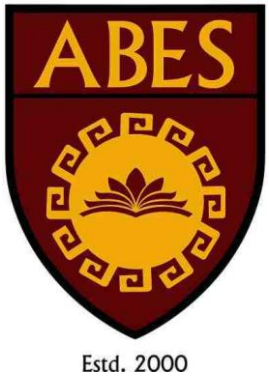
ABES ENGINEERING COLLEGE, GHAZIABAD

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING



COURSE MATERIAL

SUBJECT NAME: **Microprocessr**
SUBJECT CODE: **KCS-403**

FACULTY MENTOR: Dr. Sanjay Kr. Singh

TEAM MEMBERS: Ms. Ranjeeta Yadav
Ms. Tania Gupta
Mr. Rajeev Kumar Pandey

CHAPTER-1: INTRODUCTION TO MICROPROCESSOR

1.1 Introduction to Microprocessor

- 1.1.1 What is Microprocessor (Explaining Why/What)
- 1.1.2 Evolution of microprocessor
- 1.1.3 Need of Microprocessor?
- 1.1.4 Working of the Microprocessor.
- 1.1.5 Characteristics of Microprocessor.
- 1.1.6 Advantages & Disadvantages of Microprocessor
- 1.1.7 Applications of Microprocessor.
- 1.1.8 Circuit implementation in industries
- 1.1.9 University Questions Related to the topic

1.2 Microprocessor architecture and its operations

- 1.2.1 Microprocessor Initiated Operations
- 1.2.2 Internal Data Operations
- 1.2.3 Peripheral Or Externally Initiated Operations
- 1.2.4 University Questions Related to the topic

1.3 The 8085 MPU

- 1.3.1 Architecture
- 1.3.2 Pins and signals
- 1.3.3 Demultiplexing of AD0-AD7 Using IC 74LS373
- 1.3.4 Generation of control Signals
- 1.3.5 University Questions Related to the topic

1.4 Timing Diagrams of 8085

- 1.4.1 What is timing diagram (Explaining Why/What)?
- 1.4.2 Need of timing diagram
- 1.4.3 Working of Timing Diagram
  - 1.4.3.1 T-State
  - 1.4.3.2 Machine Cycle
  - 1.4.3.3 Instruction Cycle
- 1.4.4 Types of Machine Cycle
  - 1.4.4.1 Opcode Fetch Machine Cycle
  - 1.4.4.2 Memory Read Machine Cycle
  - 1.4.4.3 Memory Write Machine Cycle
  - 1.4.4.4 IO Read Machine Cycle
  - 1.4.4.5 IO Write Machine Cycle
- 1.4.5 Examples of Timing Diagram
- 1.4.6 University Questions Related to the Topic

1.5 Logic devices for interfacing

- 1.5.1 What are Logic Devices

1.5.2 Need of Logic Devices
1.5.3 Types of Logic Devices
1.5.3.1 Tri-State Devices
1.5.3.2 Buffer
1.5.3.3 Decoder
1.5.3.4 Encoder
1.5.3.5 Latches (Registers)
1.5.4 Applications of Logic Devices for interfacing
1.5.5 University Questions related to the topic

1.6 Memory interfacing

1.6.1 What is Memory Interfacing
1.6.2 Need of Memory Interfacing
1.6.3 Basic Terms used for Memory
1.6.4 Classification of Memory
1.6.5 Memory Structure
1.6.6 Memory Mapping Basics
1.6.7 Address Decoding
1.6.8 Applications of Memory Mapping with Microprocessor
1.6.9 University Questions Related to the topic

1.7 Interfacing Input Output devices

1.7.1 What is Interfacing
1.7.2 Need of Interfacing
1.7.3 Input Ports
1.7.4 Output Ports
1.7.5 Types of I/O ports
1.7.6 Application of input and output interfacing devices
1.7.7 University Questions Related to the topic

1.8 Memory mapped I/O

1.8.1 What is memory mapped I/O?
1.8.2 Need of memory mapped I/O
1.8.3 How does it work?
1.8.4 Applications of memory mapped I/O
1.8.5 University Questions Related to the topic

1.1 INTRODUCTION TO MICROPROCESSOR

1.1.1 WHAT IS MICROPROCESSOR?

"Microprocessor is an internal part of a digital computer. In layman term, microprocessor acts as a brain of a computer system. As we know that human body parts such as hands, legs do not work without brain, similarly microprocessor cannot perform useful work without external memory, input & output devices which combines to make a self-dependent computational system. This is also known as Microcomputer."

OR

"A *microprocessor* is a multipurpose, programmable, and clock-driven, register based electronic device that reads binary instructions from storage device called *memory*, accepts binary data as input *from input device* and processes data according to those instructions and provide result as *output*."

The main parts of a digital computer are CPU (Central Processing Unit), Memory, Input and Output devices. The block diagram of digital computer is shown in figure 1.

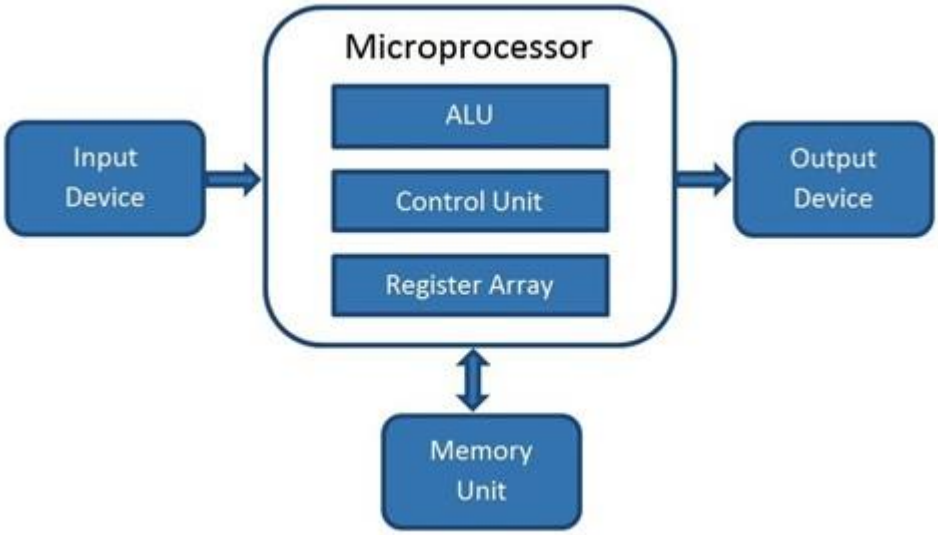


Fig.1 Microprocessor-based system (Microcomputer)

1.1.2 EVOLUTION OF MICROPROCESSOR

The microprocessor has become a more essential part of many gadgets. The evolution of microprocessor was divided into five generations such as first, second, third, fourth, and fifth generation and the characteristics of these generations are discussed below.

First Generation (4 - bit Microprocessors)

- The first-generation microprocessors were introduced in the year 1971-1972 by Intel Corporation. It was named Intel **4004** since it was a 4-bit processor.
- It was a processor on a single chip. It could perform simple arithmetic and logical operations such as addition, subtraction, Boolean OR and Boolean AND.
- I had a control unit capable of performing control functions like fetching an instruction from storage memory, decoding it, and then generating control pulses to execute it.

Second Generation (8 - bit Microprocessor)

- The second-generation microprocessors were introduced in 1973 again by Intel. It was a first 8 - bit microprocessor which could perform arithmetic and logic operations on 8-bit words.
- It was **Intel 8008**, and another improved version was **Intel 8088.**

Third Generation (16 - bit Microprocessor)

- The third-generation microprocessors, introduced in 1978 were represented by **Intel's 8086, Zilog Z800 and 80286**, which were 16 - bit processors with a performance like minicomputers.

Fourth Generation (32 - bit Microprocessors)

- Several different companies introduced the 32-bit microprocessors, but the most popular one is the **Intel 80386**.

Fifth Generation (64 - bit Microprocessors)

- From 1995 to now we are in the fifth generation. After 80856, Intel came out with a new processor namely Pentium processor followed by **Pentium Pro CPU**, which allows multiple CPUs in a single system to achieve multiprocessing.
- Other improved 64-bit processors are **Celeron, Dual, Quad, Octa Core processors**.

History of INTEL Microprocessors

Table 1 Family of Intel Microprocessor

Microprocessor	Year of Invention	Word Length/ Data bus	Address Bus	Memory addressing Capacity	Pins	Clock
4004	1971	4-bit	10-bit	640 Bytes	16	108 KHz
8008	1972	8-bit	8-bit	16KB	18	880 KHz
8080	1974	8-bit	16-bit	64 KB	40	2 MHz
8085	1976	8-bit	16-bit	64 KB	40	3-6 MHz (3.14 MHz)
8086	1978	16-bit	20-bit	1MB	40	5-10 MHz
8088	1979	8-bit	20-bit	1MB	40	5-8 MHz
80186	1982	16-bit	20-bit	1MB	68	5-8 MHz
80286	1983	16-bit	24-bit	16MB real, 4 GB virtual	68	6-12.5 MHz
80386	1985	32-bit	32-bit	4GB real, 64TB virtual	132	20-33 MHz
80486	1989	32-bit	32-bit	4GB real, 64TB virtual	168	25-100 MHz
Pentium	1993	32-bit/ 64-bit data bus	32-bit	4GB real	237	60-200
Pentium Pro	1995	32-bit/ 64-bit data bus	36-bit	64GB real	387	150-200 MHz
Pentium II	1997	32-bit/ 64-bit data bus	36-bit	64GB real	-	233-400 MHz
Pentium III	1999	32-bit/ 64-bit data bus	36-bit	64GB real	370	600-1.3 MHz
Pentium 4	2000	32-bit/ 64-bit data bus	36-bit	64GB	423	600-1.3 GHz
Itanium	2001	64-bit	64-bit	-	423	733 MHz-1.3 GHz

History of Motorola Microprocessors

Table 2 Family of Motorola Microprocessor

Microprocessor	Year of Invention	Word Length/ Data bus	Address Bus
6800	1974	8-bit	10-bit
6809	1979	8-bit	8-bit
68000	1979	16/32 bit	16-bit
68008	1982	8/32 bit	16-bit
68010	1983	16/32 bit	20-bit
68012	1983	16/32 bit	20-bit
68020	1984	32-bit	20-bit
68030	1987	32-bit	24-bit
68040	1989	32-bit	32-bit

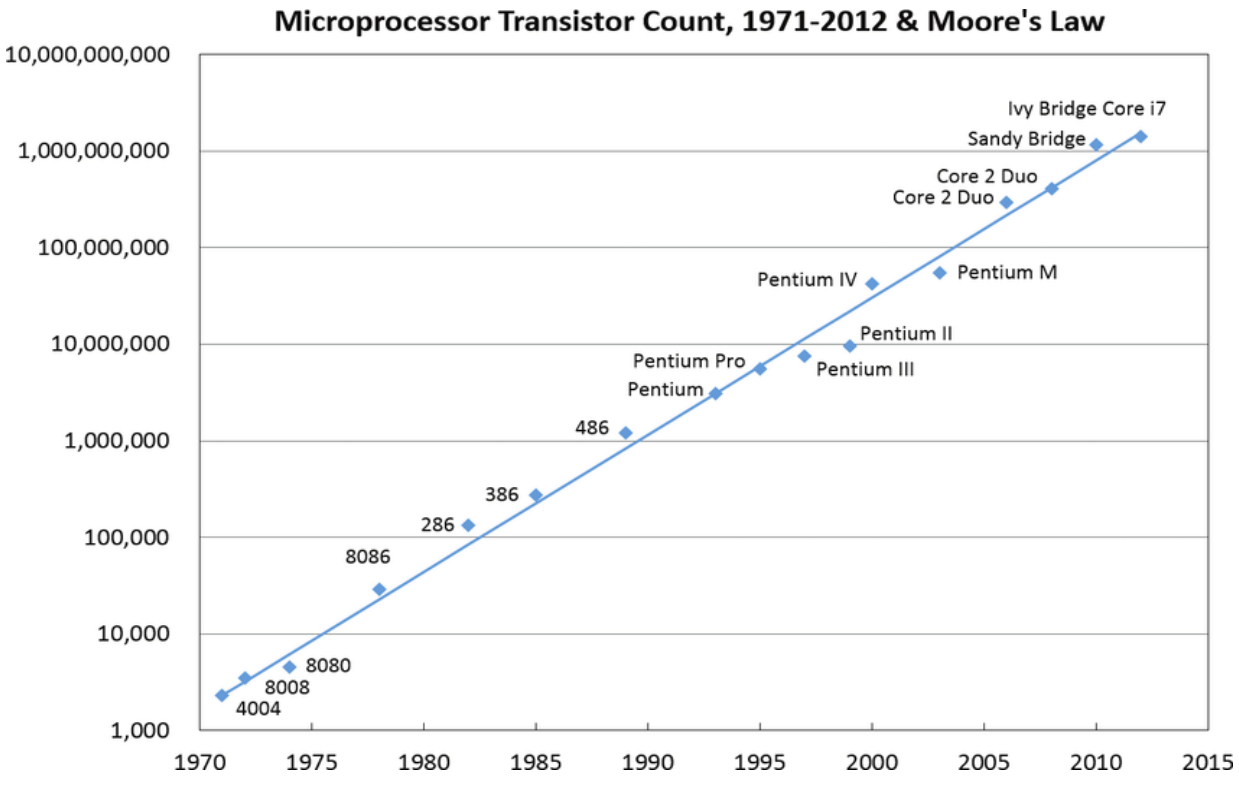


Fig. 2 Microprocessor transistor count

Technology Used:

The semiconductor manufacturing technologies used for chips are:

- Transistor-Transistor Logic (TTL)
- Emitter Coupled Logic (ECL)
- Complementary Metal-Oxide Semiconductor (CMOS)

Classification of Microprocessors:

Based on their specification, application and architecture microprocessors are classified.

Based on size of data bus:

- 4-bit microprocessor - INTEL 4004
- 8-bit microprocessor - INTEL 8008, 8080, 8085, MOTOROLA 6800
- 16-bit microprocessor - 8086, 8088, Z800, 80186, 80286
- 32-bit microprocessor - INTEL 80386, 80387, 80486, PENTIUM, PENTIUM PRO
- 64-bit microprocessor - INTEL core 2, core i3, i5, i7

Based on application:

- **General-purpose microprocessor**- used in general computer system and can be used by programmer for any application.
  Examples- 8085 to Intel Pentium
- **Microcontroller**- microprocessor with built-in memory and ports and can be programmed for any generic control application.
  Example-8051
- **Special-purpose processors**- designed to handle special functions required for an application.
  Examples-digital signal processors and application-specific integrated circuit (ASIC) chips

Based on architecture:

- Reduced Instruction Set Computer (RISC) processors
- Complex Instruction Set Computer (CISC) processors
- Von Neumann Architecture
- Harvard Architecture

CISC Processor

CISC stands for **Complex Instruction Set Computer**. The processors are designed to minimise the number of instructions per program and ignore the number of cycles per instructions. The compiler is used to translate a high-level language to assembly level language because the length of code is relatively short, and an extra RAM is used to store the instructions. These processors can do tasks like downloading, uploading, and recalling data from memory. Apart from these tasks these microprocessors can perform complex mathematical calculation in a single command.

The common characteristics of CISC architecture are:

- Complex hardware: complex as well as more addressing modes, variable instruction size.
- Many clock cycles to execute an instruction
- High code density- small program size
- Complex data types

Some of the CISC Processors are:

- 8051 microcontrollers
- IBM 370/168
- VAX 11/780
- Intel 80486

RISC Processor

RISC stands for **Reduced Instruction Set Computer**. These processors are made according to function. They are designed to reduce the execution time by using the simplified instruction set. They can carry out small things in specific commands. These processors complete commands at faster rate. They require only one clock cycle to implement a result at uniform execution time. There are number of registers and a smaller number of transistors. To access the memory location LOAD and STORE instructions are used.

The common characteristics of RISC architecture are:

- Simple hardware: simple and less addressing modes, fix instruction size
- Single clock cycle execution, uniform instruction format
- Low code density- larger program size
- Few data types in hardware
- Emphasis is on software: Compiler design is more complex

Some of the RISC processors are:

- Power PC: 601, 604, 615, 620
- Microchip PIC microcontrollers
- DEC Alpha: 210642, 211066, 21068, 21164
- MIPS: TS (R10000) RISC Processor
- PA-RISC: HP 7100LC

Von-Neumann Architecture

- It has single memory storage to hold both program instructions and data i.e. common program and data space.
- The CPU can either read an instruction or data from the memory one at a time (or write data to memory) because instructions and data are accessed using same bus system.

- The Von Neumann Architecture is named after the mathematician and computer scientist John Von Neumann. The basic organization of memory in this architecture is shown in figure 3.
- The advantage of Von Neumann architecture is simple design of microcontroller chip because only one memory is to be implemented which in turn reduces required hardware.
- The disadvantage is slower execution of a program.
- It is also referred as **Princeton architecture** as it was developed at Princeton University. **Motorola 68HC11 microcontroller** is based on Von Neumann architecture.

Harvard Architecture

- It has physically separate memory storage to hold program instructions and data i.e. separate program and data space. Since it has separate buses to access program and data memory, it is possible to access program memory and data memory simultaneously. The organization of memory and buses in this architecture is shown in figure-3.
- The advantage of a Harvard architecture microcontroller is that it is faster for a given circuit complexity because it offers greater amount of parallelism.
- The disadvantage is that it requires more hardware, because two sets of buses and memory blocks are required.
- **MCS 51 (8051 family) and PIC microcontrollers** are based on Harvard architecture. Each architecture shows in figure-3.

