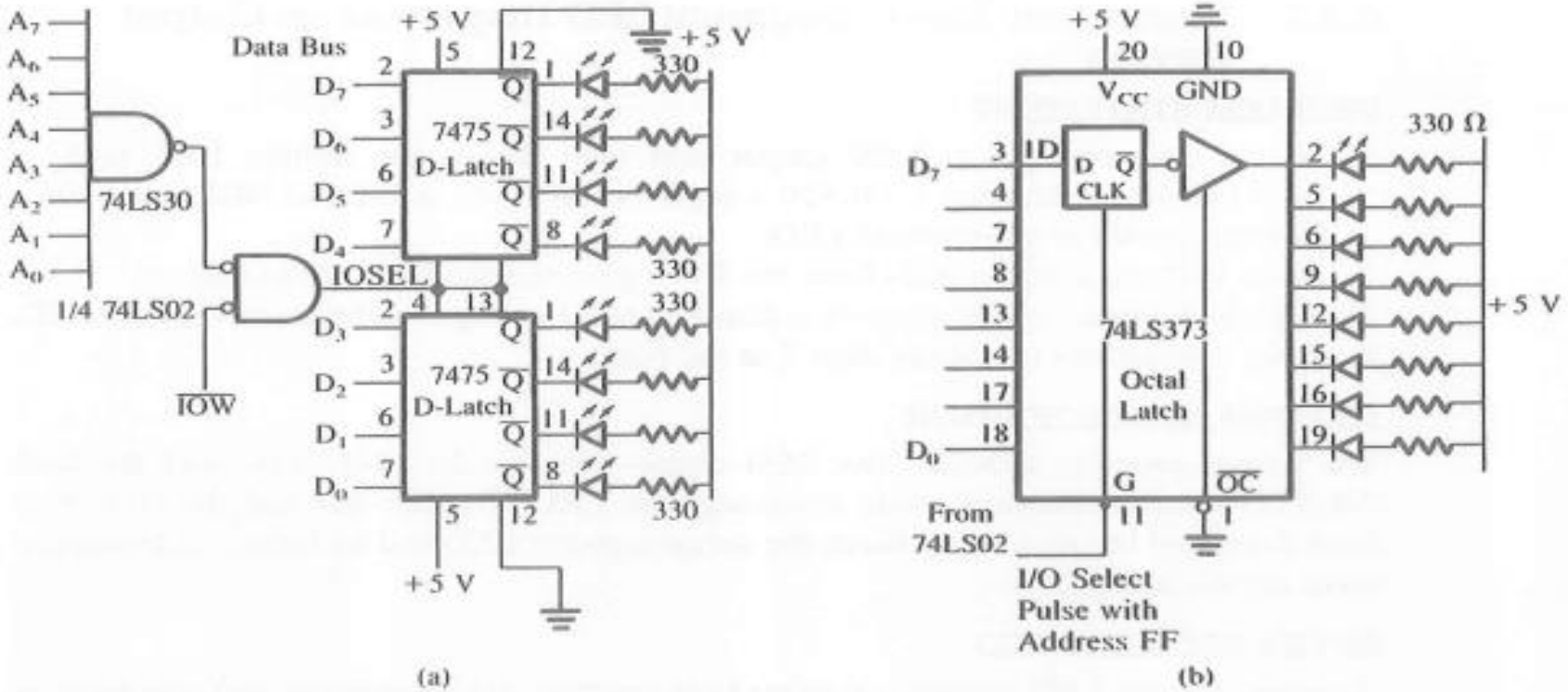


FIGURE 5.8
Interfacing LED Output Port Using the 7475 D-Type Latch (a) and Using the 74LS373 Octal D-Type Latch (b)

Interfacing output displays

PROGRAM

Address (LO)	Machine Code	Mnemonics	Comments
00	3E	MVI A,DATA	;Load accumulator with data
01	DATA*		
02	D3	OUT FFH	;Output accumulator contents
			; to port FFH
03	FF		
04	76	HLT	;End of program

Interfacing output displays

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.

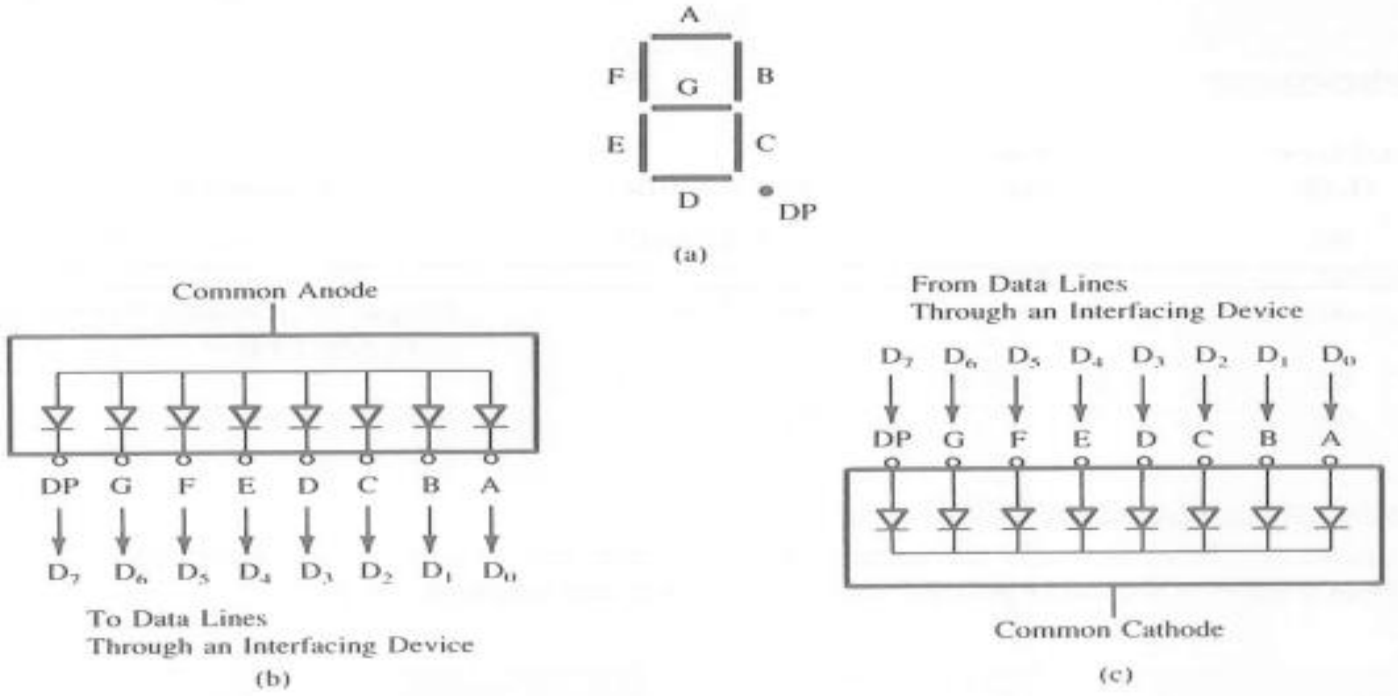


FIGURE 5.9
Seven-Segment LED: LED Segments (a); Common-Anode LED (b); Common-Cathode LED (c)

Interfacing output displays

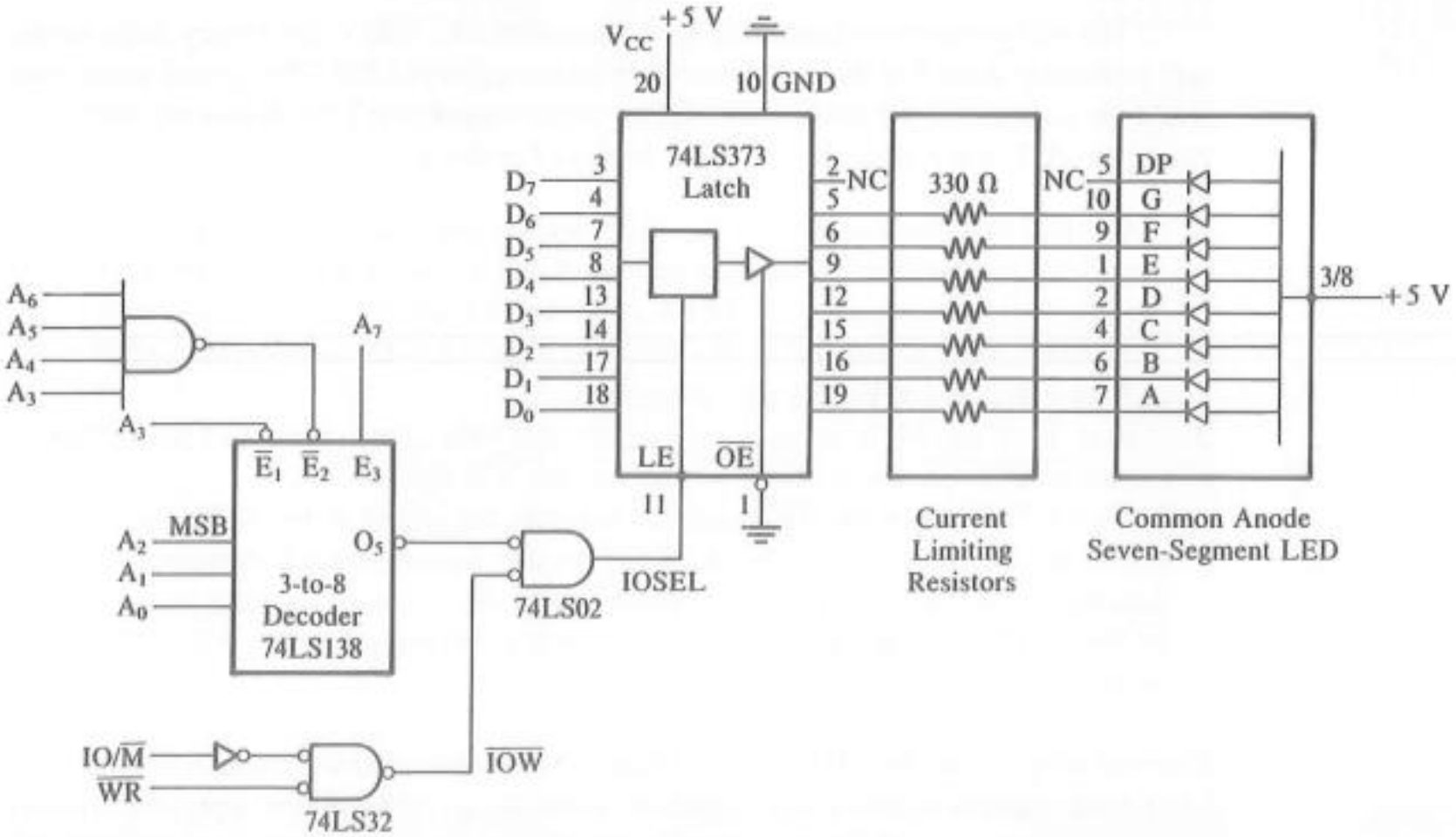


FIGURE 5.10
Interfacing Seven-Segment LED

Interfacing output displays

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	

To design an output port with the address F5H, the address lines A_7–A_0 should have the 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 digit 7 at the output port:

```
MVI A,78H    ;Load seven-segment code in the accumulator
OUT F5H      ;Display digit 7 at port F5H
HLT          ;End
```

Daily Quiz

- There are _____ general purpose registers in 8085 processor

A. 5
B. 6
C. 7
D. 8

- Flag register is an 8-bit register having _____ 1-bit flip-flops.

A. 3
B. 4
C. **5**
D. 6

- What is true about Program counter?

A. It is an 8-bit register, which holds the temporary data of arithmetic and logical operations.
B. When an instruction is fetched from memory then it is stored in the program counter
C. It provides timing and control signal to the microprocessor
D. **It is a 16-bit register used to store the memory address location of the next instruction to be executed.**

Weekly Assignment

1. Explain 8085 interfacing with I/O Devices.
2. Differentiate between Memory Mapping and Peripheral.
3. Explain Peripheral I/O Instructions.
4. Explain IN Instruction.
5. Explain OUT Instruction.

Topic-4 Objective

Name of the Topic	Objective of the topic	Mapping with CO
Interfacing input devices	To understand the concept of input devices	CO4

Interfacing Input Devices

5.3.1 Illustration: Data Input from DIP Switches

In this section, we will analyze the circuit used for interfacing eight DIP switches, as shown in Figure 5.11. The circuit includes the 74LS138 3-to-8 decoder to decode the low-order bus and the tri-state octal buffer (74LS244) to interface the switches to the data bus. The port can be accessed with the address 84H; however, it has multiple addresses, as explained below.

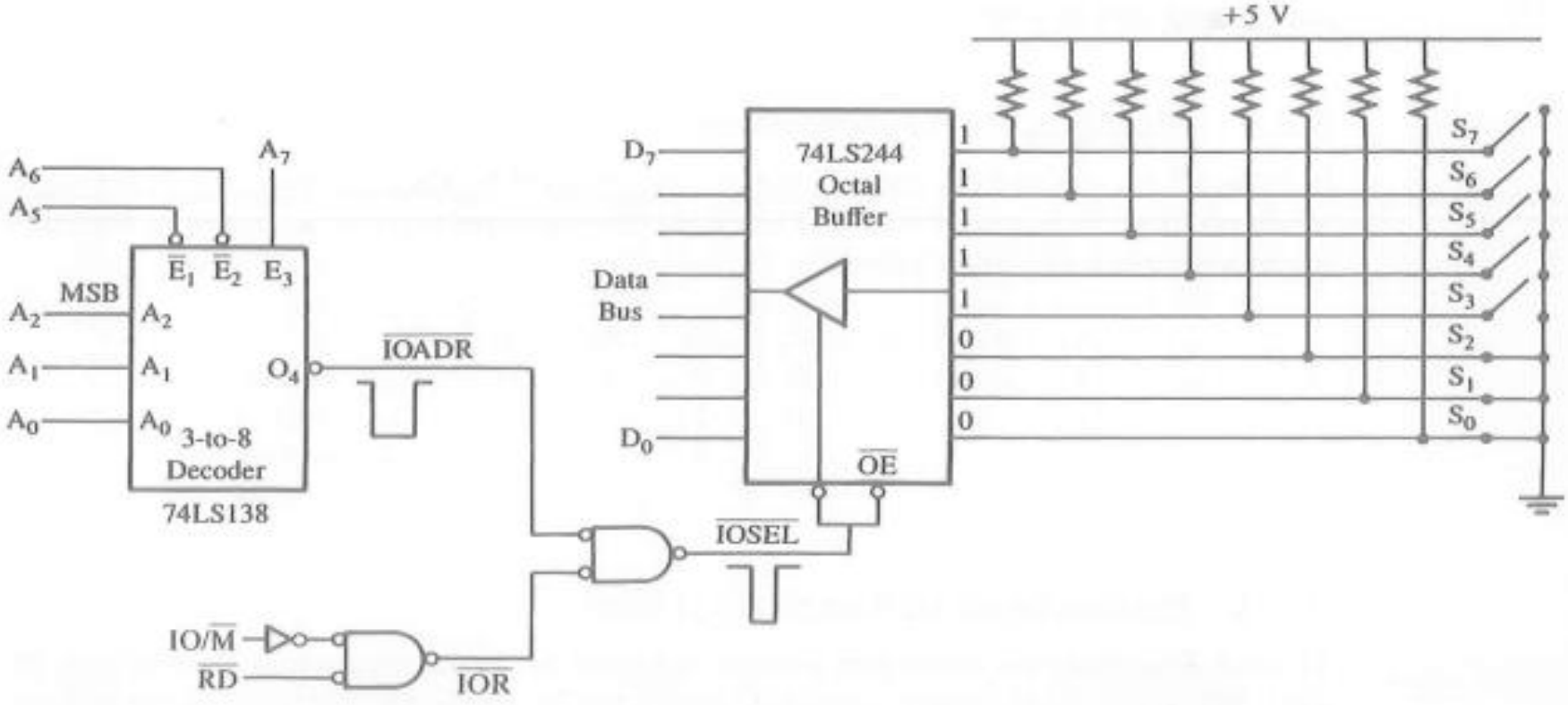


FIGURE 5.11
Interfacing DIP Switches

Interfacing Input Devices

The output line O_4 of the decoder 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
Enable Lines			Don't Care		Input			

5.3.4 Multiple Port Addresses

In Figure 5.11, the address lines A_4 and A_3 are not used by the decoding circuit; the logic levels on these lines can be 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

Daily Quiz

1. The device that enables the microprocessor to read data from the external devices is
   a) printer
   b) joystick
   c) display
   d) reader
2. The example of output device is
   a) CRT display
   b) 7-segment display
   c) Printer
   d) All of the mentioned
3. The input and output operations are respectively similar to the operations,
   a) read, read
   b) write, write
   c) **read, write**
   d) write, read

Daily Quiz

4. The operation, IOWR (active low) performs
   a) write operation on input data
   b) write operation on output data
   c) read operation on input data
   d) read operation on output data
5. The latch or IC 74LS373 acts as
   a) good input port
   b) bad input port
   c) good output port
   d) bad output port
6. While performing read operation, one must take care that much current should not be
   a) sourced from data lines
   b) sinked from data lines
   c) sourced or sinked from data lines
   d) sinked from address lines

Topic Links

1. https://www.youtube.com/watch?v=Us_e3I7Mfnk
2. https://www.youtube.com/watch?v=Tbur5vgIPuk
3. https://www.youtube.com/watch?v=VAChQ9n-xSI

Topic-5 Objective

Name of the Topic	Objective of the topic	Mapping with CO
Memory mapped I/O	To understand the concept of Memory mapped I/O	CO4

Memory mapped I/O

- In memory mapped I/O interfacing with 8085 microprocessor, the I/O devices are not given separate addresses other than treated as a memory location.
- It has address range between 0000H to FFFFH (64k). But some part of the space is reserved for I/O devices.

Why we use memory mapped I/O interfacing?

- The advantage is any instruction that related to memory handling can use in this mode. By using those instruction we can also transfer data between an I/O device and the microprocessor.
- For that, the I/O port should assigned to the memory address space rather than to the I/O address space.
- The register associated with the I/O port is simply treated as memory location register.