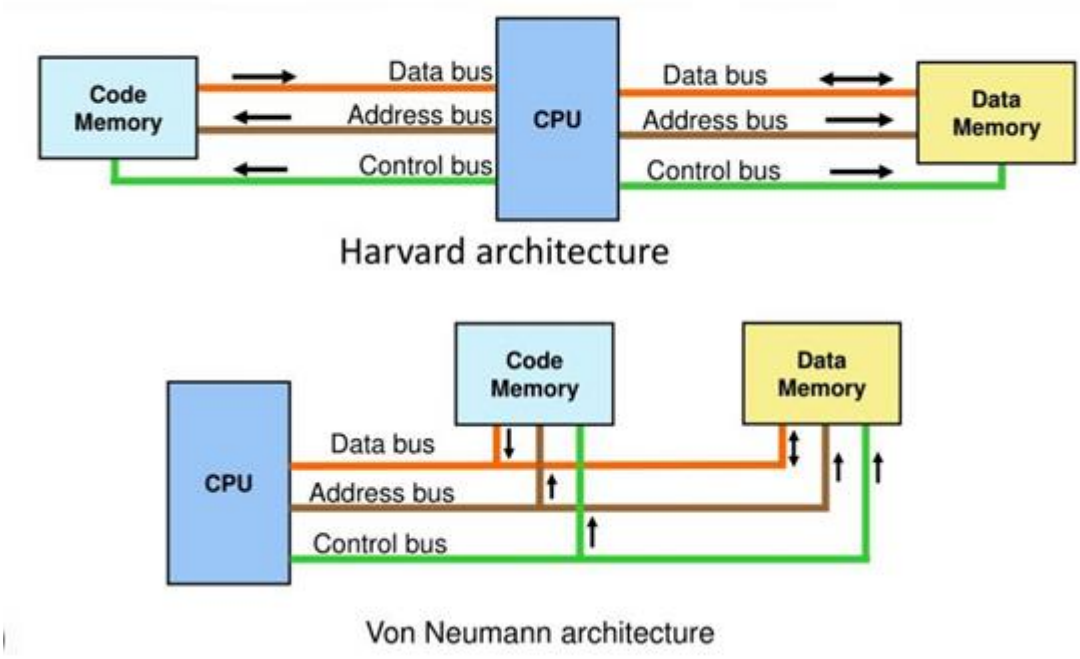


Fig.-3 Von Neumann architecture & Harvard architecture

1.1.3 NEED OF MICROPROCESSOR?

In the beginning, systems did not possess any intelligence. Those early days systems used vacuum tubes and hence these could carry out only few arithmetic operations. It was a difficult task to operate these machines, even difficult to maintain them.

A small chip, now, can handle the work that 50,000 or more vacuum tubes would do in fraction of the time taken by those tubes. These chips are termed as "Microprocessor". Microprocessor is the heart of computer systems.

Nowadays, microprocessor can be seen in almost all types of electronics devices like **mobile phones, printers, washing machines** etc. Microprocessors are also used in advanced applications like radars, satellites, and flights. Due to the rapid advancements in electronic industry and large-scale integration of devices results in a significant cost reduction and increase application of microprocessors and their derivatives.

1.1.4 HOW DOES A MICROPROCESSOR WORK?

The microprocessor follows a sequence: **Fetch, Decode, and then Execute**.

The instructions are stored in the memory in a sequential order. The microprocessor fetches those instructions from the memory, then decodes it and executes those instructions. Later, it sends the result in binary form to the output port. Between these processes, the register stores the temporarily data and ALU performs the computing functions.

Fig.1 shows the block diagram of Microcomputer. Microprocessor consists of an ALU, register array, and a control unit. ALU performs arithmetical and logical operations on the data received from the memory or an input device. Register array consists of registers identified by letters like B, C, D, E, H, L and accumulator. The control unit controls the flow of data and instructions within the computer.

Microprocessor communicates and operates in binary numbers 0 and 1. The set of instructions in the form of binary patterns is called a ***machine language*** and it is difficult for us to understand. Therefore, the binary patterns are given abbreviated names, called mnemonics, which forms the ***assembly language***. The conversion of assembly-level language into binary machine-level language is done by using an application called ***assembler.***

The microprocessor contains millions of tiny components like transistors, registers, and diodes that work together.

1.1.5 CHARACTERISTICS OF MICROPROCESSOR

Cost-effective: The microprocessor chips are available at low prices and results its low cost.

Size: The microprocessor is of small size chip, hence is portable.

Low Power Consumption: Microprocessors are manufactured by using metal-oxide semiconductor technology, which has low power consumption.

Versatility: The microprocessors are versatile as we can use the same chip in a number of applications by configuring the software program.

Reliability: The failure rate of an IC in microprocessors is very low, hence it is reliable.

1.1.6 ADVANTAGES & DISADVANTAGES OF MICROPROCESSOR

Advantages: -

- Microprocessor is the general-purpose electronic processing devices which can be programmed to execute several tasks.
- Microprocessor speed, which is measured in **hertz**. For instance, a microprocessor with GHz can perform 3 billion tasks per second.
- Microprocessor is that it can quickly move data between the various memory locations.

Disadvantages: -

- The microprocessor has a limitation on the size of data.
- Most of the microprocessor does not support floating point operations.
- The main disadvantage is it's over heating physically.
- The microprocessor is does not have any internal peripheral like ROM, RAM and other I/O devices.

1.1.7 APPLICATIONS OF MICROPROCESSOR

1. Calculators
2. Accounting system
3. Games machine
4. Complex Industrial Controllers
5. Traffic light Control
6. Data acquisition systems
7. Multiuser, multi-function environments
8. Military applications
9. Communication systems

1.1.8 CIRCUIT IMPLEMENTATION IN INDUSTRIES

Some industrial items which use microprocessors technology include: *cars, boats, planes, trucks, heavy machinery, elevators, gasoline pumps, credit-card processing units, traffic control devices, computer servers, most high tech medical devices, surveillance systems, security systems, and even some doors with automatic entry.*

EXAMPLE-1 **Room Temperature control System**

With the help of a microprocessor, the system responds by **turning ON any of the loads (Fan or a heater) automatically depending on the temperature difference**. The Fan is triggered ON when the room temperature is higher than the set temperature and the heater is triggered ON when the room temperature is lower than the set temperature. Figure-4 shows the room temperature control system.

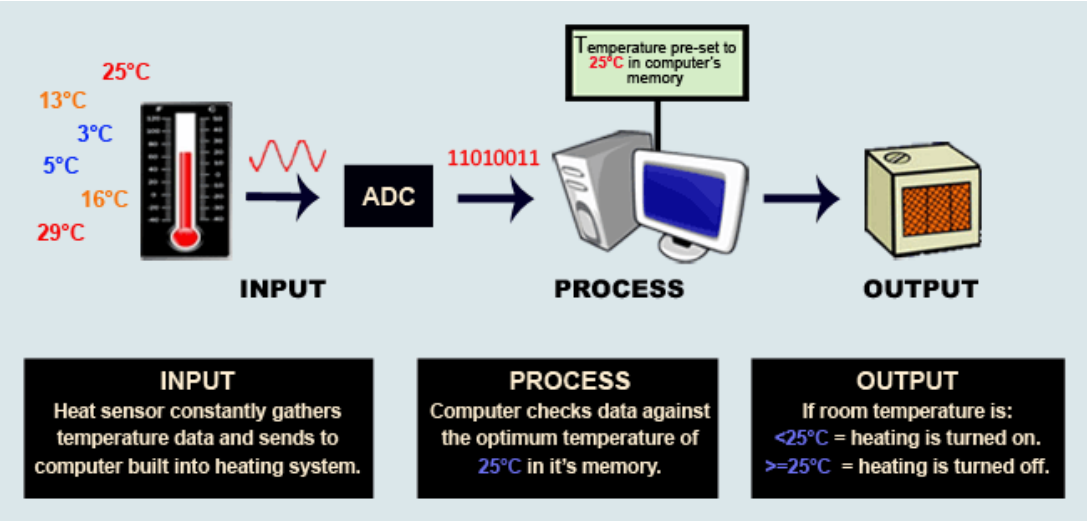


Fig. 4 Room Temperature control system

EXAMPLE-2 **Attendance Recorder**

This a cool project where you will be able to count and keep a record of the total no of students present in the class. This is designed with the help of an 8085 microprocessor. Suppose in a room you keep this device and at the door, the counter is placed which counts the total no of students entering and leaving the room. The students entering are instructed to enter their roll no in the device. Suppose 10 students entered the room and entries their roll no. The subject is boring, and two students left out before the teachers arrive. This attendance recorder keeps the record and displays the entire roll no and no of students present in the class.

1.1.9 UNIVERSITY QUESTIONS RELATED TO THE TOPIC

Two-mark questions

Q-1. Define the term bit, byte, nibble, and word.
Q-2. What is the difference between microprocessor and microcomputer?
Q-3. Draw the block diagram of Microcomputer.
Q-4. What is the heart of computer system?
Q-5. What is Microprocessor?

Five-mark questions

Q-6. Explain the evolution of Microprocessor.
Q-7. Give the difference between machine language and assembly language.
Q-8. What is the Difference between RISC and CISC architecture?
Q-9. What is the difference between Von-Newman and Harvard architecture?

Q-10. Explain the characteristics of Microprocessor.

Ten-mark questions

Q-11. Explain the five applications of Microprocessor.

GATE questions

Q-12. In a microcomputer, wait states are used to
Q-13. In a microprocessor, when a CPU is interrupted, it
Q-14. In an 8085µP system, the RST instruction will cause an interrupt

1.2 MICROPROCESSOR ARCHITECTURE AND ITS OPERATION

The internal logic design of the microprocessor called its "architecture", determine how and what various operations are performed by "MICROPROCESSOR".
Computer system consists of: -

- Microprocessor
- Memory
- Input Devices
- Output Devices

All function performed by microprocessor can by classified in three general categories:

1. **Microprocessor initiated operations**
2. **Internal data operations**
3. **Peripheral (or externally) initiated operations**

To perform these operations, microprocessor needs [logic circuit and control signals].

1.2.1 MICROPROCESSOR INITIATED OPERATIONS

Primarily microprocessor performs four operations: -

a. Memory read (Reads data from memory).
b. Memory writes (Write data into memory).
c. I/O read (Accept data to output device).
d. I/O writes (Sends data to output device).

These operations are part of communication process. Microprocessor performed these functions using sets of buses **[Data bus, Address bus, Control bus].** Microprocessor bus structure shown in figure 5

DATA BUS: 8085 Microprocessor has 8-bit data bus. So, it can be used to carry the 8 bit data starting from 00000000H (00H) to 11111111H (FFH). Here 'H' tells the Hexadecimal Number. It is bidirectional. These lines are used for data flowing in both direction means data can be

transferred and can be received through these lines. The data bus also connects the I/O ports and CPU. It has 8 parallel lines of data bus. So, it can access up to 28 = 256 data bus lines.

ADDRESS BUS: The bus over which the CPU sends out the address of the memory location is known as Address bus. The address bus carries the address of memory location to be written or to be read from. The address bus is **unidirectional.** It means bits flowing occur only in one direction, only from microprocessor to peripheral devices. We can find that how much memory location it can use the formula 2^N. Where N is the number of bits used for address lines. 8085 has 16 address lines. So, it can access up to 2^{16} = 64KB memory locations (0000H-FFFFH).

CONTROL BUS: The control bus is used for sending control signals to the memory and I/O devices. The CPU sends control signal on the control bus to enable the outputs of addressed memory devices or I/O port devices. Some of the control bus signals are as follows: **Memory read, Memory writes, I/O read, I/O write.**

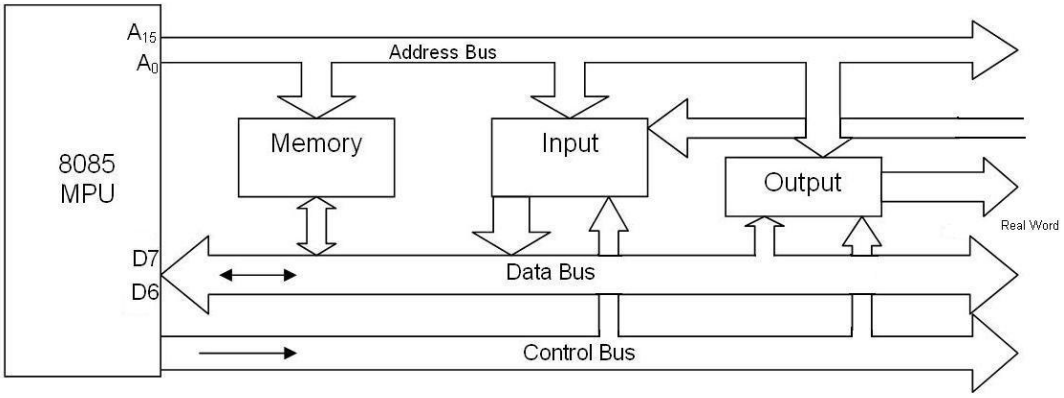


Fig.5 8085 Bus System

The microprocessor needs to perform the following steps: -

a) Identify the peripheral (memory location).
b) Transfer data.
c) Provide timing or synchronization signals.

1.2.2 INTERNAL DATA OPERATIONS

The internal architecture of the 8085/8080A microprocessor determines how and what operation can be performed with the data. These operations are: -

- Store 8-bit data
- Performed arithmetic and logical operations
- Test for conditions
- Sequence the execution of instructions.
- Store data temporarily during execution in the defined R/W memory locations called the stack.

To perform these operations the Microprocessor requires: -

- Registers
- Arithmetic logic unit (ALU) & control logic
- Internal buses (paths for information flow).

1.2.3 PERIPHERAL OR EXTERNALLY INITIATED OPERATIONS

External devices (or signals) can initiate the following operation for which individual pins on Microprocessor chip are assigned: **Reset, Interrupt, Ready, Hold.**

Reset: when reset is activated all internal operations are suspended and the program counter is cleared.
Interrupt: The Microprocessor can be interrupted from normal execution and asked to execute other instructions called **"service routine"** (emergency), Microprocessor resumes its operation after that.
Ready: 8085 has pin called ready, if the signal is low Microprocessor enters into wait state, this signal used to synchronize slower peripherals with Microprocessor.
Hold: when hold pin activated by external signal Microprocessor relinquishes control buses and allows the external peripheral to use the. For example: Hold signal is used in direct memory access data transfer.

1.2.4 UNIVERSITY QUESTIONS RELATED TO THE TOPIC

Two-mark questions

Q-1. What is System Bus?
Q-2. What is Address Bus?
Q-3. What is Data Bus?

Five-mark questions

Q-4. Define the term data bus, address bus and control bus.

1.3 8085 MICROPROCESSOR

The main features of 8085 Microprocessor are:

- It is an **8-bit microprocessor**.
- It is manufactured with N-MOS technology.
- **Data bus** is a group of **8 lines** D0 - D7.
- It has **16-bit address bus** and hence can address up to **2^{16} = 65536 bytes (64KB) memory locations** through A0-A15.
- It has **8-bit input/output address**. Hence it can access **2^8= 256 input/output ports**.
- The first 8 lines of address bus and 8 lines of data bus are multiplexed AD0 - AD7.
- It requires a signal **+5V** power supply and operates at **3 MHZ** clock frequency.
- It consists of internal clock generator. Internal clock generator consists of following tuned circuit: **LC, Crystal, and RC**. The internal clock generator divides oscillator frequency by two and generates clock frequency.

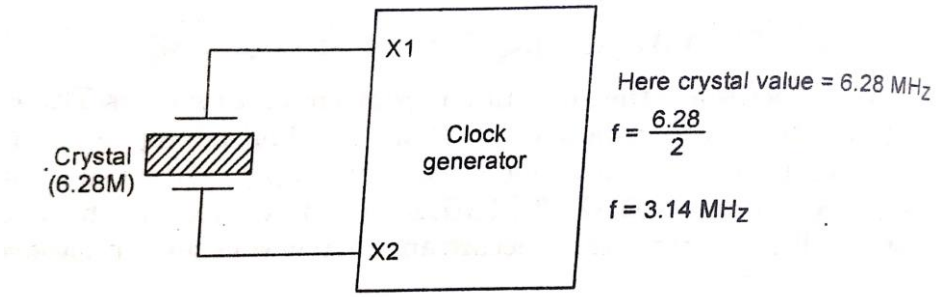


Fig. 6 Clock Generation

- It is enclosed with **40 pins DIP** (Dual in line package)
- Six 8-bit general purpose register arranged in pairs: **BC, DE, HL**.
- It has 8-bit special purpose register **Accumulator (A)**.
- It has 16-bit registers: **Program counter (PC) and Stack pointer (SP).**
- It provides five hardware interrupts:
  TRAP (RST 4.5),
  RST 7.5,
  RST 6.5,
  RST 5.5 and
  INTR.
- It has two serial input/output control pins which allows serial communication: **SID, SOD**
- It has 8-bit ALU (Arithmetic Logic Unit).
- It can support DMA function.
- It supports the instruction with five addressing modes.
  - **Immediate addressing modes**
  - **Register addressing modes**
  - **Direct addressing modes**
  - **Indirect addressing modes**
  - **Implied (Implicit) addressing modes**

1.3.1 ARCHITECTURE OF 8085 MICROPROCESSOR

Figure-7 shows the architecture of 8085. It consists of following blocks:

- Address buffer & Address/Data Buffer
- Registers
- Program status Word (PSW)
- Arithmetic and Logic Unit (ALU)
- Instruction Register and Instruction Decoder
- Increment and Decrement Latch
- Timing and Control Circuit
- Interrupt Control
- Serial I/O Control

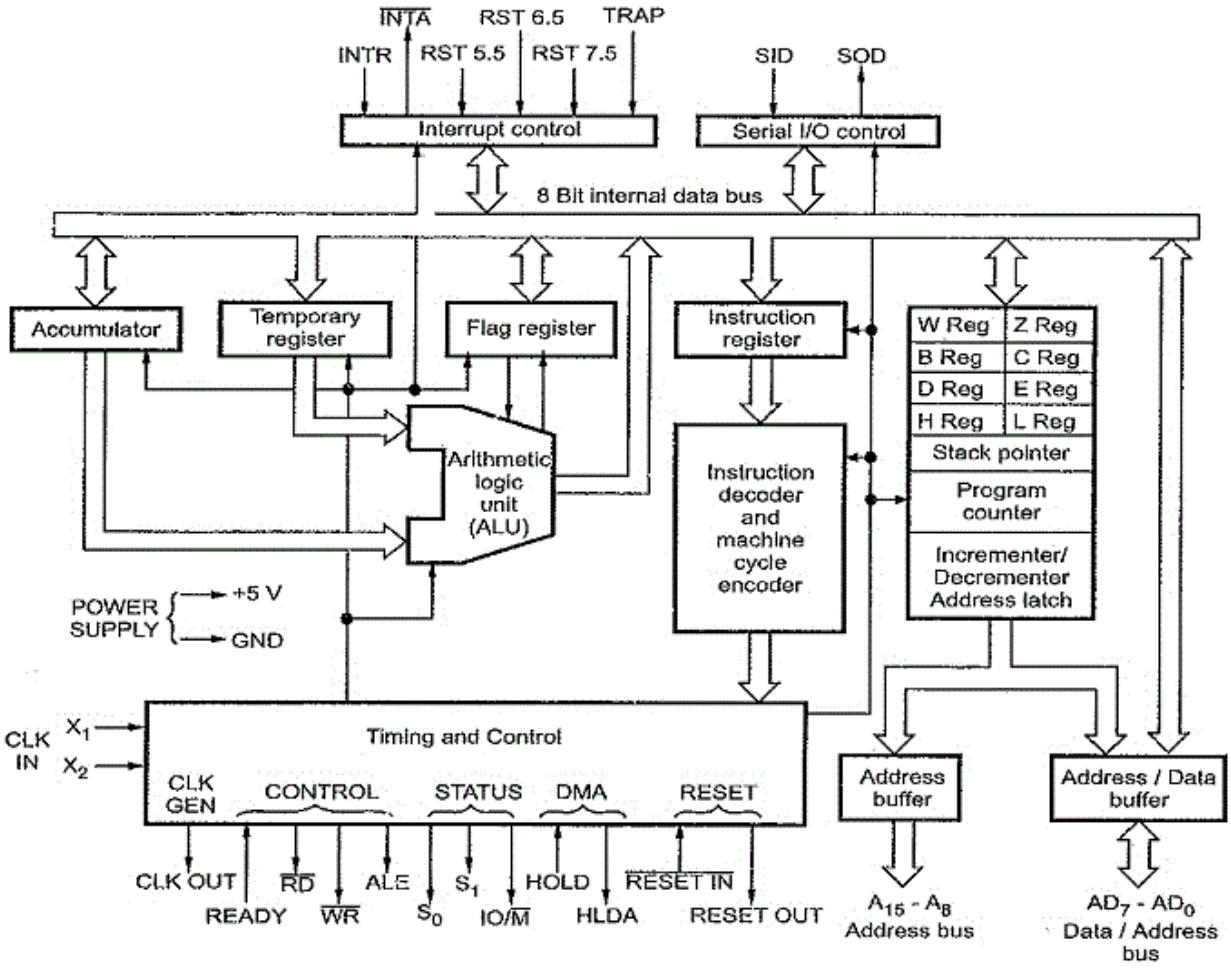


Fig.7 8085 Architecture

Address Buffer & Address/Data Buffer (AD0-AD7)

- The contents of the stack pointer and program counter are loaded into the address buffer and address-data buffer.
- These buffers are then used to drive the external address bus and address-data bus.
- As the memory and I/O chips are connected to these buses, the CPU can exchange desired data to the memory and I/O chips.