Memory mapped I/O

- For example, some of 8085A instructions that can be used for memory mapped I/O ports:

  MOV M,r
  STA addr
  MVI M, data
  SHLD addr

Timing diagram for STA 8000H

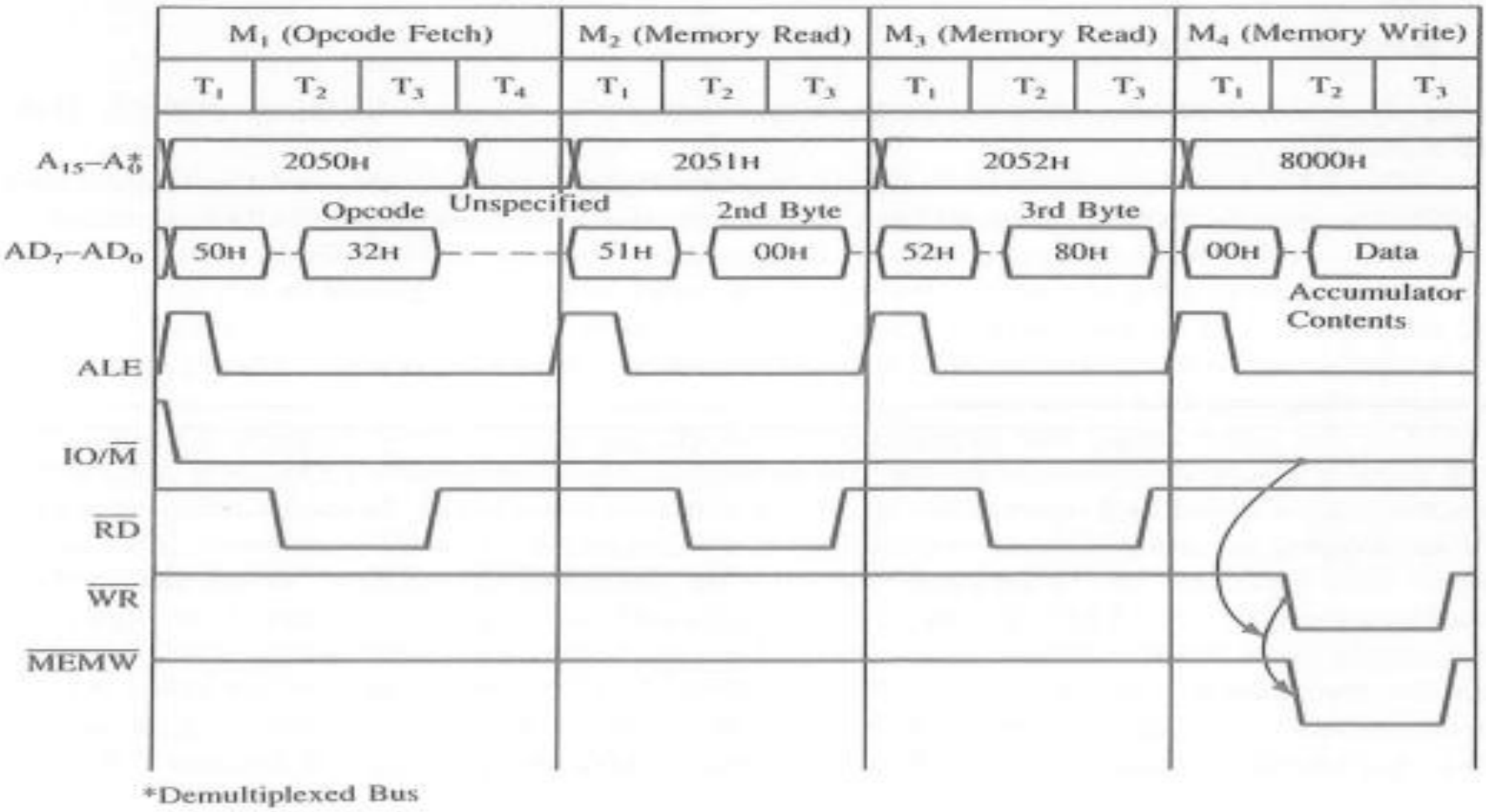


FIGURE 5.12
Timing for Execution of the Instruction: STA 8000H

Safety Control System using Memory-Mapped I/O Technique

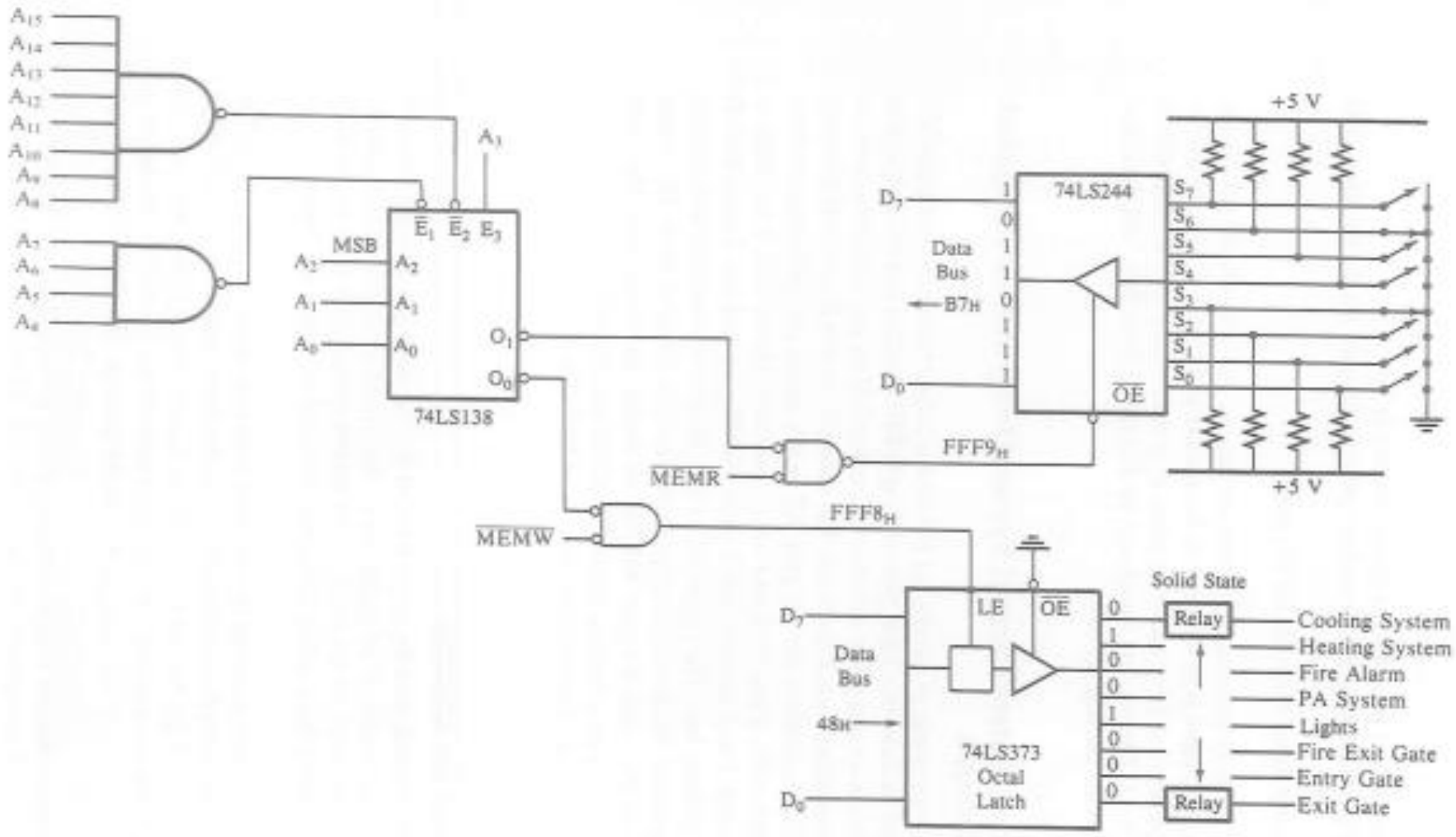
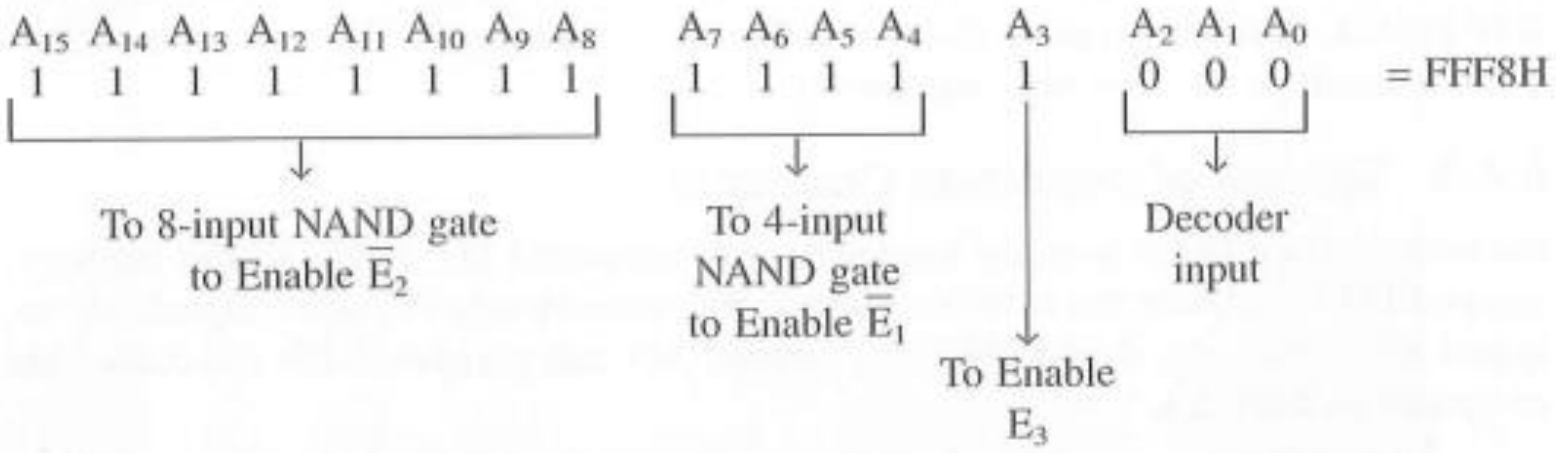