- The address-data buffer is not only connected to the external data bus but also to the internal data bus which consists of 8-bits.
- The address data buffer can both send and receive data from internal data bus.

Registers

Register is a combination of Flip flop. Flip flop can store maximum one bit of data at a time. So the combination of flip flop is known as Register. There are two types of register (8- bit & 16-bit) in 8085 microprocessors. These are given in Fig.8.

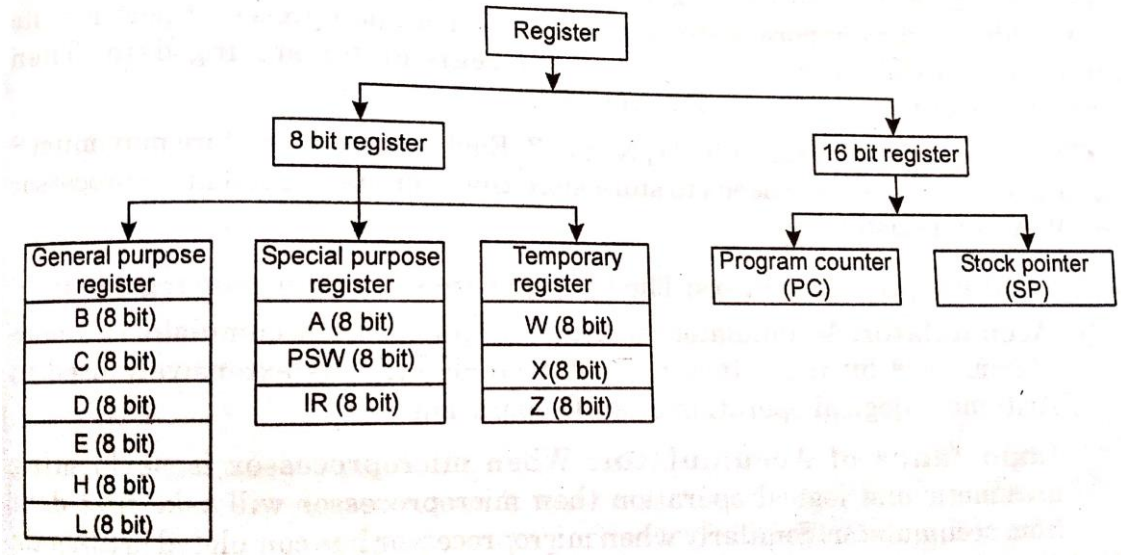


Fig. 8 8085 Microprocessor register

8-bit Registers:

1. General Purpose Register
2. Temporary Register
3. Special Purpose Register

1. **General Purpose Register:** The 8085 has six general-purpose registers to store 8-bit data; these are identified as B, C, D, E, H and L. they can be combined as register pairs - **BC, DE and HL** to perform some 16-bit operations. The programmer can use these registers to store or copy data into the register by using data copy instructions.

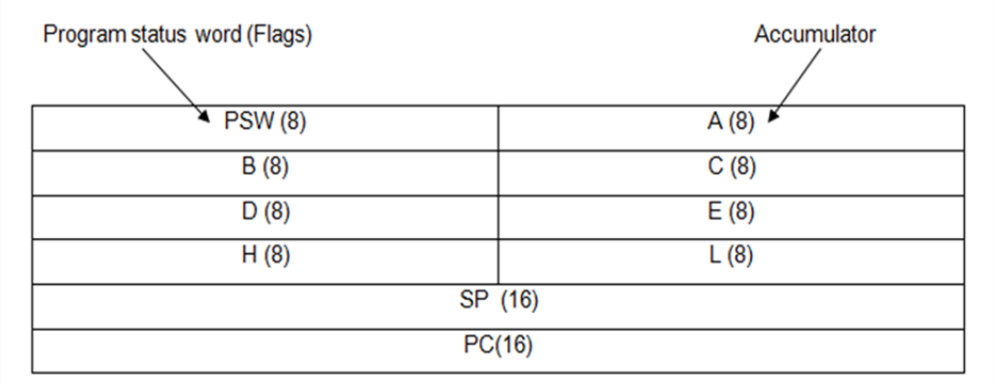


Fig. 9 8085 Programmable Registers

2. **Temporary Registers:** This temporary register can only be accessed by the microprocessor, and it is completely inaccessible to programmers. Temporary register is an 8-bit register. This register is used by control systems to hold operand, intermediate operand, and address of memory and I/O devices temporarily. W & Z, Temporary registers are used for temporary storage. For example W&Z used for swapping purpose.

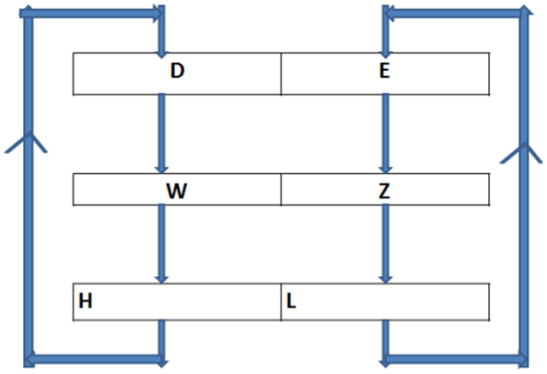


Fig. 10 swapping of content

3. **Special Purpose Registers:** Following are the special purpose registers:

Accumulator: The accumulator is an 8-bit register that is a part of ALU. This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator. The accumulator is also identified as register A.

Flag register: The ALU includes five flip-flops, which are set or reset after an operation according to data condition of the result in the accumulator and other registers. They are called **Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC)** flags. Their bit positions in the flag register are shown in Fig. . The microprocessor uses these flags to test data conditions.

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

Fig. 11 Format of flag register

1. **Sign Flag (S):** After execution of any arithmetic and logical operation, if D7 of the result is 1, the sign flag is set. Otherwise it is reset. D7 is reserved for indicating the sign; the remaining is the magnitude of number. If D7 is 1, the number will be viewed as negative number. If D7 is 0, the number will be viewed as positive number.

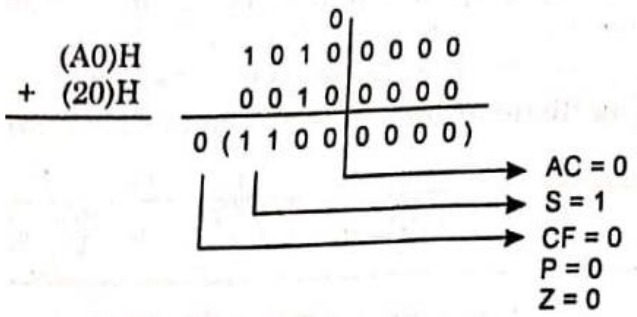
2. **Zero Flag (z):** If the result of arithmetic and logical operation is zero, then zero flag is set otherwise it is reset.

3. **Auxiliary Carry Flag (AC):** If D3 generates any carry when doing any arithmetic and logical operation, this flag is set. Otherwise it is reset.

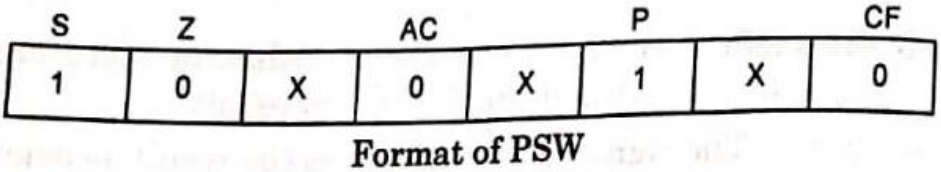
4. **Parity Flag (P):** If the result of arithmetic and logical operation contains even number of 1's then this flag will be set and if it is odd number of 1's it will be reset.

5. **Carry Flag (CY):** If any arithmetic and logical operation result any carry then carry flag is set otherwise it is reset.

Example-1

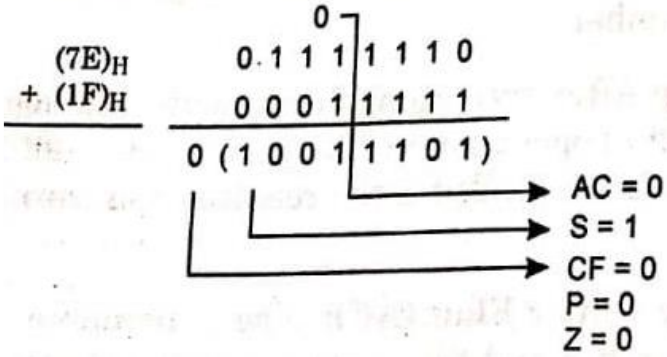


Result:

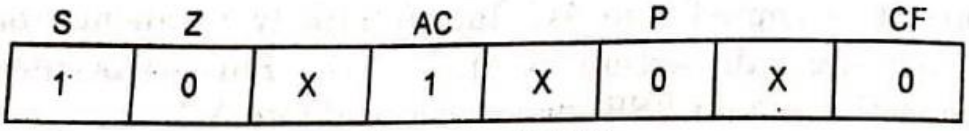


Format of PSW

Example-2

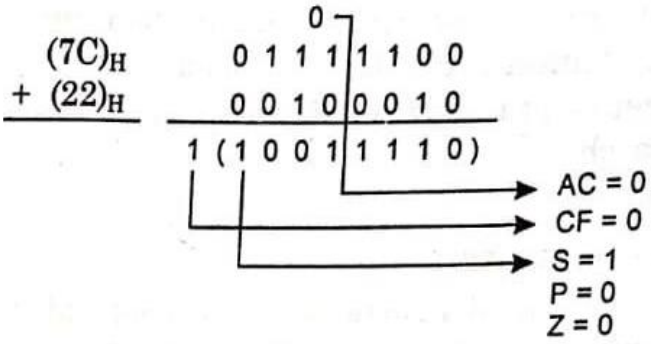


Result:

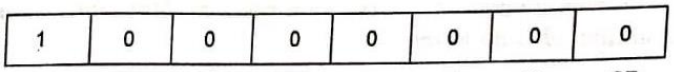


Format of PSW

Example-3



Result:



Format of PSW

16-bit Registers

Program Counter (PC)

This 16-bit register deals with sequencing the execution of instructions. This register is a memory pointer. The microprocessor uses this register to sequence the execution of the instructions. The function of the program counter is to point to the memory address from which the next byte is to be fetched. When a byte is being fetched, the program counter is automatically incremented by one to point to the next memory location.

Stack Pointer (SP)

The stack pointer is also a 16-bit register, used as a memory pointer. It points to a memory location in R/W memory, called stack. The beginning of the stack is defined by loading 16-bit address in the stack pointer.

Arithmetic and Logic Unit (ALU)

The ALU performs the actual numerical and logical operations such as Addition (ADD), Subtraction (SUB), AND, OR etc. It uses data from memory and from Accumulator to perform operations. The results of the arithmetic and logical operations are stored in the accumulator.

Instruction Register and Instruction Decoder

It is an 8-bit register that temporarily stores the current instruction of a program. Latest instruction sent here from memory prior to execution. Decoder then takes instruction and decodes or interprets the instruction. Decoded instruction then passed to next stage.

Increment and Decrement Latch

- The 8-bit contents of a register or a memory location can be incremented or decremented by 1.
- This 16-bit register is used to increment or decrement the content of program counter and stack pointer register by 1.
- Increment or decrement can be performed on any register or a memory location.

Timing and Control Unit

- Timing and control unit is a very important unit as it synchronizes the registers and flow of data through various registers and other units.
- This unit consists of an oscillator and controller sequencer which sends control signals needed for internal and external control of data and other units.
- The oscillator generates two-phase clock signals which aids in synchronizing all the registers of 8085 microprocessor.

- Signals that are associated with Timing and control unit are:
  1. Control Signals: READY, RD(bar), WR(bar), ALE
  2. Status Signals: S0, S1, IO/M (bar)
  3. DMA Signals: HOLD, HLDA
  4. RESET Signals: RESET IN (bar), RESET OUT.

Interrupt Control

- As the name suggests this control interrupts a process.
- Consider that a microprocessor is executing the main program. Now whenever the interrupt signal is enabled or requested the microprocessor shifts the control from main program to process the incoming request and after the completion of request, the control goes back to the main program.
- For example, an Input/output device may send an interrupt signal to notify that the data is ready for input.
- The microprocessor temporarily stops the execution of main program and transfers control to specific special routine known as "Interrupt Service Routine"(ISR).
- After ISR control is transferred back to main program.
- Fig. shows the idea of communication between microprocessor and device with the help of interrupt.
- Interrupt signals present in 8085 are:
  1. TRAP
  2. RST 7.5
  3. RST 6.5
  4. RST 5.5
  5. INTR

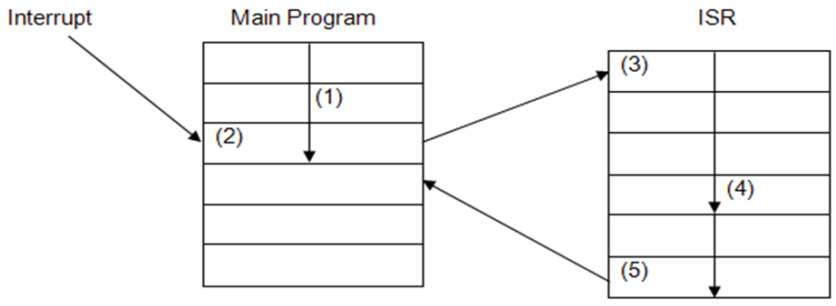


Fig. 12 Interrupt process

- The above four interrupts TRAP is a NON-MASKABLE interrupt control and other three are maskable interrupts.
- A non-maskable interrupt is an interrupt which is given the highest priority in the order of interrupts.
- Suppose you want an instruction to be processed immediately, then you can give the instruction as a non-maskable interrupt.

- Further the non-maskable interrupt cannot be disabled by programmer at any point of time.
- Whereas the maskable interrupts can be disabled and enabled using EI and DI instructions.
  Among the maskable interrupts RST 7.5 is given the highest priority above RST 6.5 and least priority is given to INTR.

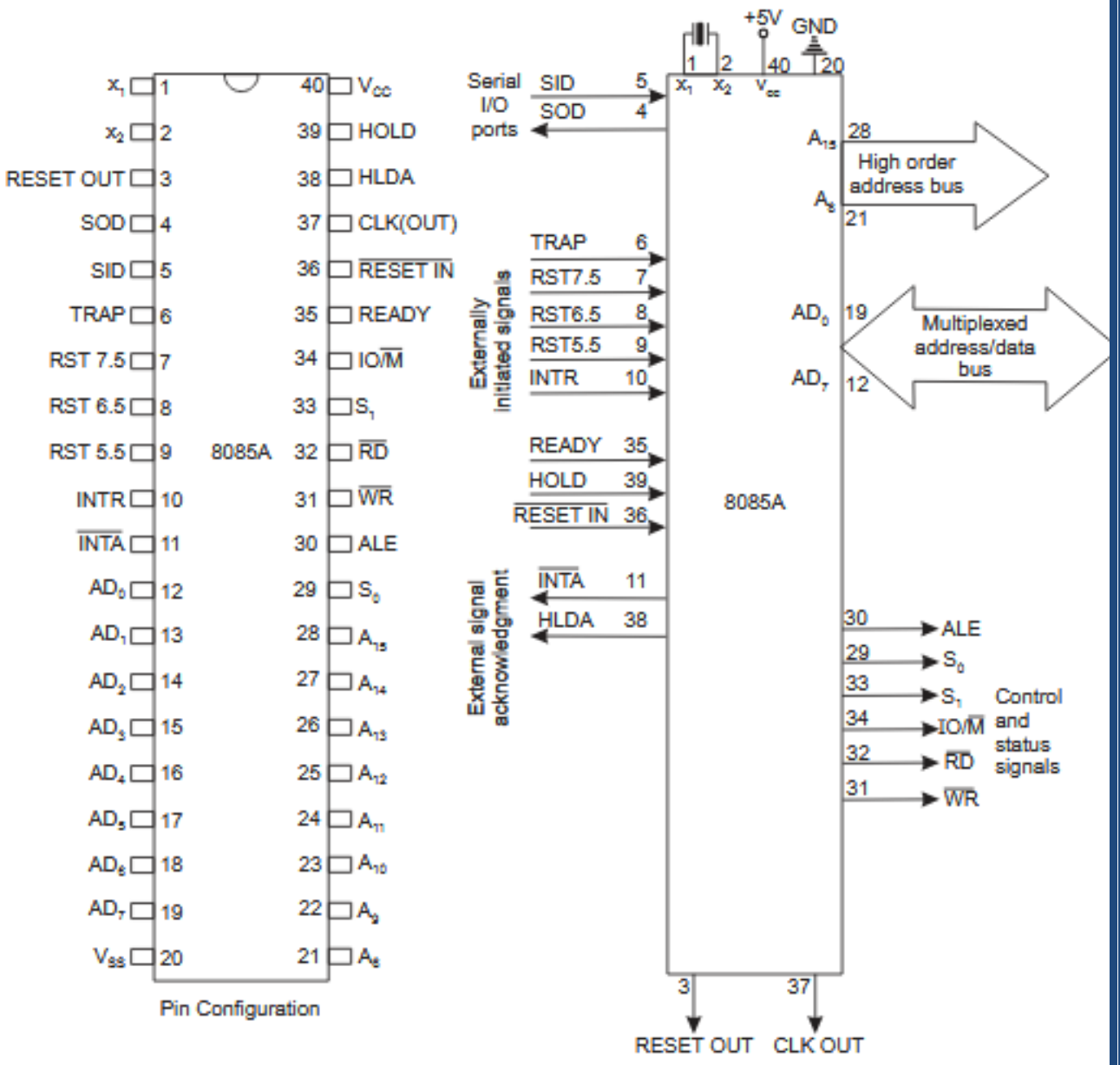
Serial I/O Control

- The input and output of serial data can be carried out using 2 instructions in 8085: SID-Serial Input Data, SOD-Serial Output Data.
- Two more instructions are used to perform serial-parallel conversion needed for serial I/O devices.