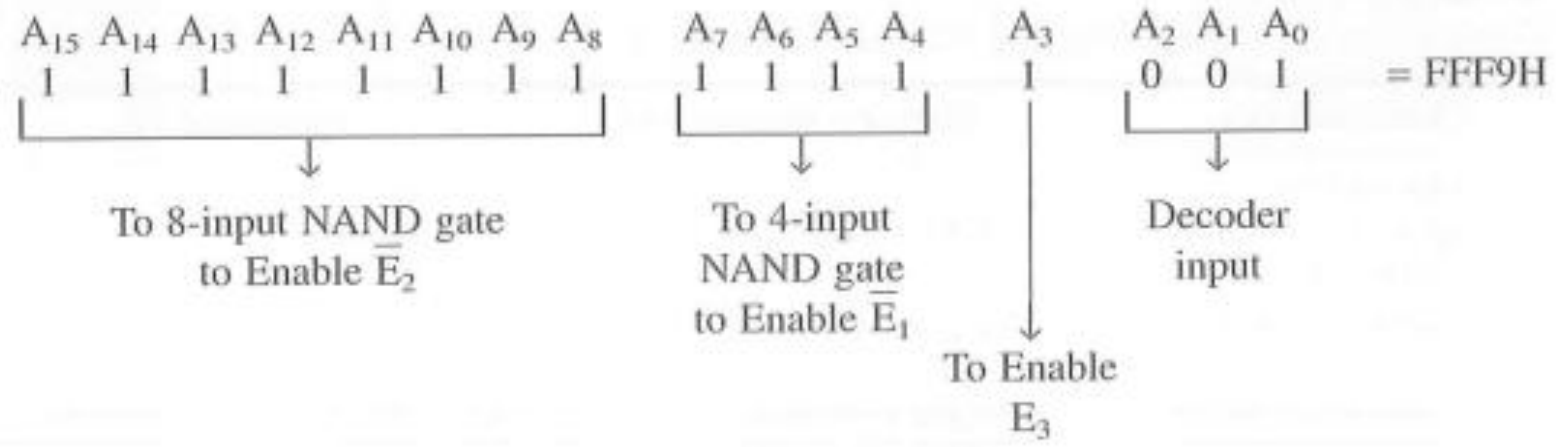
FIGURE 5.13
Memory-Mapped I/O Interfacing

Memory mapped I/O

Output port and its address



Input port and its address



Memory mapped I/O- Instructions

```
READ: LDA FFF9H     ;Read the switches
      CMA           ;Complement switch reading, convert on-switch (logic 0)
                    ;   into logic 1 to turn on appliances
      STA FFF8H     ;Send switch positions to output port and turn on/off appli-
                    ;   ances
      JMP READ      ;Go back and read again
```

Memory mapped I/O and I/O mapped I/O

TABLE 5.1
Comparison of Memory-Mapped I/O and Peripheral I/O

Characteristics	Memory-Mapped I/O	Peripheral I/O
1. Device address	16-bit	8-bit
2. Control signals for Input/Output	$\overline{MEMR}$/$\overline{MEMW}$	$\overline{IOR}$/$\overline{IOW}$
3. Instructions available	Memory-related instructions such as STA; LDA; LDAX; STAX; MOV M,R: ADD M; SUB M; ANA M: etc.	IN and OUT
4. Data transfer	Between any register and I/O	Only between I/O and the accumulator
5. Maximum number of I/Os possible	The memory map (64K) is shared between I/Os and system memory	The I/O map is independent of the memory map; 256 input devices and 256 output devices can be connected
6. Execution speed	13 T-states (STA,LDA) 7 T-states (MOV M,R)	10 T-states
7. Hardware requirements	More hardware is needed to decode 16-bit address	Less hardware is needed to decode 8-bit address
8. Other features	Arithmetic or logical operations can be directly performed with I/O data	Not available

Daily Quiz

1. In memory-mapped I/O _____
   a) The I/O devices and the memory share the same address space
   b) The I/O devices have a separate address space
   c) The memory and I/O devices have an associated address space
   d) A part of the memory is specifically set aside for the I/O
2. The usual BUS structure used to connect the I/O devices is _____.
   a) Star BUS structure
   b) Multiple BUS structure
   c) Single BUS structure
   d) Node to Node BUS structure
3. The advantage of I/O mapped devices to memory mapped is _____.
   a) The former offers faster transfer of data
   b) The devices connected using I/O mapping have a bigger buffer space
   c) The devices have to deal with fewer address lines
   d) No advantage as such

Daily Quiz

4. The system is notified of a read or write operation by _____.
   a) Appending an extra bit of the address
   b) Enabling the read or write bits of the devices
   c) Raising an appropriate interrupt signal
   d) Sending a special signal along the BUS
5. To overcome the lag in the operating speeds of the I/O device and the processor we use _____.
   a) BUffer spaces
   b) Status flags
   c) Interrupt signals
   d) Exceptions
6. The method of accessing the I/O devices by repeatedly checking the status flags is _____.
   a) Program-controlled I/O
   b) Memory-mapped I/O
   c) I/O mapped
   d) None of the mentioned

Weekly Assignment

7. The system is notified of a read or write operation by _____.
   a) Appending an extra bit of the address
   b) Enabling the read or write bits of the devices
   c) Raising an appropriate interrupt signal
   d) Sending a special signal along the BUS
8. To overcome the lag in the operating speeds of the I/O device and the processor we use _____
   a) BUffer spaces
   b) Status flags
   c) Interrupt signals
   d) Exceptions
9. The method of accessing the I/O devices by repeatedly checking the status flags is _____
   a) Program-controlled I/O
   b) Memory-mapped I/O
   c) I/O mapped
   d) None of the mentioned

Topic Links

1. https://www.youtube.com/watch?v=Us_e3I7Mfnk
2. https://www.youtube.com/watch?v=Tbur5vgIPuk
3. https://www.youtube.com/watch?v=VAChQ9n-xSI

Topic-6 Objective

Name of the Topic	Objective of the topic	Mapping with CO
Interfacing keyboard and seven segment displays	To understand the Interfacing keyboard and seven segment displays	CO4

Interfacing keyboard and seven segment displays

Write a program to monitor the keyboard to sense a key pressed and display the number of the key at the seven-segment LED. For example, when the key K_7 is pressed, the digit 7 should be displayed at port B.

Solution:

1. Check if a key is pressed.
2. Debounce the key.
3. Identify and encode the key in appropriate binary format.
4. Obtain the seven-segment code and display it.

Interfacing keyboard and seven segment displays

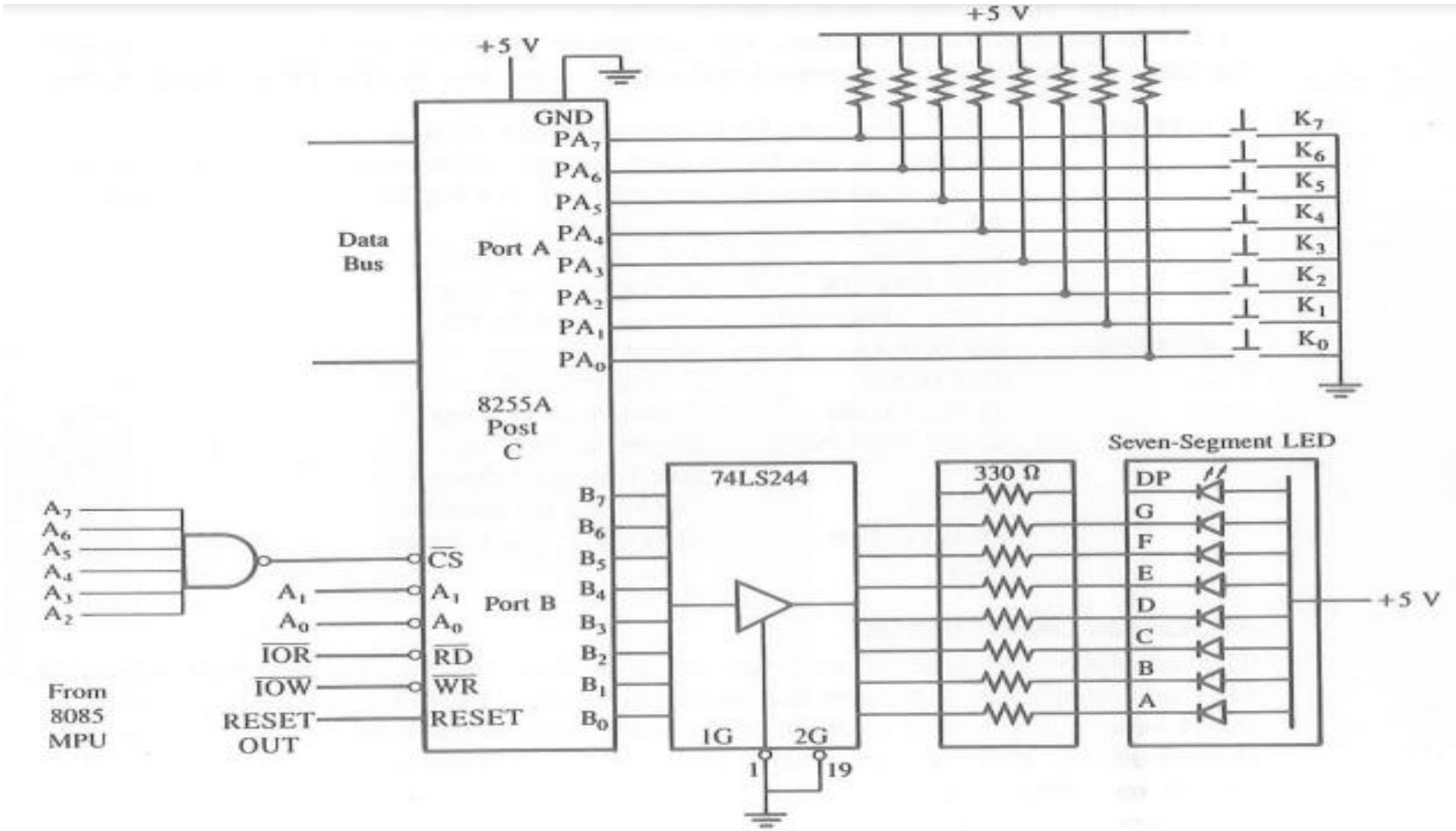


FIGURE 15.15
Interfacing a Keyboard and a Seven-Segment LED

Interfacing keyboard and seven segment displays

Table 15.2: related key codes and their encoded binary digits

Key press	Data at Port A	Description	Encoded binary digits
none	11111111=FFH	All keys are connected to 5 volts	none
SW_7	01111111=7FH	Only SW_7 connected to ground	00000111=07
SW_6	10111111=BFH	Only SW_6 connected to ground	00000110=06
SW_5	11011111=DFH	Only SW_5 connected to ground	00000101=05
SW_4	11101111=EFH	Only SW_4 connected to ground	00000100=04
SW_3	11110111=F7H	Only SW_3 connected to ground	00000011=03
SW_2	11111011=FBH	Only SW_2 connected to ground	00000010=02
SW_1	11111101=FDH	Only SW_1 connected to ground	00000001=01
SW_0	11111110=FEH	Only SW_0 connected to ground	00000000=00

Seven segment displays

MAIN PROGRAM

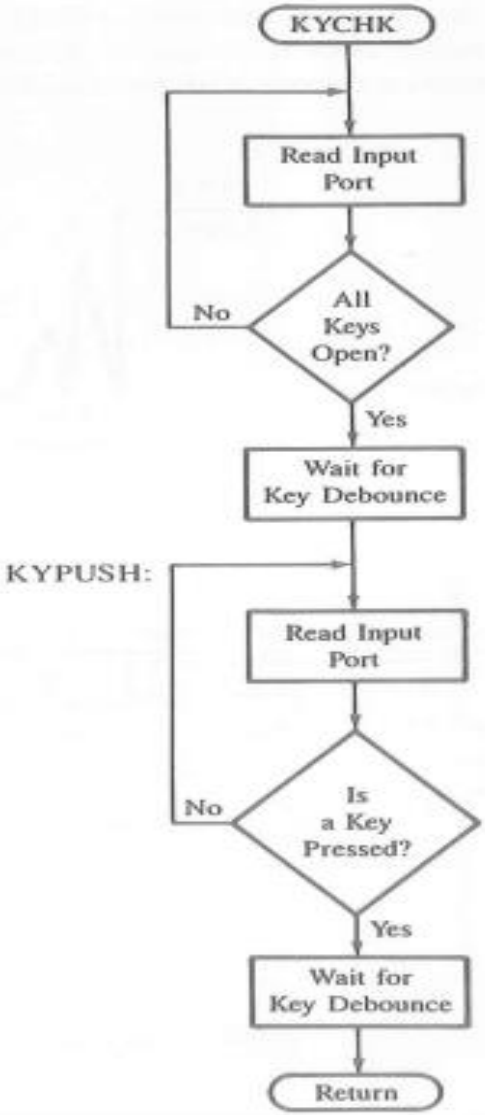
KYBOARD:

```
PORT A          EQU FCH
PORT B          EQU FDH

CNTRL    EQU FFH
CNWORD   EQU 90H
STACK    EQU 20AFH
         LXI SP,STACK
PPI:     MVI A,CNWORD
         OUT CNTRL
NEXTKY:  CALL KYCHK
         CALL KYCODE
         CALL DSPLAY
         JMP NEXTKY
```

Interfacing keyboard and seven segment displays

FIGURE 15.16
Flowchart: Key Check Subroutine



Interfacing keyboard and seven segment displays

PROGRAM FOR KEYCHECK SUBROUTINE

```
KYCHK:      IN PORT A
            CPI 0FFH
            JNZ KYCHK
            CALL DBOUNCE
KYPUSH:     IN PORT A
            CPI 0FFH
            JZ KYPUSH
            CALL DBOUNCE
            CMA
            ORA A
            JZ KYPUSH
            RET
```

Key Debounce

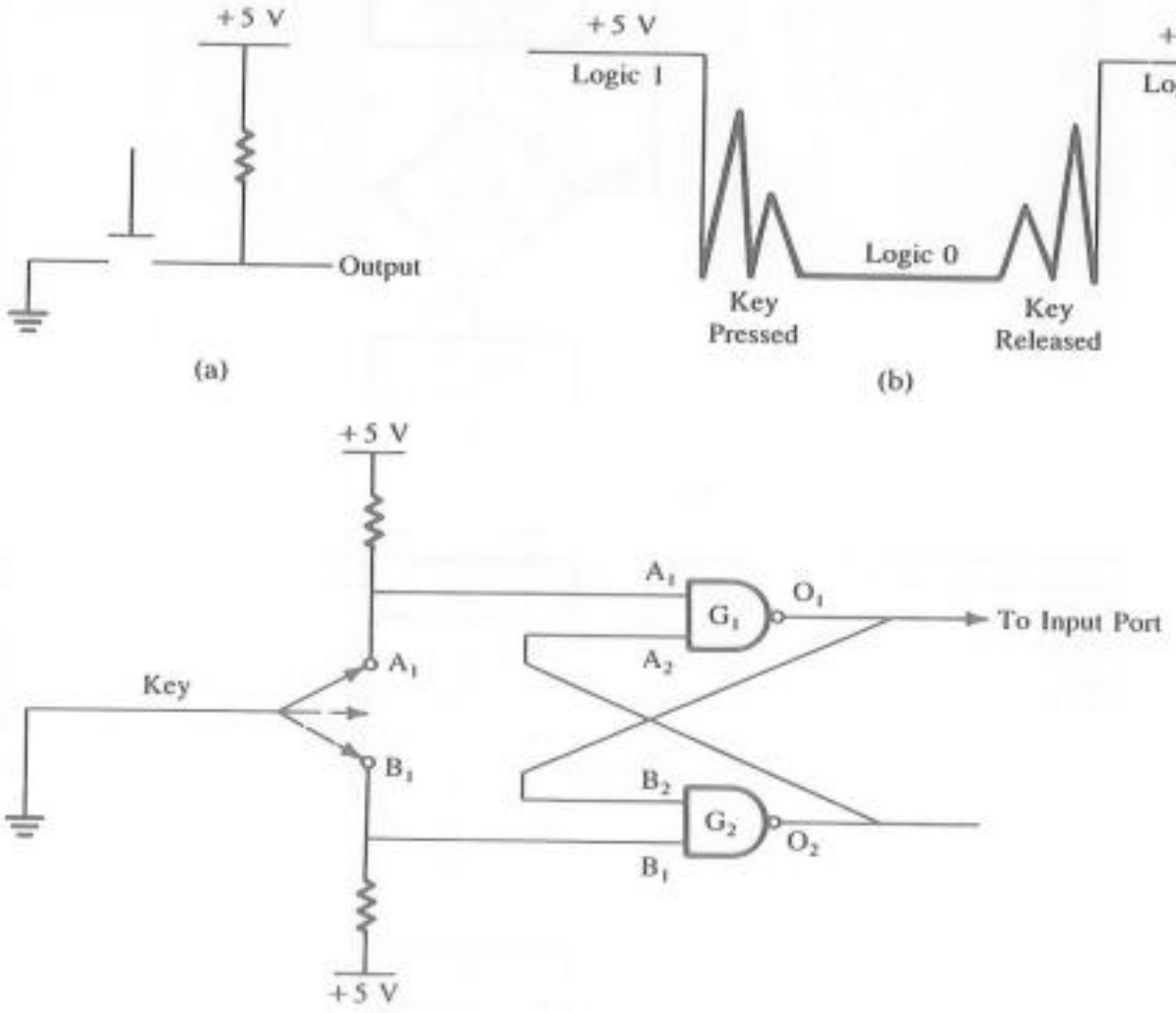


FIGURE 15.17
Pushbutton Key (a), Key Bounce (b), and Key Debounce Circuit Using NAND Gates (c)