1.3.2 8085 PIN DESCRIPTION

The logic pin layout and signal groups of the 8085 microprocessor are shown in Fig.13. The signals of this 40 pin IC is grouped into 7 categories, which are given below:

- Power supply and clock signals
- Data bus
- Address bus
- Serial I/O ports
- Control and status signals
- Interrupts and externally generated signals
- Direct memory access



(a) (b)

Fig.13 (a) Pin Diagram of 8085 (b) Functional diagram of 8085

Power supply and clock signals: In 40 pin configuration, 4 pins are allotted to this particular category.

- **V_{CC}** – Pin number 40 denotes V_{CC}, and an external power supply of + 5 V is provided at this pin.
- **Vss** – Its pin number is 20. This pin shows the grounded connection of the microprocessor.
- **X_1 and X_2** – These are represented by pin number 1 and 2 respectively in the pin configuration. These 2 pins are connected with a crystal or LC network to maintain the internal frequency of the clock generator.
- **CLK (OUT)** – It is the 37th pin of the 8085 IC and acts as the system clock that keeps the record of time duration required by each operation to get completed.

Address Bus: This category contains 8 pins.

The address bus has 16 lines i.e.; it can carry 16 bits at a time. However, out of 16, 8 are multiplexed with the data bus and the leftover 8 are separately shown by pin number 21 to 28 in the pin configuration.

These are used to carry the address of data and instruction from the processor to the memory location and are unidirectional in nature. These are denoted by A_8 **to** A_{15} that represents the 8 MSB of the memory location or input-output address.

Data Bus with multiplexed address bus: This category also contains 8 pins.

The size of the data bus of the 8085 microprocessor is 8 bits. However, to reduce the number of bus lines these 8-bit data bus lines are multiplexed with the 8-bit address bus.
These are shown by pin number 12 to 19. The address bus is denoted by A whereas the data bus is denoted by D. The pin configuration denotes the lower order multiplexed address and data bus bits from AD_0 **to** AD_7.
We have already discussed that the address bus contains the address of the desired memory location from where the data or instruction is to be fetched. While the data bus contains the data or instruction that is needed to be fetched from the memory.

Serial I/O ports: It has basically 2 pins.

- **SID** – SID denotes serial input data pin and its pin is numbered as 5. With this pin, data is serially fed to the processor directly through the input devices.
- **SOD** – SOD denotes serial output data pin and its pin number is 4, in the pin configuration of 8085. Once the data is processed in the microprocessor then this pin represents bit by bit results at the output devices.

Control and status signals: Basically, 6 pins of the pin configuration are used by control and status signals.

- **ALE** – ALE is an acronym for address latch enable and is pin number 30 in the configuration. We know that 8 lower order bits of the 16-bit address bus are multiplexed with the 8-bit data bus. This pin gets enabled at the time when the address is present at the multiplexed address and data bus. Otherwise, it gets disabled showing the absence of an address on the bus.
- **RD** – This pin is numbered 32 in the configuration and a low signal in this pin shows the read operation either from I/O devices or from the memory unit. Thereby indicating that the data bus is now in a state or position to accept the data from the memory or I/O devices.
- **WR** – It is the 31st pin in the pin diagram and a low signal in this pin represents the write operation at the memory or I/O devices. This indicates that the data present in the data bus is to be written into the desired memory address or I/O device by the processor.
- **IO/M** – It is pin number 34 and indicates the selection of a memory address or input-output device. This shows whether the read/write operation is to be carried out at the memory location or at the I/O device.
  The low signal at this pin shows that operation is performing over memory location. As against, a high signal at this pin represents the operation at I/O device.
- **S_0 and S_1** – The pins S_0 and S_1 represent the status signal at pin number 29 and 33 respectively. These signals show the type of recent operation of the microprocessor. The table below represents the status of the data bus under different conditions:

Machine cycle	Status			Controls		
	IO / $\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch (OF)	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/O Read (I/OR)	1	1	0	0	1	1
I/O Write (I/OW)	1	0	1	1	0	1

Interrupts and externally generated signals:

Interrupts are the signals that are generated to break the sequence of an on-going operation. When an interrupt signal is generated then CPU immediately stops its recent task under operation and switches to some other program known as interrupt service routine (ISR). However, after handling ISR, the CPU gets back to its main program for execution.

In the pin configuration, 5 types of interrupts are shown by 5 different pins from pin number 6 to 10. These pins are used to manage the interrupt. Basically, there exist 2 types of interrupts:

Maskable Interrupt and **Non- maskable interrupt**

Out of the 5 major interrupts 4 are the maskable interrupts. These are INTR, RST5.5, RST6.5, and RST 7.5 and are easily manageable interrupts.

However, TRAP is a non-maskable interrupt and holds the topmost priority among all interrupts in the 8085 microprocessor.

- **RESET IN** – It is pin number 36 in the pin diagram. An active low signal at this pin resets the PC of the microprocessor to 0. Or we can say, after resetting the PC holds its initial memory address.
- **RESET OUT** – It is the 3rd pin in the pin diagram. This pin generates a signal to provide information about the resetting of the microprocessor. Also, we can say that once a processor is reset then all the connected devices must also be reset.
  So, enabling this signal shows the resetting of the interconnected devices.

- **INTA**: It is the 11th pin of the 8085 pin configuration. A signal at this pin acknowledges the generated interrupt.

Direct Memory Access (DMA):

We are aware of the fact that memory and I/O devices are connected with each other by the microprocessor. So, the intermediate i.e., CPU manages the data transfer between the input-output device and memory.

However, when data in a large amount is to be transferred between I/O devices and memory the CPU gets disabled by tri-stating its buses. And this transfer is manageable by external control circuits. The DMA has 2 pins.

- **HOLD** – This signal is generated at pin number 39. This pin generates a signal to notify the processor that more than one request is present to access the data and address bus.
  When this signal gets enabled, the CPU frees the bus after completion of the recent operation. Once the hold signal gets disabled, the processor can access the bus again.

- **HLDA** -This signal is generated at pin number 38. This signal is enabled at the time when the processor gets HOLD signal and it releases HLDA i.e., hold acknowledge signal. In order to show that the multiple requests are kept on hold and will be considered once the bus gets free after the recent operation.
  After the disabling of hold request, the HLDA signal becomes low.

- **READY** -This is the 35th numbered pin in the pin diagram that maintains synchronization between the processor and peripherals, memory. It is clear that a microprocessor has a much faster response than peripherals and memory.
  So, this pin is enabled when the processor as well as the peripherals and memory both become ready to begin the next operation. In the case when the READY pin is disabled, then the microprocessor is in the WAIT state.

1.3.3 DEMULTIPLEXING OF AD0-AD7 USING IC 74LS373

The dual-purpose of the AD0-AD7 pins is achieved through multiplexing. In simple words, multiplexing allows us to use the pins of a microprocessor for more than one function. For extracting the data and the lower 8 bits of the address, we demultiplex AD0-AD7 using IC 74LS373.

Need

It is very important to reduce number of pins in an IC. As we know address in 8085contains 16 bits, and data contains 8 pins, we need 24 pins in IC to be used as address and data pins, rather than this designer has a different approach to reduce the usage of pins for address & amp; data. Only 16 pins are used totally for address and data, in which 8 pins (AD0 -AD7) are combined, used to generate address & data. Microprocessor generates both data & addresses one same 8 pins. The thing is to resolve both address & data from these pins. This process is achieved through demultiplexing the address & data signals

The Address and Data Busses

The address bus has 8 signal lines A8 –A15which are unidirectional. The other 8 address bits are multiplexed (time shared) with the 8 data bits. So, the bits AD0 –AD7 are bi-directional and serve as A0 –A7and D0 –D7at the same time. During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits. In order to separate the address from the data, we can use a latch to save the value before the function of the bits changes.

Demultiplexing AD7-AD0

From the above description, it becomes obvious that the AD7–AD0 lines are serving a dual purpose and that they need to be demultiplexed to get all the information. The high order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period and they would be lost if they are not saved externally. Also, notice that the low order bits of the address disappear when they are needed most. To make sure we have the entire address for the full three clock cycles, we will use an external latch to save the value of AD7–AD0 when it is carrying the address bits. We use the ALE signal to enable this latch.

ALE operates as a pulse during T1, we will be able to latch the address. Then when ALE goes low, the address is saved and the AD7–AD0 lines can be used for their purpose as the bi-directional data lines.

The high order address is placed on the address bus and hold for 3 clk periods, the low order address is lost after the first clk period, this address needs to be holding however we need to use latch. The address AD7 –AD0 is connected as inputs to the latch 74LS373. The ALE signal is connected to the enable (G) pin of the latch and the OC –Output control –of the latch is grounded which you can see in the given fig.14.

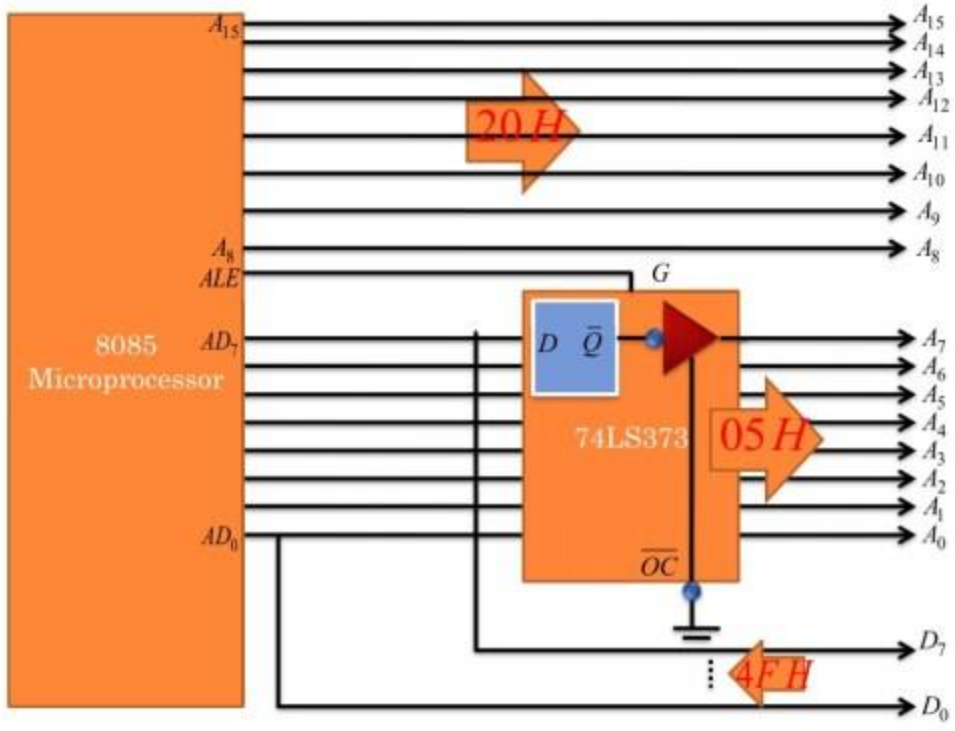


Fig.14 Demultiplexing of AD0-AD7 using IC 74LS373

Procedure

Every Instruction has some machine cycles to complete its executions sometime called T states. Whenever an instruction is executed by MPU first of all MPU sends ALE signal to address latch IC to enable all D Latches to receive new address from MPU now (in first T state)

Microprocessor generates the address on Address Bus, half portion of address (lower order address) is generated on AD0 -AD7. This Address bits are captured by D latches and stored in. Now during next cycles say T2 T3 and so on, MPU can use AD 0 -AD 7 as Data Bus to send receives data. During this period the initially generated Address is also available at output pins of D Latches .The IC 74LS31 is used as Address Latch it contains 8 D Latches to store lower half of address.(8 bits).

1.3.4 GENERATION OF CONTROL SIGNALS

- For dealing with different peripherals using 8085 microprocessor there are two modes of operations.
- One is to **read data from any of the memory device or input device**.
- Second is to **write data on some location, this location can be any of the output device location or any of the memory location**.
- To deal with these modes 8085 microprocessor architecture gives three different control signals. Namely **IO/$\overline{M}$, $\overline{RD}$ and $\overline{WR}$**. The IO/$\overline{M}$ is an output pin of the 8085 microprocessor which serves dual purpose, the high going pulse on this pin indicates the I/O type of operation.

- We can state that, at this time 8085 is working with the input or output devices. The low going pulse on this pin indicates the memory operation. The second one is the $\overline{RD}$ stand for read signal.
- This is active low signal, indicates the memory or I/O type of read operation and the selected memory or I/O device is to be read. And the third is $\overline{WR}$ stands for write signal. This is also active low signal, indicates the memory or I/O type of write operation and data available on the Data bus is to be written in to the selected memory or I/O location, data is set up at the trailing edge of the pin.
- To deal with different I/O as well as memory device individually, we have to generate four individual control signals. This control signals used to select any of the I/O or memory device, with a specific type of operation either of read or write. In our case, we are interested with two operations with memory as well as output devices.
- First of all, we required fetching the instructions place inside the memory and next we write data word on output port. According to our need we have develop the logic to generate control signals. The fig. 15 describes the combination logic of signal generation.

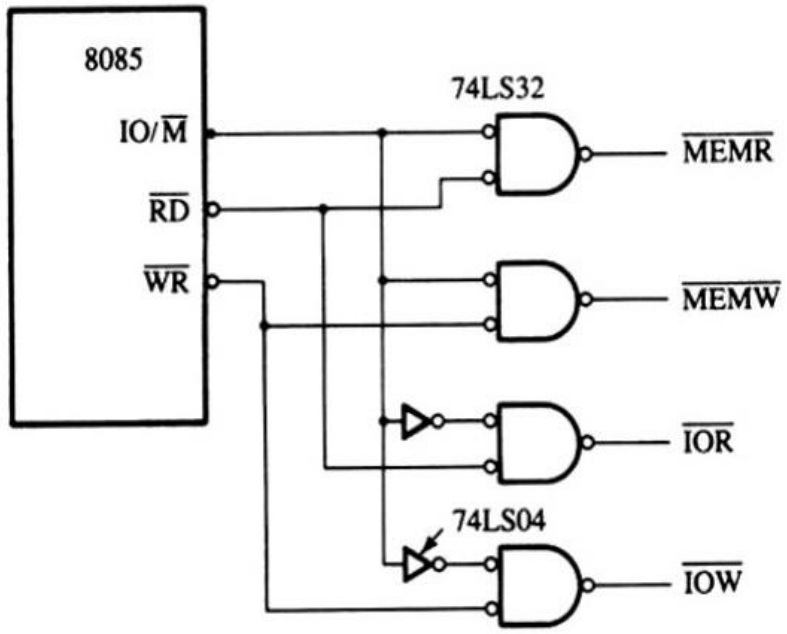


Fig.15 Schematic Diagram to generate control signals

1.3.5 UNIVERSITY QUESTIONS RELATED TO THE TOPIC

Two-mark questions

Q-1. List out the interrupts of 8085.

Q-2. What is the work of Program Counter?

Q-3. What is the work of Stack Pointer?

Q-4. What is the need of demultiplexing the bus AD7-AD0?

Q-5. What is the function of accumulator?

Q-6. What is the use of ALE pin?

Five-mark questions

Q-7. Draw the programming model of 8085 Microprocessor and explain each of its components in brief.

Q-8. Explain the flag register of 8085 in detail.

Q-9. Explain how will you demultiplex the bus AD7-AD0?

Q-10. Which control signals are used in 8085 Microprocessor? Explain them.

Ten-mark questions

Q-11. Draw the internal architecture of 8085 and explain its each block.

Q-12. Draw the pin diagram of 8085 MPU. Explain all the pins of 8085 MPU.

Q-13. How any registers are there in 8085? Explain the use of each register.

GATE Questions

Q-14. In an 8085 microprocessor, the shift registers which store the result of an addition and the overflow bit are, respectively.

Answer: A and F

Q-15. An 8 Kbyte ROM with an active low Chip Select input (CS) CS is to be used in an 8085 microprocessor based system. The ROM should occupy the address range 1000H to 2FFFH. The address lines are designated as *A*15 to *A*0, where *A*15 is the most significant address bit. Which one of the following logic expressions will generate the correct CSCS signal for this ROM?

Answer:

$$A_{15} + A_{14} + \left(A_{13} \cdot A_{12} + \overline{A_{13}} \cdot \overline{A_{12}} \right)$$

1.4 TIMING DIAGRAM OF 8085 MICROPROCESSOR

1.4.1 WHAT IS TIMING DIGRAM

"The Timing Diagram is the graphical representation of the time taken for the execution of each instruction by a microprocessor. The execution time is represented in T-states."

1.4.2 NEED OF TIMING DIGRAM

As the heartbeat is required for the survival of the human being, the CLK is required for the proper operation of different sections of the microprocessors. All actions in the microprocessor

is controlled by either leading or trailing edge of the clock. The 3-status signals : IO / M, S1, and S0 are generated at the beginning of each machine cycle.

As far as execution of instructions is concerned, in 8085 microprocessor, each instruction is divided into two parts:

1- Operation code(opcode)
2- Operand

- The **Opcode** tells us what the operation is and the operand is the necessary information required for the instruction.
- The **Operand** may be either data or an address or other information required for the instruction. Each instruction is divided into machine cycles and each machine cycle is divided into clock cycles or T-state.