Interfacing keyboard and seven segment displays

In the software technique, when a key closure is found, the microprocessor waits for 10 to 20 ms before it accepts the key as an input. The delay routine is as follows:

```
DBONCE:  ;This is a 20 ms delay routine
         ;The delay COUNT should be calculated based on system frequency
         ;This does not destroy any register contents
         ;Input and Output = None
         PUSH B          ;Save register contents
         PUSH PSW
         LXI B,COUNT     ;Load delay count
LOOP:    DCX B           ;Next count
         MOV A,C
         ORA B           ;Set Z flag if (BC) = 0
         JNZ LOOP
         POP PSW         ;Restore register contents
         POP BC
         RET
```

Interfacing keyboard and seven segment displays

```
KYCODE: ;This routine converts (encodes) the binary hardware reading of the key
        ;  pressed into appropriate binary format according to the number of the
        ;  key.
        MVI C,08H     ;Set code encounter
NEXT:   DCR C         ;Adjust key code
        RAL           ;Place MSB in CY
        JNC NEXT      ;If bit = 0, go back to check next bit
        MOV A,C       ;Place key code in the accumulator
        RET
```

Seven segment displays

```
DSPLAY:  ;This routine takes the binary number and converts into its common-
         ;   anode seven-segment LED code. The codes are stored in memory
         ;   sequentially, starting from the address CACODE
         ;Input: Binary number in accumulator
         ;Output: None
         ;Modifies contents of HL and A
         LXI H,CACODE      ;Load starting address of code table in HL
         ADD L             ;Add digit to low-order address in L
         MOV L,A           ;Place code address in L
         MOV A,M           ;Get code from memory
         OUT PORTB         ;Send code to port B
         RET
CACODE:  ;Common-anode seven-segment codes are stored sequentially in memory
         DB 40H,79H,24H,30H,19H,12H  ;Codes for digits from 0 to 5
         DB 02H,78H,00H,18H,08H,03H  ;Codes for digits from 6 to B
         DB 46H,21H,06H,0EH          ;Codes from digits from C to F
```

Seven segment displays

Applications of Seven Segment Displays:

Common applications of seven segment displays are in:

1. Digital clocks
2. Clock radios
3. Calculators
4. Wristwatches
5. Speedometers
6. Motor-vehicle odometers
7. Radio frequency indicators

Weekly Assignment

1. Write down the applications of seven segment display.
2. Explain Interfacing the keyboard.
3. Explain the working of seven segment display.

Topic Links

1. https://www.youtube.com/watch?v=Us_e3I7Mfnk
2. https://www.youtube.com/watch?v=Tbur5vgIPuk
3. https://www.youtube.com/watch?v=VAChQ9n-xSI

Topic-7 Objective

Name of the Topic	Objective of the topic	Mapping with CO
The 8085 Interrupts	To understand the Interrupts	CO4

The 8085 Interrupts

Interrupts in 8085

- Interrupts are the signals generated by the external devices to request the microprocessor to perform a task.
- There are 5 interrupt signals, i.e.

  TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR.
- Interrupt are classified into following groups based on their parameter −

Vector interrupt

- In this type of interrupt, the interrupt address is known to the processor.

  For example: RST7.5, RST6.5, RST5.5, TRAP.

Non-Vector interrupt

- In this type of interrupt, the interrupt address is not known to the processor so, the interrupt address needs to be sent externally by the device to perform interrupts.

  For example: INTR.

The 8085 Interrupts

Maskable interrupt

- In this type of interrupt, we can disable the interrupt by writing some instructions into the program.

  For example: RST7.5, RST6.5, RST5.5.

Non-Maskable interrupt

- In this type of interrupt, we cannot disable the interrupt by writing some instructions into the program.

  For example: TRAP.

Software interrupt

- In this type of interrupt, the programmer has to add the instructions into the program to execute the interrupt.
- There are 8 software interrupts in 8085, i.e. RST0, RST1, RST2, RST3, RST4, RST5, RST6 and RST7.

Hardware interrupt

- There are 5 interrupt pins in 8085 used as hardware interrupts, i.e. TRAP, RST7.5, RST6.5, RST5.5, INTA.

The 8085 Interrupts

- INTA' is not an interrupt, it is used by the microprocessor for sending acknowledgement.
- TRAP has the highest priority, then RST7.5 and so on.

Interrupt Service Routine (ISR)

- A small program or a routine that when executed, services the corresponding interrupting source is called an ISR.

TRAP

- It is a non-maskable interrupt, having the highest priority among all interrupts.
- By default, it is enabled until it gets acknowledged.
- In case of failure, it executes as ISR and sends the data to backup memory.
- This interrupt transfers the control to the location 0024H.

The 8085 Interrupts

RST7.5

It is a maskable interrupt, having the second highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 003CH address.

RST 6.5

It is a maskable interrupt, having the third highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 0034H address.

RST 5.5

It is a maskable interrupt. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 002CH address.

INTR

It is a maskable interrupt, having the lowest priority among all interrupts. It can be disabled by resetting the microprocessor.

The 8085 Interrupts

When **INTR signal goes high**, the following events can occur −

•The microprocessor checks the status of INTR signal during the execution of each instruction.

•When the INTR signal is high, then the microprocessor completes its current instruction and sends active low interrupt acknowledge signal.

•When instructions are received, then the microprocessor saves the address of the next instruction on stack and executes the received instruction.

Daily Quiz

1. An interrupt breaks the execution of instructions and diverts its execution to
a) Interrupt Service Routine
b) Counter Word Register
c) Execution Unit
d) Control Unit

2. While executing the main program, if two or more interrupts occur, then the sequence of appearance of interrupts is called
a) multi-interrupt
b) nested interrupt
c) interrupt within interrupt
d) nested interrupt and interrupt within interrupt

Daily Quiz

3. An interrupt breaks the execution of instructions and diverts its execution to
a) Interrupt Service Routine
b) Counter Word Register
c) Execution Unit
d) Control Unit

2. While executing the main program, if two or more interrupts occur, then the sequence of appearance of interrupts is called
a) multi-interrupt
b) nested interrupt
c) interrupt within interrupt
d) nested interrupt and interrupt within interrupt

Topic Links

1. https://www.youtube.com/watch?v=Us_e3I7Mfnk
2. https://www.youtube.com/watch?v=Tbur5vgIPuk
3. https://www.youtube.com/watch?v=VAChQ9n-xSI

Topic-8 Objective

Name of the Topic	Objective of the topic	Mapping with CO
8085 vector interrupts	To understand the Vector Interrupts	CO1

8085 Vector Interrupts

- An interrupt vector is a pointer to where the ISR is stored in memory.
- All interrupts (vectored or otherwise) are mapped onto a memory area called the Interrupt Vector Table (IVT).
  - The IVT is usually located in memory page 00 (0000H - 00FFH).
  - The purpose of the IVT is to hold the vectors that redirect the microprocessor to the right place when an interrupt arrives.

8085 Vector Interrupts

- The 8085 has 3 Masked/Vectored interrupt inputs.
  - RST 5.5, RST 6.5, RST 7.5
    - They are all maskable.
    - They are automatically vectored according to the following table:

Interrupt	Vector
RST 5.5	002CH
RST 6.5	0034H
RST 7.5	003CH

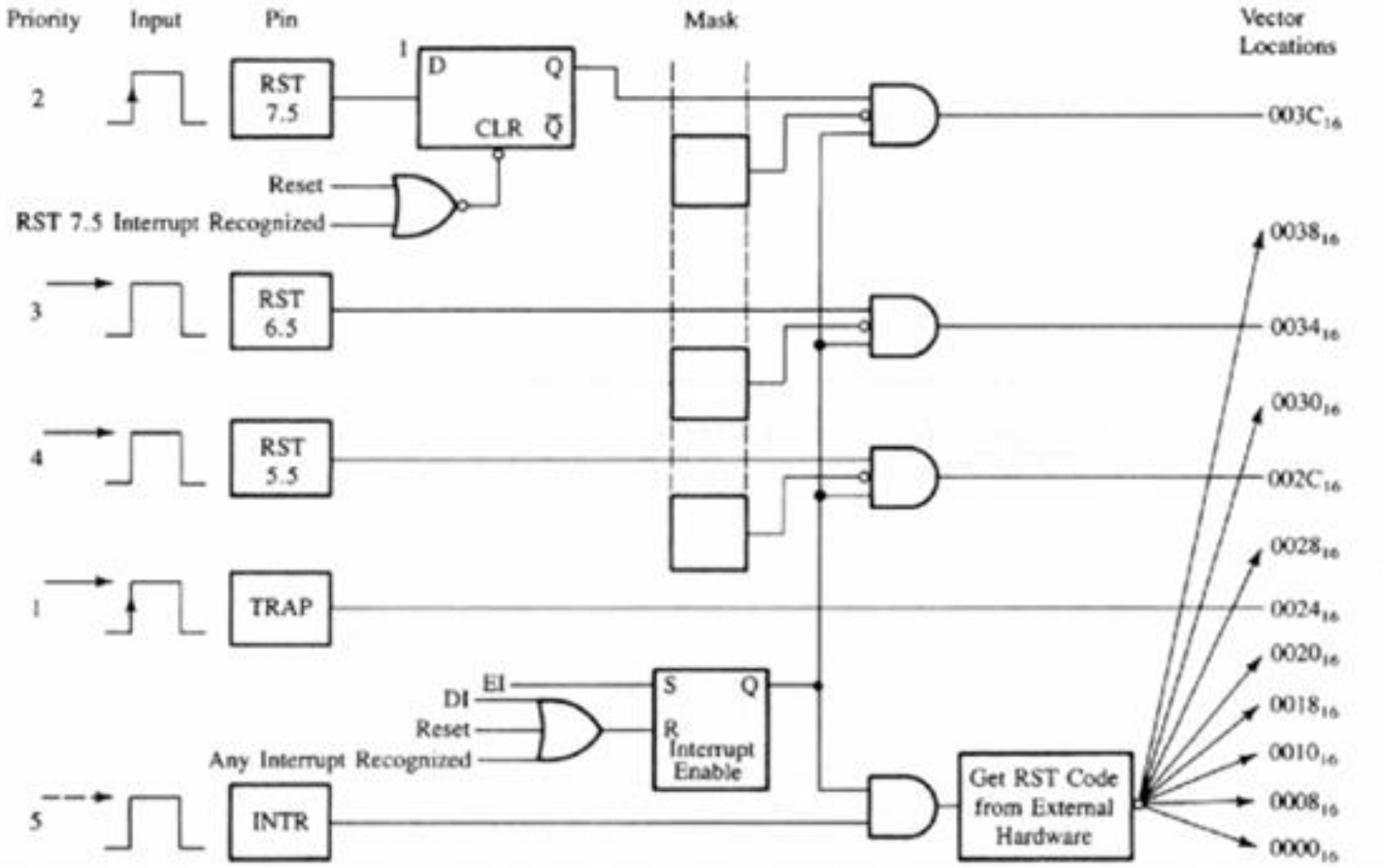
  - The vectors for these interrupt fall in between the vectors for the RST instructions. That's why they have names like RST 5.5 (RST 5 and a half).

8085 Vector Interrupts

Masking RST5.5, RST 6.5 and RST 7.5

- These three interrupts are masked at two levels:
  - Through the Interrupt Enable flip flop and the EI/DI instructions.
    - The Interrupt Enable flip flop controls the whole maskable interrupt process.
  - Through individual mask flip flops that control the availability of the individual interrupts.
    - These flip flops control the interrupts individually.

Vectored Interrupts



8085 Vector Interrupts

The 8085 Maskable/Vectored Interrupt Process

1. The interrupt process should be enabled using the EI instruction.
2. The 8085 checks for an interrupt during the execution of every instruction.
3. If there is an interrupt, and if the interrupt is enabled using the interrupt mask, the microprocessor will complete the executing instruction, and reset the interrupt flip flop.
4. The microprocessor then executes a call instruction that sends the execution to the appropriate location in the interrupt vector table.

8085 Vector Interrupts

The 8085 Maskable/Vectored Interrupt Process

5. When the microprocessor executes the call instruction, it saves the address of the next instruction on the stack.
6. The microprocessor jumps to the specific service routine.
7. The service routine must include the instruction EI to re-enable the interrupt process.
8. At the end of the service routine, the RET instruction returns the execution to where the program was interrupted.

Topic-9 Objective

Name of the Topic	Objective of the topic	Mapping with CO
8259 programmable interrupt controller	To understand the Vector Interrupts	CO1

8259 Programmable Interrupt Controller

- PIC works as an overall manger in an interrupt driven data transfer scheme
- It accepts request from peripheral devices , determines their priority and then executes the highest priority interrupt
- It is a programmable interrupt controller which can manage 8 levels of interrupt
- It is compatible with 8086
- It can be configured in Master Slave or cascade mode to handle upto 64 interrupts
- It can be initialized in various priority modes like :

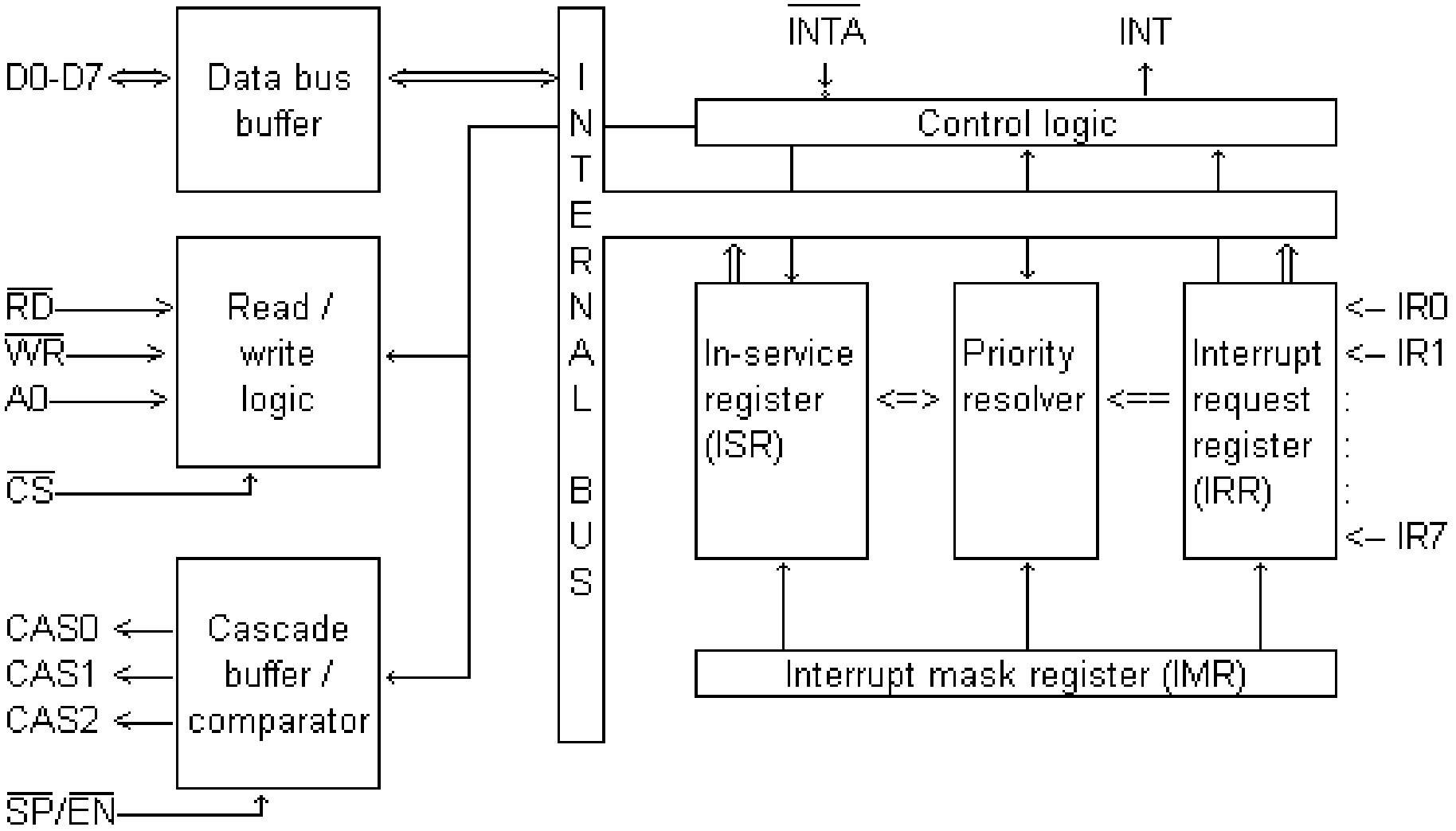
a) Fixed priority

b) Automatic rotating priority

c) Specific priority

Cont'd.

8259 internal block diagram



Cont'd.