1.4.3 WORKING OF TIMING DIGRAM

To know the working of 8085 microprocessor, we should know the timing diagram of 8085 microprocessor. With help of timing diagram, we can easily calculate the execution time of instruction and as well as program. Before going for timing diagram of 8085 microprocessor, we should know some basic parameters to draw timing diagram of 8085 microprocessor. Those parameters are:

- **Instruction Cycle (IC)**
- **T-state (T)**
- **Machine cycle (MC)**

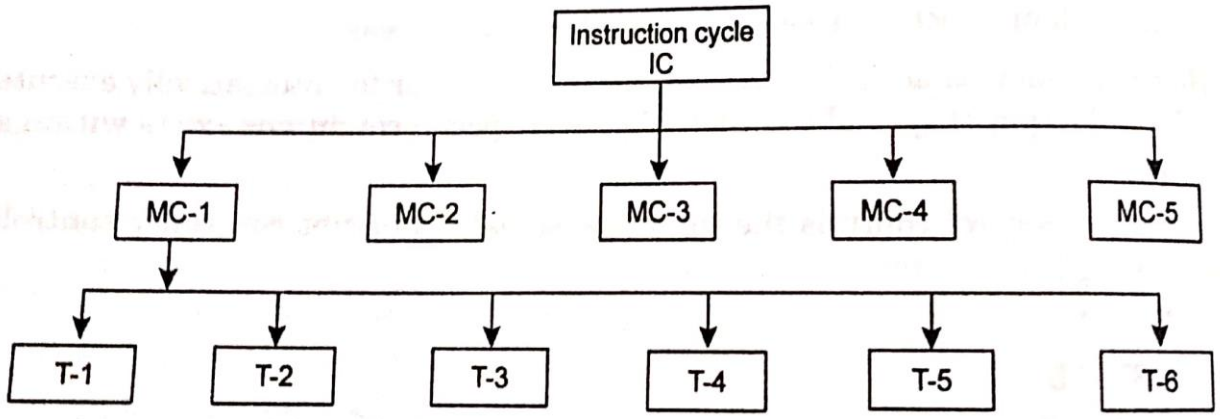


Fig.16 No. of cycles in timing diagram

1.4.3.1 T-STATE

"Processors work in synchronization with a periodic signal called 'clock signal'. The part of any operation carried out during one time period of this clock signal is known as a T state. Sometimes, 'T state' is also used to refer to a time of one clock period. For studying the rest of the concepts, it becomes a basic unit of measuring time."

Clock Signal

- The 8085 divides the clock frequency provided at X1 and X2 inputs by 2 which is called operating frequency.
- One T-state is equal to the time period of the internal clock signal of the microprocessor

Example

If the internal clock frequency of 8085 microprocessor is 3 MHZ, One T-state is equal to

$$T = \frac{1}{f} = \frac{1}{3\ x\ 10^6} =. 333\ X10^{-6} sec = 333\ X\ 10^{-9}\ sec$$

Time required for Execution of Instruction = Total no. of T-states * clock period

1.4.3.2 MACHINE CYCLE

"The time required for the microprocessor to access memory or an IO device either for a read operation or a write operation is called a machine cycle."

There are seven different types of machine cycles in 8085, which are listed below:

Sr. No.	Machine cycle	T-state	IO/M	S1S0	Other control signals
1	Opcode fetch machine cycle (OFMC)	4T or 6T	0	11	$\overline{RD}$ = 0
2	Memory read machine cycle (MRMC)	3T	0	10	$\overline{RD}$ = 0
3	Memory write machine cycle (MWMC)	3T	0	01	$\overline{WR}$ = 0
4	IO read machine cycle (IORMC)	3T	1	10	$\overline{RD}$ = 0
5	IO write machine cycle (IOWMC)	3T	1	01	$\overline{WR}$ = 0
6	Interrupt acknowledge machine cycle	6T or 12T	1	11	INTA = 0
7	Bus Idle machine cycle	NA	0	00	INTA = $\overline{RD}$ = $\overline{WR}$ = 1

1.4.3.3 INSTRUCTION CYCLE

"For the execution of any instructions, basically, two steps are followed – fetch and then execute. The time (or the number of 'T states') required to fetch and execute an instruction is called an instruction cycle."

One instruction cycle will consist of either one or more than one machine cycle.

IC=FC+EC

Where IC= time required for fetch cycle

EC= time required for execution cycle

1.4.4 REPRESENTATION OF SIGNALS IN TIMING DIAGRAM

Clock Signal

One T-state is equal to the time period of the internal clock signal of the microprocessor

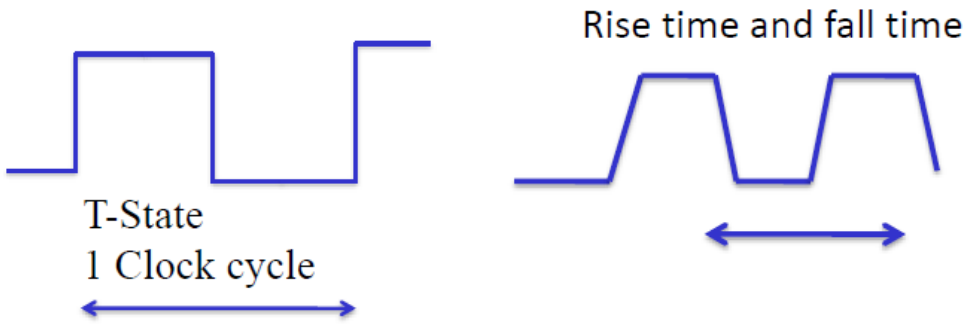


Fig.17 Clock cycles in timing diagram

Single Signal

Single signal status is represented by a line. It may have status either logic 0 or logic 1 or tri-state.

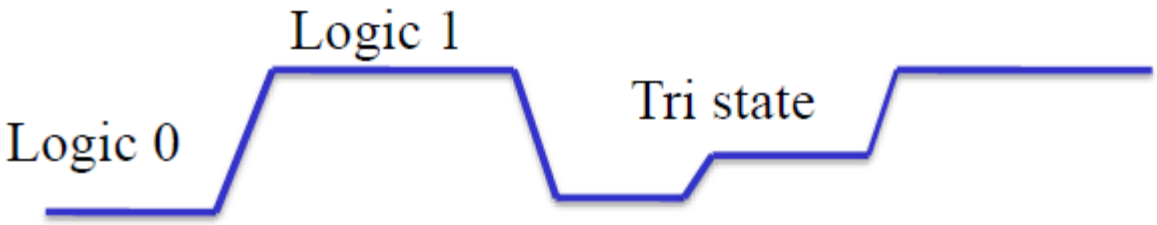


Fig.18 Single signal in timing diagram

Group of signals

Group of signals is also called a bus. Eg: Address bus, data bus

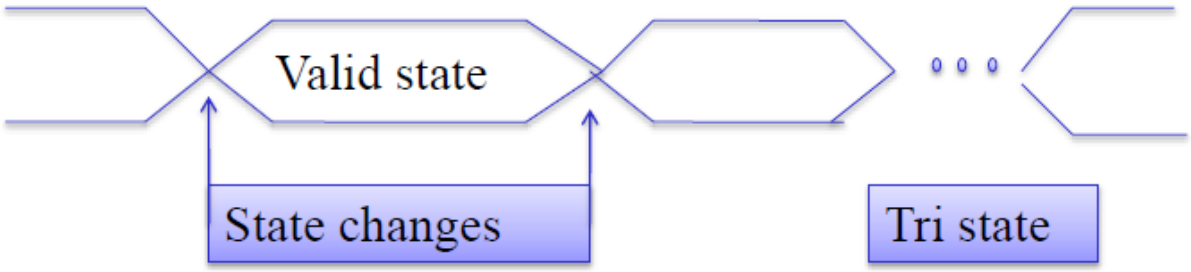


Fig.19 Group of signals in timing diagram

ALE

This signal is active high signal. It is activated in the beginning of T1 state of each machine cycle except bus idle machine cycle and it remains active in the T1.

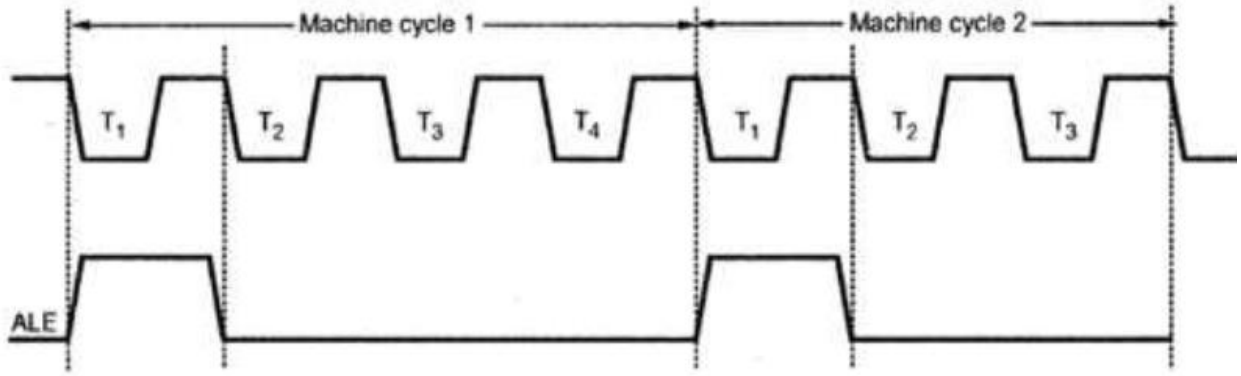


Fig.20 ALE activation and its period

A0 - A7 (Lower Byte Address)

This is available on the multiplexed address/data bus (AD0 - AD7) during T1 state of each machine cycle except bus idle machine cycle.

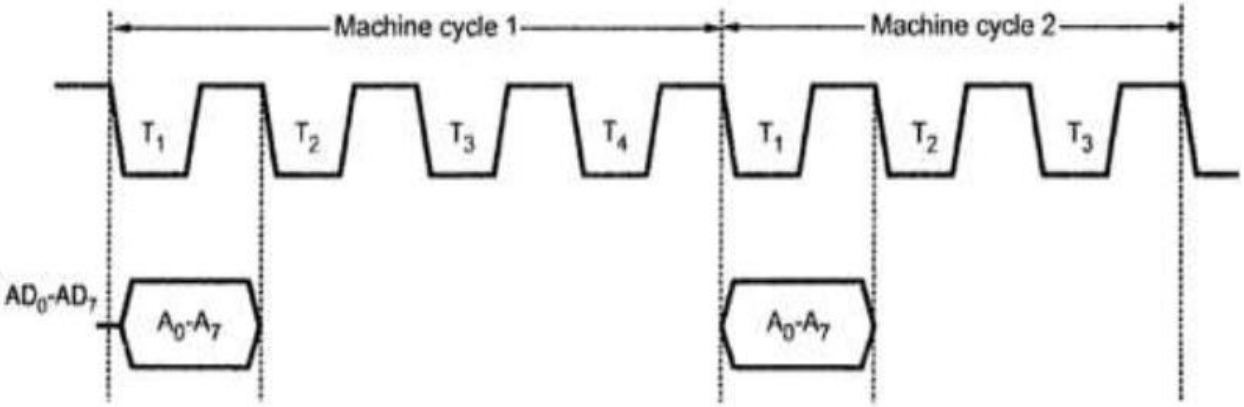


Fig.21 Lower address on the multiplexed bus

D0 - D7 (Data Bus)

The data from memory or I/O device and from microprocessor to memory or I/O device is transferred during T2 and T3 – states.

- In **read machine cycle**, data will appear on the data bus during the later part of the T2 - state as shown in fig.22. This is because to read data from memory or I/O device it is necessary to select memory or I/O device, after the selection, device will put the data from the selected location on the data bus. This action needs finite time. This time is referred as "**Access Time**".
- In case of **write cycle**, data is available in the register set of the microprocessor and it can put that data on the data bus with zero access time.

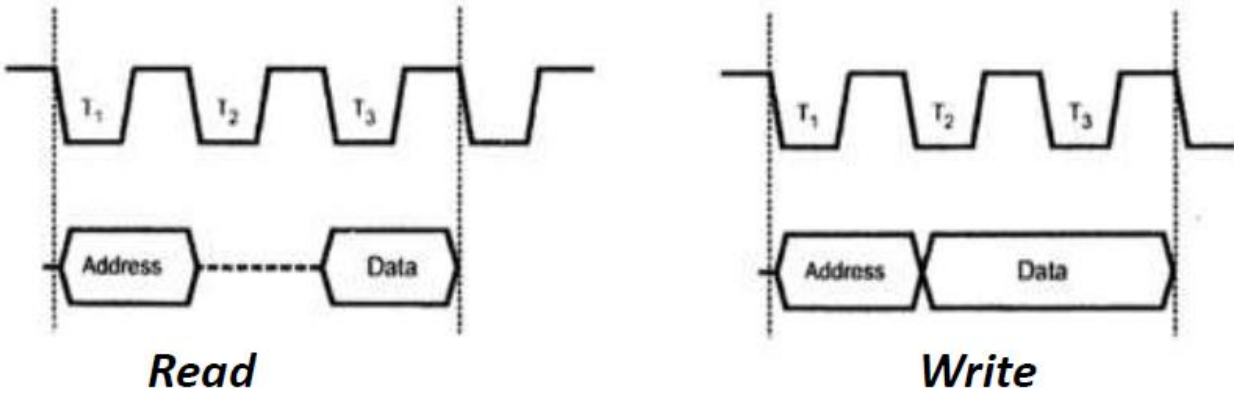


Fig.22 Data bus

A8 - A15 (Higher Byte Address)

The higher byte of address is available on the A8 – A15 bus during T1, T2, and T3 – state of each machine cycle, except bus idle machine cycle

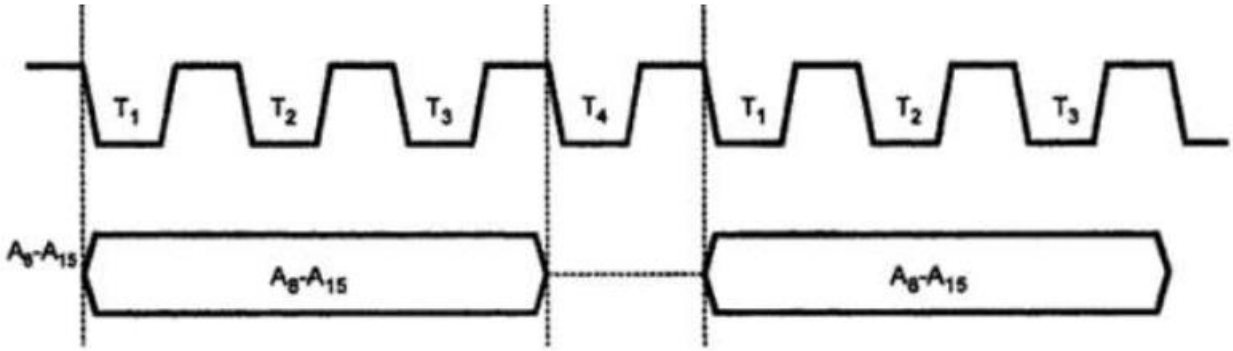


Fig.23 Higher byte address

IO/$\overline{M}$, S0, S1

These are called status signals. They decide the type of machine cycle to be executed. They are activated at the beginning of T1 – state of each machine cycle and remain active till the end of the machine cycle.

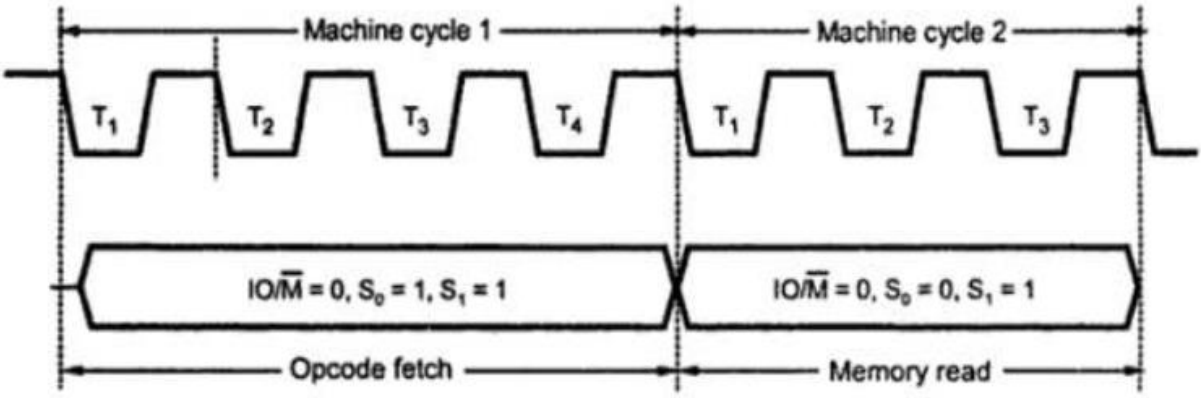


Fig.24 Status signals

$\overline{RD}$ & $\overline{WR}$

- These decide the direction of the data transfer.
- When $\overline{RD}$ signal is active, data is transmitted from memory or I/O device to microprocessor.
- When $\overline{WR}$ signal is active, data is transmitted from microprocessor to the memory or I/O device. Both signals are never active at a time.
- Since the data transfer in 8085 takes place during T2 and T3, these signals are activated during T2 and T3 as shown in the fig.25.

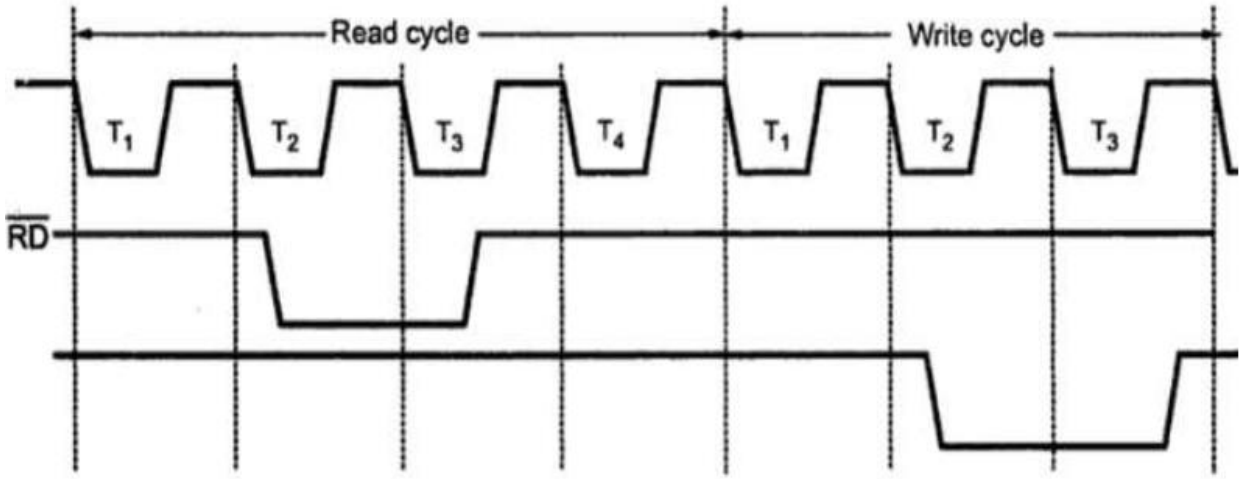


Fig.25 $\overline{RD}$ and $\overline{WR}$ signals

1.4.5 TYPES OF MACHINE CYCLE

There are five machine cycles in 8085 Microprocessor:

1. OPCODE FETCH MACHINE CYCLE (OFMC)
2. MEMORY READ MACHINE CYCLE (MRMC)
3. MEMORY WRITE MACHINE CYCLE (MWMC)
4. IO READ MACHINE CYCLE (IORMC)
5. IO WRITE MACHINE CYCLE (IOWMC)

1.4.5.1 OPCODE FETCH MACHINE CYCLE (OFMC)

The opcode fetch machine cycle (OFMC) involves the fetching of the opcode of the instruction to be executed and the decoding process of that opcode. Usually, it consists of four T states. The timing diagram of a typical OFMC is shown in fig.26 and explained below.