The PIC can be divided into 4 main sections :

1. Interrupt and Control logic section
2. Data bus buffer
3. RD/WR control logic section
4. Cascade buffer/ Comparator section

Cont'd.

i. Interrupt Request:

- 8259 has **8 different Interrupt Request levels IR0-IR7** i.e. 8 different devices can be connected to these 8 IR levels
- These can be **edge triggered or level triggered** and will send interrupt signals to 8259 through these levels.

ii. Interrupt Request Register (IRR)

- It consist of **8-bit register which contains 8 FFs** D7-D0 corresponding to 8 IR levels IR7-IR0
- When any device gives interrupt signal on IR level , then the corresponding bit of IRR is set otherwise it remains 0

iii. Interrupt Mask Register (IMR)

- It contains **8-bit register which contains 8 FFs** corresponding to 8 pins IR7-IR0 . The programmer needs to transfer 8-bit data into this IMR
- If any of the IMR levels is to be **enabled** then corresponding bit of IMR should be **set**.
- To keep any of the IR levels **disabled**, the programmer has to transfer **logic 0** in the corresponding FF of IMR

Cont'd.

iv. Control Logic

- The function of this block is to accept O/P commands from the CPU
- It contains the initialization command word register and operation command register which stores various control formats for the device operation
- This block has 2 pins INTR and INTA'. **INTR is an O/P pin** which is connected to INTR pin of 8085 MP.
- **INTA' is an I/P pin** which is connected to INTA' of 8085 pin of MP
- After receiving signal at INTA' from MP , **8259 places the opcode for Call instruction on the system data bus** and on further receiving INTA', it provides the address of ISS on the system data bus

Cont'd.

2. Data bus buffer

- It is an 8-bit bidirectional buffer
- It is used to interface the 8259 with system data bus.

3. RD/WR control logic section

It consist of following pins:

i. CS'

- A low on this line enables 8259
- No read write operation can be performed when the device is selected

ii. WR'

- A low on this I/P enables the **CPU(8085) to write control words to 8259**

iii. RD'

- A low on this I/P enables **8259 to send status** of ISR , IRR and IMR on the data bus

Cont'd.

4. Cascade buffer/ Comparator section

- It consist of 3 cascade signals CAS0, CAS1 and CAS2 and SP'/EN' (Slave Program / Enable Buffer
- This section is used to send signals from Master O/P to Slaves when multiple devices are cascaded
- The signals on these lines indicate the slave selected (000 to 111)
- SP'/EN' is a dual function pin
- When 8259 is not in buffered mode
- This pin programs the device as master if it is 1 otherwise it is slave
- When 8259 is in buffered mode, this pin acts like a O/P that controls the data bus transceivers
- When only one 8259 is used cascade lines are left open
- 8259 can be cascaded to other 8259s to give a total of 64 interrupt pins
- The one which is connected to MP is called Master and others are known as Slaves

Operation of 8259

1. When any one or more than one interrupting device given the interrupt signal then the corresponding bits of Interrupt Request Register (IRR) is set
2. Then the Priority Resolver (PR) reads 8bit data of IRR in order to obtain the info of interrupt signal received. Priority Resolver then checks Interrupt Mask Register (IMR) bit corresponding to the highest priority active IR level. If this IR level is masked then 8259 checks IMR bit corresponding to next priority active IR level
3. If this highest priority active IR level is not masked, then 8259 checks ISR. If all the 8-bit of ISR are 0 or any lower priority IR level bit is set then 8259 gives 8259 gives INT=1 . If the same IR level or higher priority level of IR bit of ISR is set then 8259 doesn't give INT=1
4. If 8259 gives INT=1, then the MP receives INTR=1 and then gives first pulse on INTA' pin

Cont'd.

5. When 8259 receives 1st INTA' pulse, then 8259 resets the corresponding IR level bit of IRR and sets corresponding IR level bit of ISR . 8259 then gives opcode of CALL instruction on data pins D7-D0
6. MP then transfers this CALL opcode into IR and decodes it. As CALL is 3 byte instruction, hence MP gives 2 more pulse on INTA' pin
7. In response to 2nd and 3rd INTA' pulse, 8259 gives LSB/MSB of CALL address on data pins. So, MP executes the instruction " CALL address". Hence, MP executes program of the corresponding interrupting device stored at this address

Quiz

- The interrupt control logic

a) manages interrupts

b) manages interrupt acknowledge signals

c) accepts interrupt acknowledge signal

d) all of the mentioned

- The register that stores the bits required to mask the interrupt inputs is

a) In-service register

b) Priority resolver

c) Interrupt Mask register

d) None

Quiz

- The register that stores all the interrupt requests in it in order to serve them one by one on a priority basis is

a) Interrupt Request Register

b) In-Service Register

c) Priority resolver

d) Interrupt Mask Register

- The number of hardware interrupts that the processor 8085 consists of is

a) 1

b) 3

c) 5

d) 7

Quiz

- The interrupt control logic manages

a) Interrupts
b) Interrupt acknowledge signals
c) Accept interrupt acknowledge signals
d) All of these

- In cascaded mode the no. of vectored interrupts provided by 8259 is

a) 4
b) 8
c) 16
d) 64

Recap

- Features of 8259 PIC
- Block Diagram PIC
- Operation of 8259 PIC
- Modes of 8259 PIC

Topic Links

1. https://www.youtube.com/watch?v=Us_e3I7Mfnk
2. https://www.youtube.com/watch?v=Tbur5vgIPuk
3. https://www.youtube.com/watch?v=VAChQ9n-xSI

Expected Questions

1. Explain 8085 interfacing with memory Chips.
2. Explain Timing Diagram for Memory Read Machine Cycle.
3. Draw and explain the timing diagram of memory write cycle with example.
4. Draw and explain the timing diagram of opcode fetch cycle.
5. Explain 8085 interrupts.
6. Explain 8085 interfacing with keyboard.
7. What is 8259 Programmable interrupt controller?
8. Differentiate between I/O mapped I/O and memory mapped I/O.
9. Describe basic interfacing concepts in detail.

Old Question Papers

B.TECH.

Theory Examination (Semester-VI) 2015-16

MICROCONTROLLER

Time : 3 Hours ***Max. Marks : 100***

Section-A

1. Attempt all parts. All parts carry equal marks. Write answer of each part in short. (2×10=20)

(a) What is the role of processor reset and system reset?

(b) Define the term RISC and CISC.

(c) Draw the PSW format of 8051 microcontroller & state various conditions of flags.

(d) Is it possible to write PSW register? In 8051, which register bank conflicts with the stack?

(e) How Embedded Microcontrollers are differing than Embedding Microprocessor?

(f) How will you assign Counter to count an external event?

1 (1) P.T.O.

(g) Compare microprocessors and microcontrollers.

(h) If you write to SBUF in serial mode 1, nothing is being transmitted. What may be the probable reason for this.

(i) What is the advantage of two or three level handling of interrupts?

Old Question Papers

2. Attempt any five questions from this section. (10×5=50)

(a) Is it possible to address 8051 individual bits? What are the addresses of bit addressable locations? How is bit addressing distinguished from the byte-wise addressing by the 8051 microcontroller? Explain.

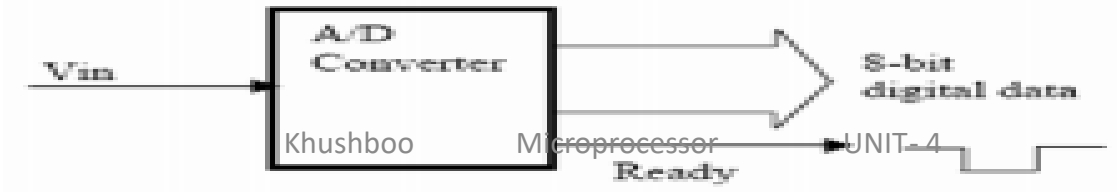
(b) Generate a square wave with an ON time of 5 ms and on OFF time of 7ms on all pins of port0. Assume an XTAL of 22MHz.

(c) Enlist Draw generalized functional block diagram of a 8051 microcontroller specifying each block.

(d) A simple 8-bit analog-to-digital converter device, as shown, is to be interfaced to an 8051 microcomputer.

1 (2) P.T.O.

The READY line goes low when conversion data is available. The READY line should be used to interrupt the 8051 microcontroller.



Old Question Papers

ii. Write an assembly language program which will capture 250 data samples from the A/D converter and store this data in XDATA memory. The program is to be interrupt driven.

(e) Draw a simple block diagram for the transmitter section of an 8051 UART which supports 9-bit data transmission. Briefly explain the function of each block in your diagram.

(f) What does it mean when it is said that a given sensor has a linear output? Draw 8051 connection to ADC0848 and Temperature Sensor.

(g) Draw the architecture of 8096 microcontroller and compare with 8051 architecture

(h) Describe MC68HC11 microcontroller main Features with its Architecture.

Section-C

Attempt any two questions from this section. (15×2=30)

3. (i) Assume that XTAL=11.0592 MHz. What values do we need to load timer's registers if we want to have time delay of 2ms? Show the program for Timer 1 to create a pulse width of 2ms on P2.3.

(ii) Assume that bit P2.2 is used to control the outdoor light and bit P2.5 to control the light inside a building .Show how to turn on the outside light & turn off the inside one.

References

- Ramesh Gaonkar, “Microprocessor Architecture Programming and Applications with the 8085”, 5th Edition Penram International Publication , (India) Pvt. Ltd. 2009.
- D.V. Hall : Microprocessors Interfacing , TMH, 2nd Edition, 2006.

Thank You