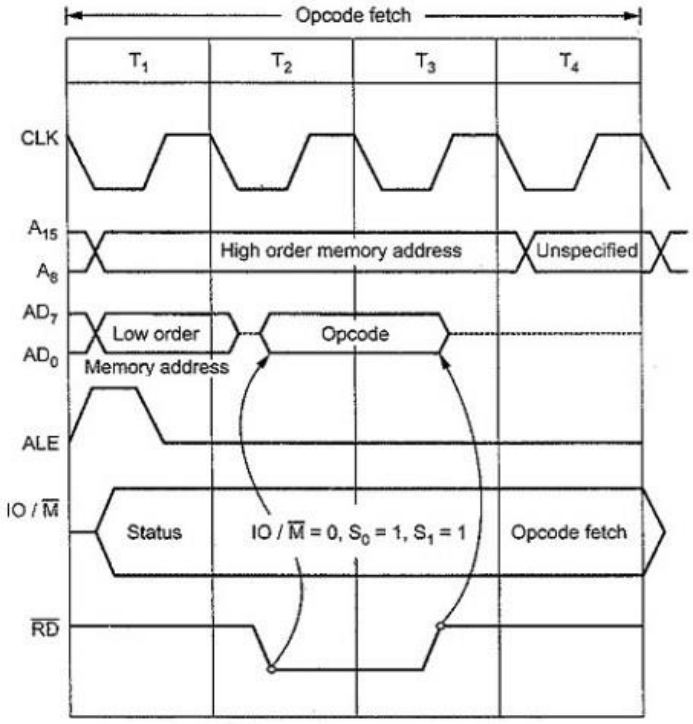


Fig.26 Opcode fetch machine cycle

1st T state

- During the first T state, the address of the location where the opcode is stored is loaded on the address bus. In 8085, this address is stored in a 16-bit register called the program counter. Higher eight bits of the address are loaded on A8-A15, and the lower eight bits of the address are loaded into AD0-AD7 for demultiplexing.
- Also, the ALE signal becomes active in the first T state to indicate that the data on AD0-AD7 pins are the lower address bits.
- IO/M signal becomes low at the beginning of the first T state to indicate that the opcode will be fetched from memory (reading from memory).
- At the beginning of the first T state, signals S1 and S0 take the value 1 and 1 respectively to indicate that it is an opcode fetch machine cycle.

2nd T state

- By the beginning of the 2nd T state or the end of 1st T state, the ALE signal goes low. By this time, 8085 expects that the lower address bits are latched, and AD0-AD7 is free to be used as a data bus.
- At the beginning of the second T state, RD goes low, indicating that the read process has started. Meanwhile, higher address bits are present in A8-A15, and lower address bits are expected to be latched.
- As RD goes low, the opcode (eight bits) is loaded into the data bus AD0-AD7.

3rd T state

- The opcode loaded on the data bus is present there until the middle of the third T state.
- During the third T state, RD goes up, indicating that the read operation is completed and 'the opcode is fetched' and placed in the instruction register.
- The data on the data bus and the higher address bits on A8-A15 exist until the middle of this T state.

4th T state

- During the fourth T state, the fetched opcode is decoded. There is nothing much to observe in the timing diagram during this process.
- In case of some simple one-byte instructions like STC (set carry flag), execution is also completed during the fourth T state. One such instruction is MOV A, D.
- During the fourth T state, after decoding the opcode, the microprocessor decides if it needs fifth and sixth T states, or should proceed to the next machine cycle.
- PC is incremented by 1 here or in the sixth T state if the OFMC is extended upto sixth T state.

5th and 6th T state

- In case of one-byte instructions that operate on 16-bit data and some other instructions, OFMC may extend up to six T states. During the fifth and sixth T states, execution of these instructions takes place. Since these instructions are simple, they get executed in the OFMC itself. Examples of such instructions are DCX, INX, PCHL, SPHL, CALL, RSTN and conditional RET.

1.4.5.2 MEMORY READ MACHINE CYCLE (MRMC)

Contents from a memory location are read during the memory read machine cycle (MRMC). This machine cycle spans over three T states. Each of these T states is explained here along with a timing diagram. The first three T states are almost the same as the first three T states of Opcode Fetch Machine Cycle. The timing diagram of a MRMC is shown in fig.27 and explained below.

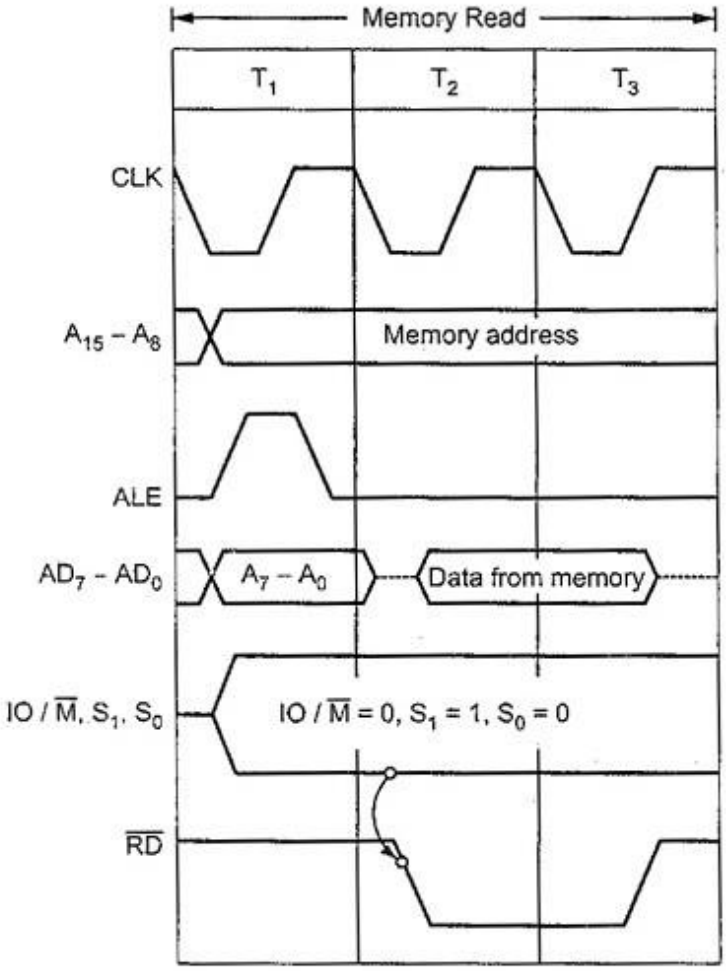


Fig.27 Memory Read machine cycle

1st T state

- Higher address bits loaded into A8-A15.
- Lower address bits loaded into AD0-AD7.
- ALE signal goes high in the beginning to indicate that AD0-AD7 contains lower address bits.
- IO/M goes low since it is a memory operation.
- S1 and S0 become 1 and 0 respectively, indicating Memory Read Machine Cycle.
- ALE goes low by the end of the first T state. Lower address bits are expected to be latch by this time.

2nd and 3rd T states

- RD goes low, indicating the initiation of the read operation.
- Data is read from the memory location and is loaded into the data bus AD0-AD7. The data is loaded into the data bus at the beginning of the 2nd T state and exists until the end of the third T state.
- By the end of the third T state, RD goes high, indicating the end of the read operation.
- PC is incremented by 1 (only in cases described in the note in the red box above).

1.4.5.3 MEMORY WRITE MACHINE CYCLE (MWMC)

Contents are written to a memory location/stack during a memory write machine cycle (MWMC). This machine cycle spans over three T states. Each of these T states is explained here along with the timing diagram. PC is not incremented in this machine cycle. This is very similar to MRMC, except a few differences. The timing diagram of a MWMC is shown in fig.28 and explained below.

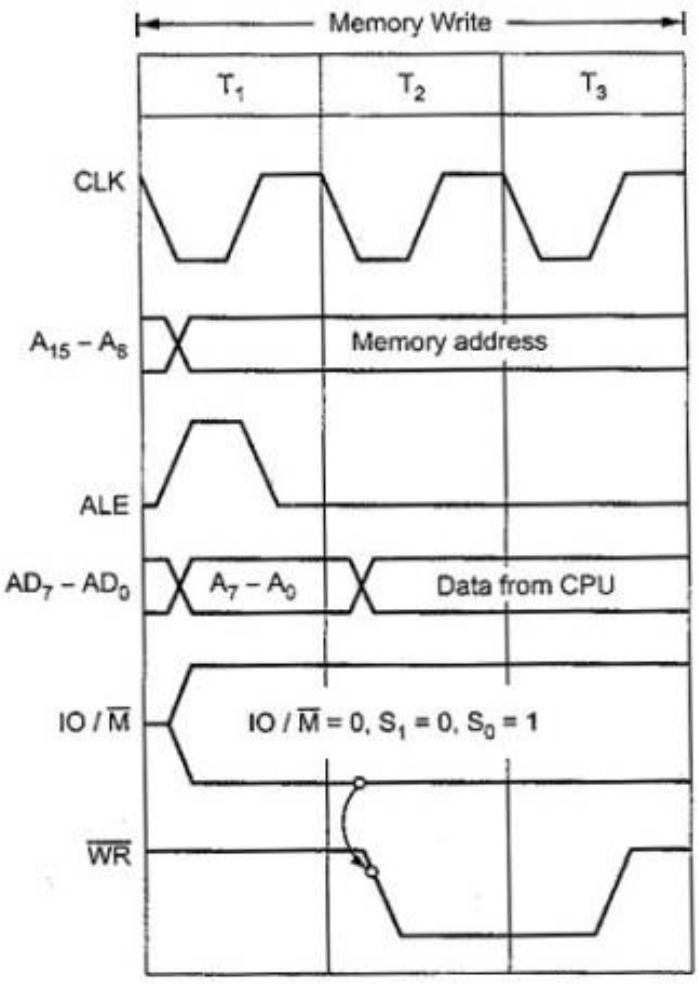


Fig.28 Memory write machine cycle

1st T state

- Higher address bits loaded into A8-A15.
- Lower address bits loaded into AD0-AD7.
- ALE signal goes high in the beginning to indicate that AD0-AD7 contains lower address bits.
- IO/M goes low since it is a memory operation.
- S1 and S0 become 0 and 1 respectively, indicating MWMC.
- ALE goes low by the end of the first T state. Lower address bits are expected to be latch by this time.

2nd and 3rd T states

- WR goes low, indicating the initiation of the write operation.
- Data to be written is loaded on the data bus at the beginning of the second T state and exists until the end of the third T state when the data is transferred from the data bus to the memory location.
- By the end of the third T state, WR goes high, indicating the end of the write operation. Thus, MWMC comes to an end.

1.4.5.4 IO READ MACHINE CYCLE (IORMC)

Contents from an IO device are read during IO read machine cycle (IORMC). This machine cycle spans three T states and is similar to MRMC except for the IO/M signal. The destination of this read operation is the accumulator. The Program Counter is not incremented here. IO/M goes high instead of going low, indicating that the microprocessor is talking to an IO device. Each of these T states is explained here along with a timing diagram. The timing diagram of an IORMC is shown in fig.29 and explained below.

1st T state

- 8-bit address is loaded into A8-A15.
- The same 8-bit address is loaded into AD0-AD7.
- ALE signal goes high in the beginning to indicate that AD0-AD7 contains address bits and not data bits.
- IO/M goes high since the microprocessor is dealing with an IO device.
- S1 and S0 become 1 and 0 respectively, indicating a ‘read’ machine cycle.
- ALE goes low by the end of the first T state. Address bits in AD0-AD7 are expected to be latch by this time.

2nd and 3rd T states

- RD goes low, indicating the initiation of the read operation.
- Data is read and is loaded on the data bus (AD0-D7) at the beginning of the second T state and exists until the end of the third T state.
- In the third T state, the data is transferred from the data bus to the accumulator.
- By the end of the third T state, RD goes high, indicating the end of the read operation.

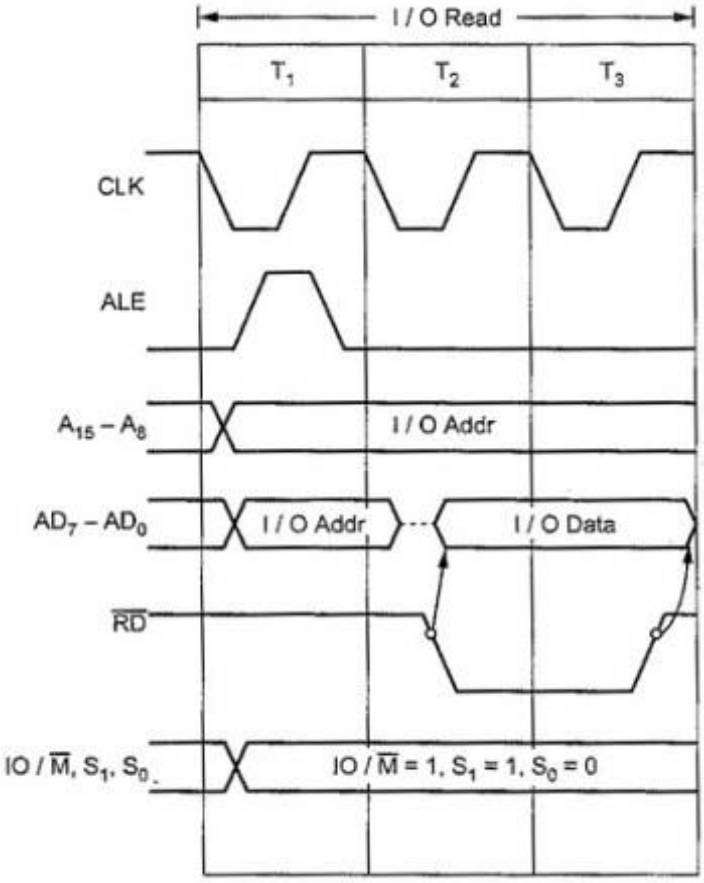


Fig.29 IO Read Machine Cycle

1.4.5.5 IO WRITE MACHINE CYCLE (IOWMC)

Contents are written to an IO device during IO write machine cycle (IOWMC). This machine cycle spans three T states and is similar to MWMC except for the IO/M signal. IO/M goes high instead of going low, indicating that the microprocessor is talking to an IO device. The contents of the accumulator are transferred to the data bus and written to an output device in this cycle. The T states are explained here along with a timing diagram for your reference. The timing diagram of an IOWMC is shown in fig.30 and explained below.

1st T state

- 8-bit address is loaded into A8-A15.
- The same 8-bit address is loaded into AD0-AD7.
- ALE signal goes high in the beginning to indicate that AD0-AD7 contains address bits.
- IO/M goes high since the microprocessor is dealing with an IO device.
- S1 and S0 become 0 and 1 respectively, indicating a write machine cycle.
- ALE goes low by the end of the first T state. Address bits in AD0-AD7 are expected to be latch by this time.

2nd and 3rd T states

- WR goes low, indicating the initiation of the write operation.
- Data to be written is loaded on the data bus at the beginning of the second T state and exists until the end of the third T state.
- In the third T state, the data is transferred from the data bus to the IO device.
- By the end of the third T state, WR goes high, indicating the end of the write operation.

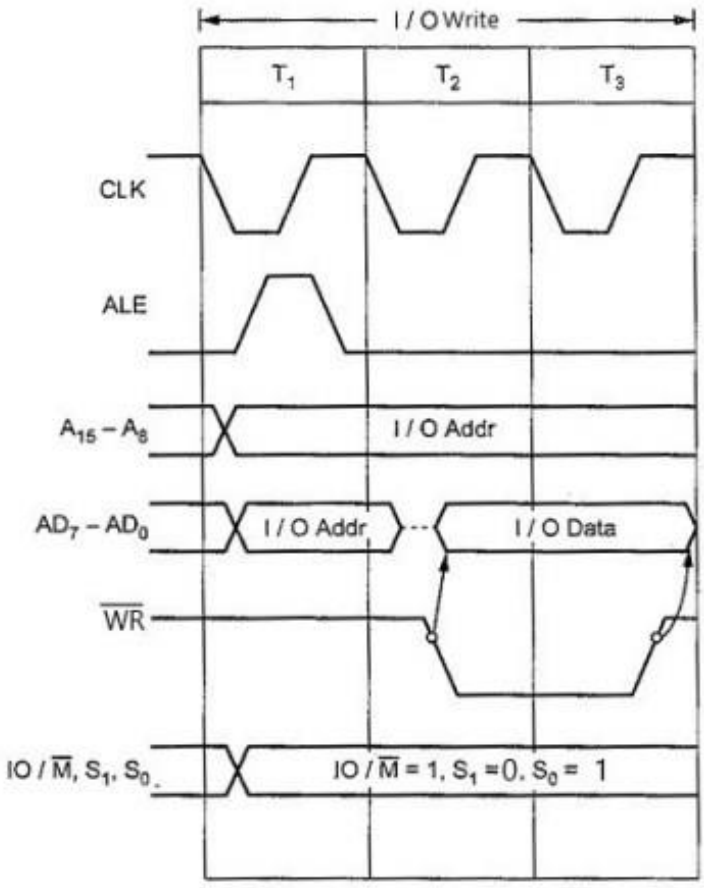


Fig.30 IO Write Machine Cycle

1.4.6 EXAMPLES OF TIMING DIAGRAM

Example-1 Timing diagram for STA 526AH

- STA means Store Accumulator -The contents of the accumulator are stored in the specified address (526A).
- The opcode of the STA instruction is said to be 32H. It is fetched from the memory 41FFH (see fig. 22). - *OF machine cycle*
- Then the lower order memory address is read (6A). - *Memory Read Machine Cycle*
- Read the higher order memory address (52).- *Memory Read Machine Cycle*
- The combinations of both the addresses are considered and the content from accumulator is written in 526A. - *Memory Write Machine Cycle*
- Assume the memory address for the instruction and let the content of accumulator is C7H. So, C7H from accumulator is now stored in 526A.

Address	Mnemonics	Opcode
41FF	STA 526A$_H$	32$_H$
4200		6A$_H$
4201		52$_H$

- **It require 4 m/c cycles & 13 T states**

1. **Opcode fetch(4T)**
2. **Memory read(3T)**
3. **Memory read(3T)**
4. **Memory write(3T)**

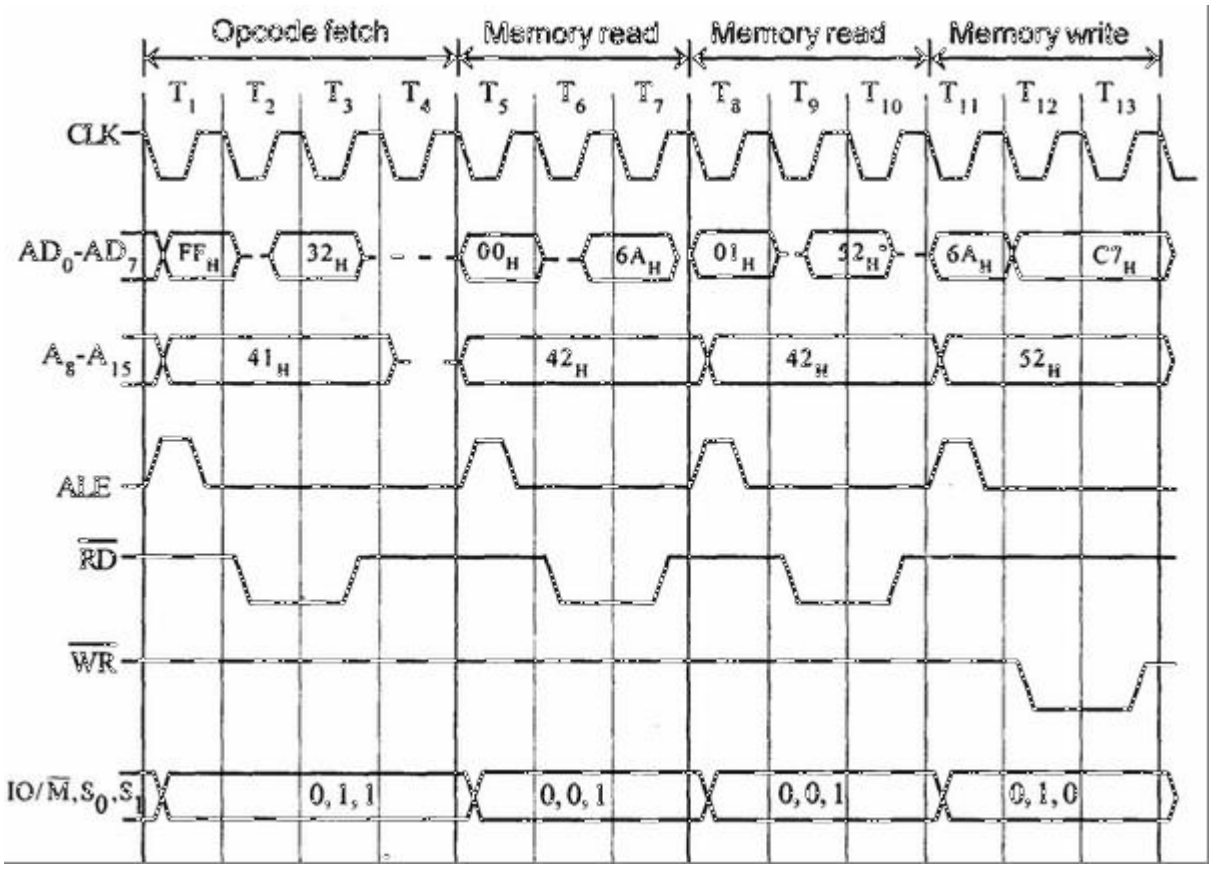


Fig.31 Timing diagram for STA 526AH

Example-2 Timing diagram for IN C0H

- Fetching the Opcode DBH from the memory 4125H.
- Read the port address C0H from 4126H.
- Read the content of port C0H and send it to the accumulator.
- Let the content of port is 5EH.

Address	Mnemonics	Opcode
4125	IN $C0_H$	DB_H
4126		$C0_H$

➢ **It require 3 m/c cycles & 10 T states**

1. **Opcode fetch(4T)**
2. **Memory read(3T)**
3. **I/O read(3T)**

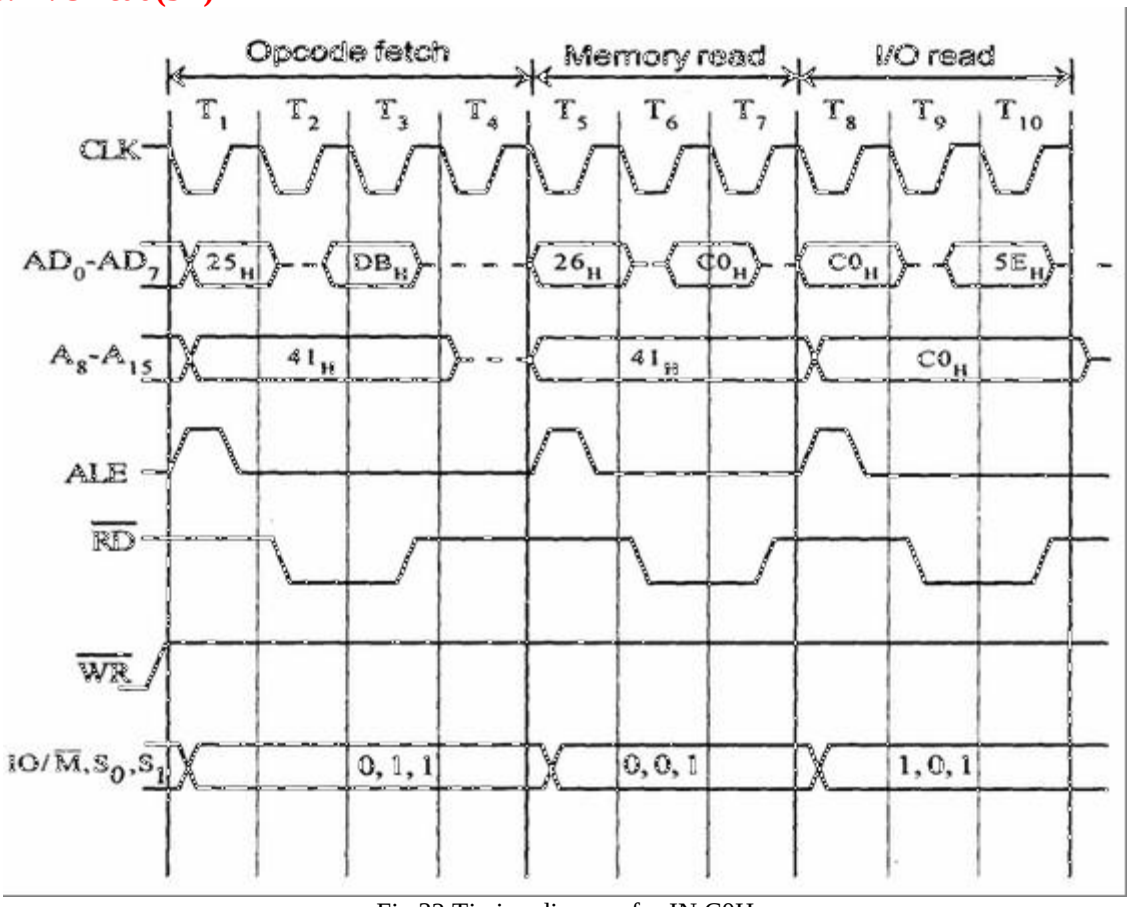


Fig.32 Timing diagram for IN C0H

1.4.7 UNIVERSITY QUESTIONS RELATED TO THE TOPIC

Two-mark Questions

Q-1. If the clock frequency is 5 MHZ, How much time is required to execute an instruction of 18-T states?

Five-mark Questions

Q-2. How Instruction Cycle, Machine Cycle and Clock cycles are related? Explain them with Sketches?

Q-3. Draw the timing diagram for MVI A, B

Ten-mark Questions

Q-4. Draw the neat timing Diagram of instruction-LDA 2050H

GATE Questions

Q-5. The clock frequency of an 8085 microprocessor is 5 MHz. If the time required to execute an instruction is 1.4 μs, then the number of T-states needed for executing the instruction is

Answer: 7T states needed

1.5 LOGIC DEVICES FOR INTERFACING

1.5.1 What is Logic Devices?

- Several types of interfacing devices are necessary to interconnect the components of a bus-oriented system.
- The devices used in today's microcomputer systems are designed using medium scale integration (MSI) technology.
- Tri-state devices are essential to proper functioning of the bus-oriented system in which the dame nus lines are shared by several components.

1.5.2 Types of Logic Devices

There are several types of interfacing devices used :

- Tri-State Devices
- Buffer
- Decoder
- Encoder
- Latches (Registers)

1.5.2.1 Tri-State Devices

- Tri-state logic devices have three states:
  - ✓ Logic 1 or Low
  - ✓ Logic 0 or High
  - ✓ High impedance
- A tri-state logic device has a extra input line called Enable.
- When this line is active (Enabled), a tri-state device functions in the same way as ordinary logic devices.
- When this line is not active(disabled), the logic device goes into a high impedance state, as if it is disconnected from the system and practically no current is drawn from the system.
- In microcomputer systems, peripherals are connected in parallel between the address bus and the data bus.
- Because if the tri-state interfacing devices, peripherals do not load the system buses.
- The microprocessor communicates with one device at a time by enabling the tri-state line of the interfacing device. Tristate Devices shown in fig.33.

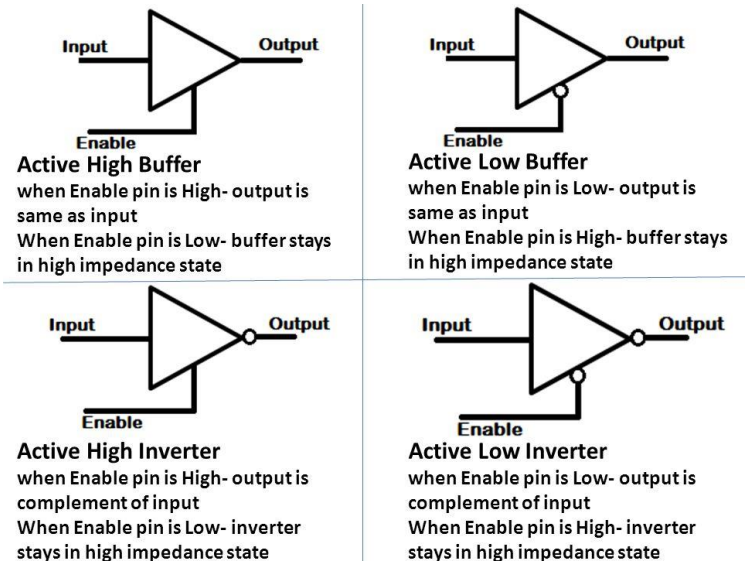


Fig. 33 States of Tri-State Device

1.5.2.2 Buffer

- A Digital Buffer is a single input device that does not invert or perform any type of logical operation on its input signal.
- In other words, the logic level of the output is same as that of the input.
- The buffer is a logic circuit that amplifies the current or power.
- The buffer is used primarily to increase the driving capability of a logic circuit.
- It is also known as driver.

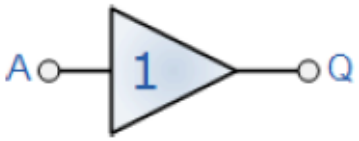
Symbol	Truth Table	
 **A Tri-state Buffer**	A	Q
	0	0
	1	1
Boolean Expression **Q = A**	Read as **A** gives **Q**	

Fig. 34 Tristate-Buffer

Tri-state Buffer

A Tri-state Buffer can be thought of as an input controlled switch which has an output that can be electronically turned "ON" or "OFF" by means of an external "Enable" signal input. This Enable signal can be either a logic "0" or a logic "1" type signal. When Enable line is low (logic „0“), the circuit functions as a buffer. When Enable line is high (logic „1“), its output produces an open circuit condition that is neither "High" nor "low", but instead gives an output state of very high impedance, high-Z, or more commonly Hi-Z. Then this type of device has two logic state inputs, "0" or a "1" but can produce three different output states, "0", "1" or "Hi-Z" which is why it is called a "3-state" device. There are two different types of Tri-state Buffer, one whose output is controlled by an "Active-HIGH" Enable signal and the other which is controlled by an "Active-LOW" Enable signal.

Examples

i. The fig. 35 shows two groups of four buffers with non-inverted tri-state output. The buffers are controlled by two active low Enable lines ($\overline{1G}$ *and* $\overline{2G}$). Until these lines are enabled, the output of the drivers remains in the high impedance state.

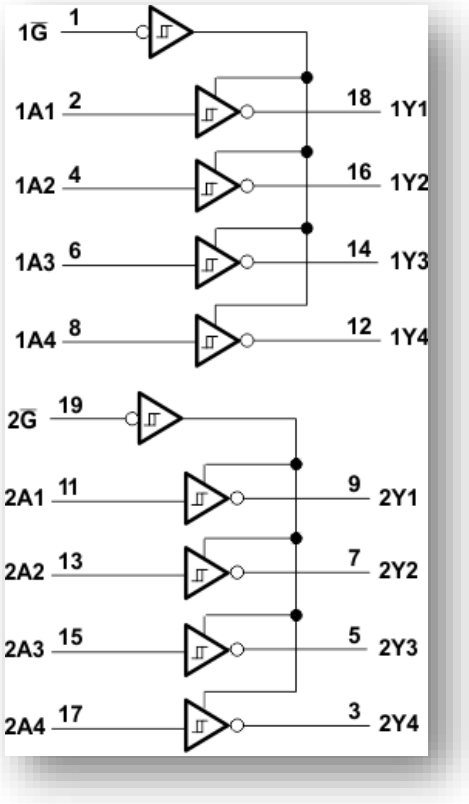


Fig. 35 groups of four buffers

ii. The fig. 36 shows the logic diagram of the bidirectional buffer 74LS245, also called an octal bus transceiver. It includes 16 bus drivers, eight for each direction, with tri-state output.

Enable $\overline{G}$	Direction control DIR	Operation
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = High level voltage
L = Low level voltage
X = Irrelevant

74HC245

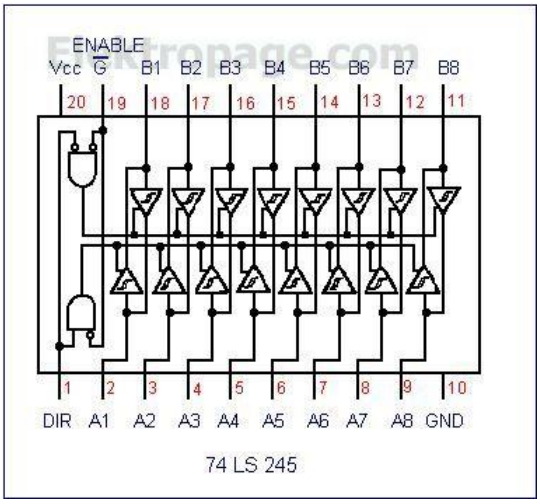


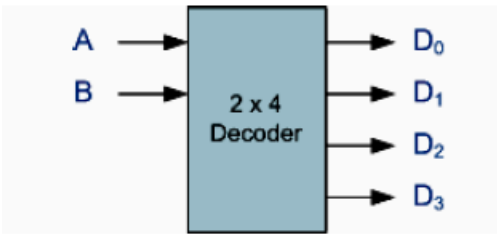
Fig.36 Logic diagram & function table of 74LS245

1.5.2.3 Decoder

- It is a combinational logic circuit that converts the binary code data at its input into an equivalent decimal code at its output.
- Binary Decoders have inputs of 2-bit, 3-bit or 4-bit codes depending upon the number of data input lines, and a n-bit decoder has 2^n output lines. Therefore, if it receives n inputs (usually grouped as a binary or Boolean number) it activates one and only one of its 2^n outputs based on that input with all other outputs deactivated.
- A decoder's output code normally has more bits than its input code and practical binary decoder circuits include 2-to-4, 3-to-8 and 4-to-16 line configurations.
- A binary decoder converts coded inputs into coded outputs, where the input and output codes are different, and decoders are available to "decode" either a Binary or BCD (8421 code) input pattern to typically a Decimal output code.

Examples:

i. **2-to-4 line decoder**

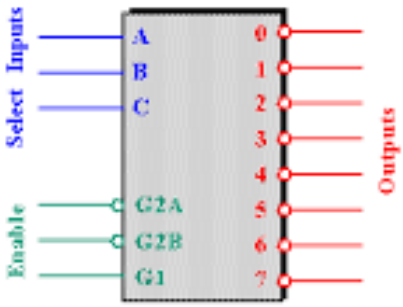


A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Fig.37 Logic diagram and function table of 2 to 4 line Decoder

- ✓ In this simple example of a 2-to-4 line binary decoder, the binary inputs A and B determine which output line from D0 to D3 is "HIGH" at logic level "1" while the remaining outputs are held "LOW" at logic "0" so only one output can be active (HIGH) at any one time.
- ✓ Therefore, whichever output line is "HIGH" identifies the binary code present at the input, in other words it "de-codes" the binary input and these types of binary decoders are commonly used as Address Decoders in microprocessor memory applications.

ii. **3 to 8 line decoder**



Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Fig.38 Logic diagram and function table of 3-to-8-line decoder

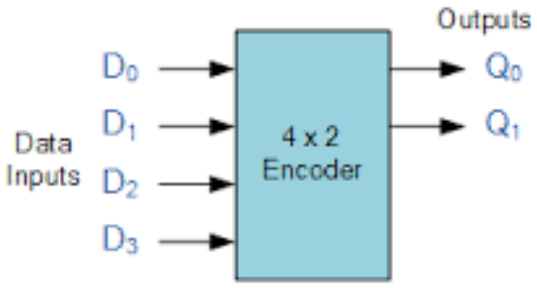
Some binary decoders have an additional input labelled "Enable" that controls the outputs from the device. This allows the decoders outputs to be turned "ON" or "OFF".

1.5.2.4 Encoder

- The encoder is a logic circuit that provides the appropriate code (binary, BCD, etc.) as output for each input signal.
- A binary encoder is a multi-input combinational logic circuit that converts the logic level "1" data at its inputs into an equivalent binary code at its output.
- Generally, digital encoders produce outputs of 2-bit, 3-bit or 4-bit codes depending upon the number of data input lines.
- An "n-bit" binary encoder has 2^n input lines and n-bit output lines with common types that include 4-to-2, 8-to-3 and 16-to-4 line configurations.
- The output lines of a digital encoder generate the binary equivalent of the input line whose value is equal to "1" and are available to encode either a decimal or hexadecimal input pattern to typically a binary or BCD output code.

Examples:

i. 4-to-2-line Encoder



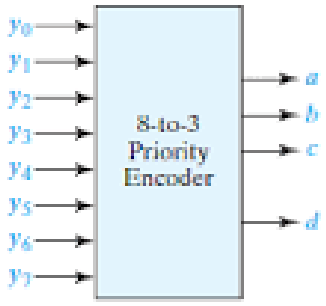
Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

Fig.39 Logic diagram and function table of 4-to-2-line encoder

- ✓ One of the main disadvantages of standard digital encoders is that they can generate the wrong output code when there is more than one input present at logic level "1".
- ✓ For example, if we make inputs D1 and D2 HIGH at logic "1" at the same time, the resulting output is neither at “01” nor at "10" but will be at "11" which is an output binary number that is different to the actual input present.
- ✓ Also, an output code of all logic "0"s can be generated when all of its inputs are at "0" or when input D0 is equal to one.
- ✓ One simple way to overcome this problem is to "Prioritise" the level of each input pin and if there was more than one input at logic level "1" the actual output code would only correspond to the input with the highest designated priority.
- ✓ Then this type of digital encoder is known commonly as a Priority Encoder or P-encoder for short.

ii. 8 to 3 Priority Encoder

The Priority Encoder solves the problems mentioned above by allocating a priority level to each input. The priority encoders output corresponds to the currently active input which has the highest priority. So, when an input with a higher priority is present, all other inputs with a lower priority will be ignored.



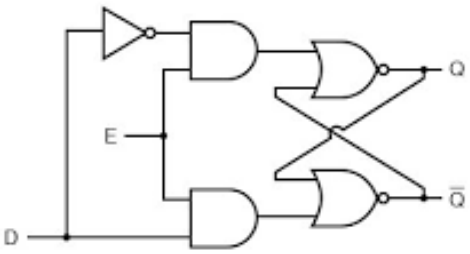
y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	a	b	c	d
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
X	1	0	0	0	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0	1
X	X	X	1	0	0	0	0	0	1	1	1
X	X	X	X	1	0	0	0	1	0	0	1
X	X	X	X	X	1	0	0	1	0	1	1
X	X	X	X	X	X	1	0	1	1	0	1
X	X	X	X	X	X	X	1	1	1	1	1

Fig.40 Logic diagram and function table of 8-to-3 priority encoder

- ✓ In an 8-to-3bit priority encoder which has eight active LOW (logic "0") inputs and provides a 3-bit code of the highest ranked input at its output.
- ✓ Priority encoders output the highest order input first for example, if input lines "D2", "D3" and "D5" are applied simultaneously the output code would be for input "D5" ("101") as this has the highest order out of the 3 inputs.
- ✓ Once input "D5" had been removed the next highest output code would be for input "D3" ("011"), and so on.

1.5.2.5 Latches (Registers)

- ✓ The D flip-flop is the most important of the clocked flip-flops as it ensures that inputs S and R are never equal to one at the same time.
- ✓ D-type flip-flops are constructed from a gated SR flipflop with an inverter added between the S and the R inputs to allow for a single D (data) input.
- ✓ This single data input D is used in place of the "set" signal, and the inverter is used to generate the complementary "reset" input thereby making a level-sensitive D type flip-flop from a level-sensitive RS-latch as now S = D and R = not D.



E	D	Q	$\overline{Q}$
0	0	latch	latch
0	1	latch	latch
1	0	0	1
1	1	1	0

Fig.41 D -Latch

1.5.3 University Questions related to the topic

Two-mark questions

Q-1. What is the use of bidirectional buffer?
Q-2. What do you mean by priority encoder?
Q-3. What do you mean by tri-state devices?
Q-4. With the help of a neat symbol explain tri-state buffer.
Q-5.

Five-mark questions

Q-6. What are tri-state devices and why are they essential in a bus oriented system?
Q-7. With the help of neat block diagram explain 4 to 2 encoder.
Q-8. With the help of neat block diagram explain 8 to 3 encoder.
Q-9. With the help of neat block diagram explain 3 to 8 decoder.
Q-10. Explain the working of D flip-flop.

Ten-mark questions

Q-11. Explain different types of encoder.
Q-12. Explain decoder in detail.

1.6 MEMORY INTERFACING

1.6.1 WHAT IS MEMORY INTERFACING

- Memory is used for storing information in the form of binary bit – zero (0) and one (1).
- Logic 0 and Logic 1 is stored as a voltage signal 0V and 5V.
- Memory system can store large number of digital information (data).
- Digital system can store the data for both long and short period.
- To store the information in terms of voltage-either logic 1 or logic 0, we use a *cell*.
- One cell can store maximum one bit of data at a time.
- The combination of a group of cells is called as *memory.*

1.6.2 NEED OF MEMORY INTERFACING

The 8085 reads and writes data from or to memory. But we know that unlike microcontrollers that have a certain amount of built-in memory, microprocessors do not have their own memory except for a few registers. So, where does 8085 read and write data?

Since 8085 does not have any substantial internal memory, we need to attach external memory chips to read and write data.

1.6.3 BASIC TERMS USED FOR MEMORY

- **Cell:** A device which is used to store one bit (logic 0 or logic 1) data at a time is called as one bit cell or one bit memory cell.
- **Register:** Register is used for storing any data in the form of binary bits. For storing binary bit, D type flip-flop is used. So, 8-bit register will consist of 8 flip-flops which can store 8 bits of data.
- **Capacity of memory:** This means how many bits can be stored in a particular memory.

  For example:

  i. 1024 x 8 memory – 1024 are the number of words & 8 are the number of bits per word.
  ii. 2048 x 8 memory - 2048 are the number of words & 8 are the number of bits per word.
  iii. 65536 x 8 memory - 65536 are the number of words & 8 are the number of bits per word.

- **Address Bus:** To transfer one bit of data, we require one wire. So, a group of address lines is called address bus.
  For example: 16 address lines means 16 wires or 16 conducting paths are required to transfer the address and 16 bits of data.
- **Read Operation:** The binary data (logic 1 or logic 0) can be retrieved from the specific memory locations.
- **Write Operation:** Write operation means storing operation. A data (logic 1 or logic 0) is placed in a specified memory location. Memory location is selected by address lines.
- **Tri-State Buffers:** An important circuit element that is used extensively in memory. This buffer is a logic circuit that has three states: Logic 0, logic1, and high impedance. When

this circuit is in high impedance mode it looks as if it is disconnected from the output completely. This circuit has two inputs and one output. The first input behaves like the normal input for the circuit. The second input is an "enable". If it is set high, the output follows the proper circuit behavior. If it is set low, the output looks like a wire connected to nothing.

- **Input /Output Devices:** There are two ways to interface 8085 with I/O devices in parallel data transfer mode: Memory Mapped IO and IO mapped IO.

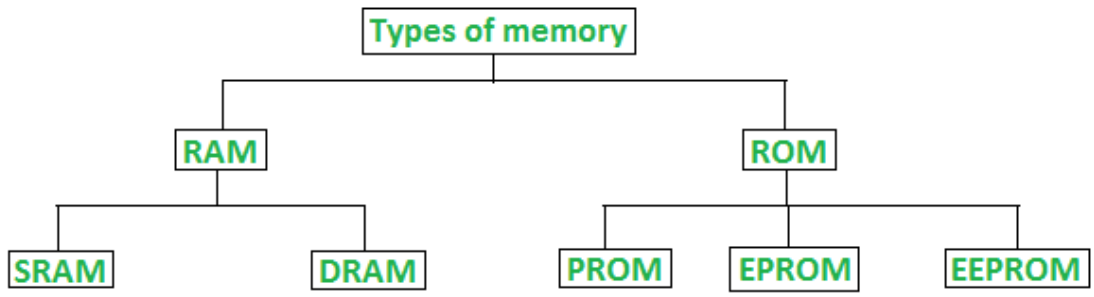
  i. *Memory mapped I/O:* It considers them like any other memory location. They are assigned a 16-bit address within the address range of the 8085.The exchange of data with these devices follows the transfer of data with memory. The user uses the same instructions used for memory.

  ii. *I/O mapped I/O:* It treats them separately from memory: I/O devices are assigned a "port number" within the 8-bit address range of 00H to FFH. The user in this case would access these devices using the IN and OUT instructions only.

1.6.4 CLASSIFICATION OF MEMORY

There are two basic kinds of memory used in microprocessor systems:

i. Random Access Memory (RAM)
ii. Read Only Memory (ROM)



RAM

- This is also known as volatile memory.
- Here, the data is not stored permanently i.e. when power supply of memory is made off, then the stored data is lost.
- **Static RAM** (SRAM) is composed of D-Type Flip-Flops, and is extremely fast, however it is also expensive. It is therefore usually reserved for applications requiring a high speed (such as graphics display memory or cache memory).
- The main computer memory is usually formed from **Dynamic RAM** (DRAM), which uses an array of "capacitor" storage elements. Although slower than SRAM, it is also much cheaper, and therefore it usually possible to buy many MB of Dynamic RAM per computer.

ROM

- This is also known as non-volatile memory.
- Here, the data is stored permanently i.e. even if the power supply goes off, the stored data is not lost.
- It is usually used for storing programs and data in a computer.
- **PROM** (Programmable Read Only Memory) can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM program.
- **EPROM** (Erasable and Programmable Read Only Memory) can be erased by exposing it to ultra-violet light for duration of up to 40 minutes.
- **EEPROM** (Electrically Erasable and Programmable Read Only Memory) is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. In EEPROM, any location can be selectively erased and programmed.

1.6.5 MEMORY STRUCTURE

- Read/write memories consist of an array of registers, in which each register has unique address.
- The size of the memory is N x M as shown below where N is the number of registers and M is the word length, in number of bits.
- The address pins are connected as input to the decoder. If the number of address pins is n, the number of decoder output or the number of memory locations will be 2^n.
- The number of address lines of microprocessor depends on the size of the memory.

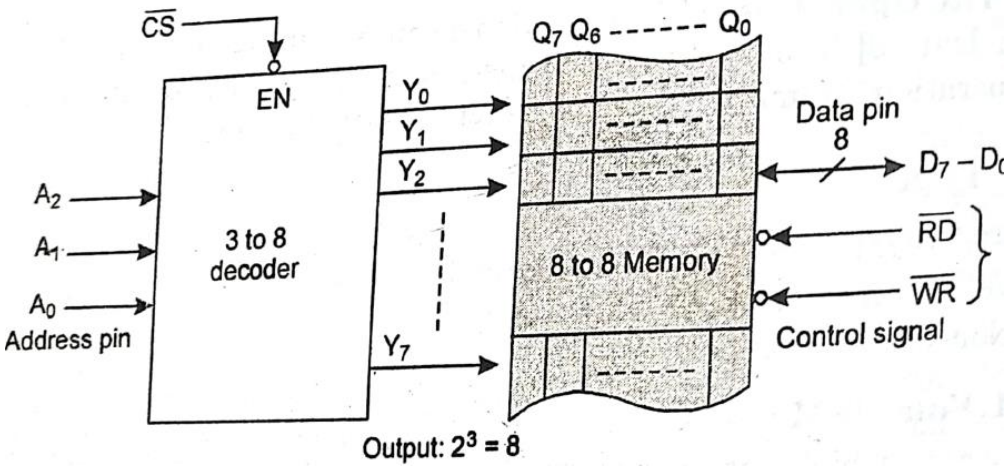


Fig.42 Block diagram of memory device

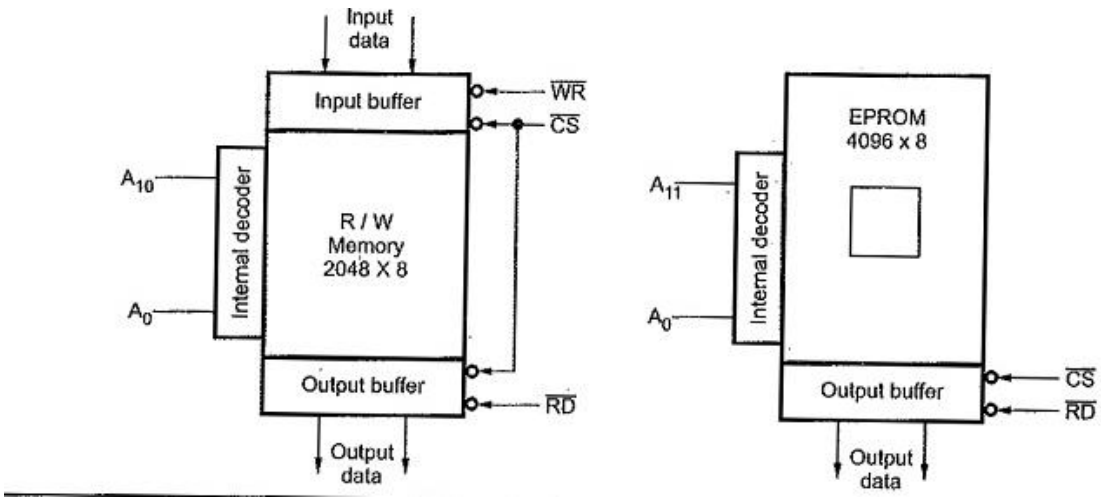


Fig.43 Logic diagram for RAM & EPROM

Examples

i. If memory is having 12 address lines and 8 data lines, then Number of registers/ memory locations (capacity) = $\mathbf{2^N = 2^{12} = 4096}$

Word length = M bit = 8 bit

ii. If memory has 8192 memory locations, then it has 13 address lines.

Table shows the details of different EPROM ICs and how the number of address lines is decided depending on the size of memory.

Sr. No.	IC Number	Memory Size Address Data	Number of Pins
1.	2716	2k × 8	24
2.	2732	4k × 8	24
3.	2764	8k × 8	28
4.	27128	16k × 8	28
5.	27256	32k × 8	28
6.	27512	64k × 8	28

The number of address line of microprocessor depend on the size of the memory.

Sr. No.	Number of Address Pin (n)	Number of Memory Location (2n) (Size of memory in byte)
1.	8	2^8 = 256 location
2.	9	2^9 = 512 location
3.	10	2^{10} = 1024 location
4.	11	2^{11} = 2048 = 2k location
5.	12	2^{12} = 4096 = 4k location
6.	13	2^{13} = 8192 = 8k location
7.	14	2^{14} = 16384 = 16k location
8.	15	2^{15} = 32768 = 32k location
9.	16	2^{16} = 65536 = 64k location

- Address Pins: The memory chip will have address pin to accept the address value.
  For example:
  i. For 1KB memory, there will be 10 address lines from A_0 to A_9.
  ii. For 2KB memory, there will be 11 address lines from A_0 to A_{10}.

- Data Pins: If each memory location contains SFIF, then after selecting one memory location, 8-bit data will be transferred in parallel. So, there will be 8 data pins.
  But if any memory location contains 4FIF in each location, then after selecting one memory location, only four bits of data will be transferred in parallel. So, there will be 4 data pins.

- Control Pins:
  i. $\overline{RD}$ – For reading 8-bit data from one memory location, microprocessor has to give logic 0 on this pin of memory IC.
  ii. $\overline{WR}$ - For storing 8-bit data from one memory location, microprocessor has to give logic 0 on this pin of memory IC.
  iii. $\overline{CS}$ – This is chip select pin. When logic 0 is given, the decoder is enabled. Hence, depending upon the address on the address pin, anyone decoder output becomes active and one memory location is selected.
  When logic 1 is applied, the decoder is disabled and even if some address is applied on the address pin, all the decoder outputs remain inactive.

1.6.6 MEMORY STRUCTURE

- Mapping means connecting or interfacing. Thus, memory mapping means connecting memory IC with microprocessor.
- Each memory location should relate to microprocessor such that:
  i. Each memory location should have different 16-bit address.
  ii. Each memory location can store 8 bits of data
- For memory mapping, the corresponding pin of memory should relate to the corresponding pin of microprocessor.

Examples

i. Indicate the different pins in 1k x 8 ROM or 1K ROM.

- ✓ 1k location means 10 address pins from A_0 to A_9.
- ✓ x 8 (byte) means each location has 8 bits. So, there will be 8 data pins from D_0 to D_7.
- ✓ ROM has control signals $\overline{RD}$ *and* $\overline{CS}$.
- ✓ Power Supply of + 5V and ground are present in all the IC.

ii. Indicate the different pins in 2k x 8 EPROM or 2K EPROM.

- ✓ 2k location means 10 address pins from A_0 to A_9.
- ✓ x 8 (byte) means each location has 8 bits. So, there will be 8 data pins from D_0 to D_7.
- ✓ ROM has control signals $\overline{RD}$ *and* $\overline{CS}$.
- ✓ Power Supply of + 5V and ground are present in all the IC.
- ✓ EPROM has one input +VPP (Program Voltage Pin).

iii. Indicate the different pins in 4k x 4 RAM or 4K nibble RAM.

- ✓ 4k location means 12 address pins from A_0 to A_{11}.
- ✓ x 4 (nibble) means each location has flip-flops. So, there will be 4 data pins from D_0 to D_3. As each location contains four flip-flops, hence, we have to connect two ICs in parallel.
- ✓ ROM has control signals $\overline{RD}, \overline{WR}$ *and* $\overline{CS}$.
- ✓ Power Supply of + 5V and ground are present in all the IC.

The following steps are involved in interfacing memory with 8085 processor:

- First decide the size of memory requires to be interfaced. Depending on this we can say how many address lines are required for it. For example, if you want to interface 4KB (212) memory it requires 12 address lines. Remaining address lines can be used in address decoding.
- Depending on the size of memory required and given address range, construct address decoding circuitry. This address decoding circuitry can be implemented with NAND gates and/or decoders or using PAL (when board size is a constraint).
- Connect data bus of memory to processor data bus.
- Generate the control signals required for memory using IO/M', WR', RD' signals of 8085 processor.

1.6.7 ADDRESS DECODING

- The result of 'address decoding' is the identification of a register for a given address.
- A large part of the address bus is usually connected directly to the address inputs of the memory chip.
- This portion is decoded internally within the chip.
- What concerns us is the other part that must be decoded externally to select the chip. This can be done either using logic gates or a decoder.
- Address Decoding Techniques:
  - Absolute decoding/Full Decoding
  - Linear decoding/Partial Decoding

Absolute decoding:

In absolute decoding technique, all the higher address lines are decoded to select the memory chip, and the memory chip is selected only for the specified logic levels on these high-orders address lines; no other logic levels can select the chip. Figure below shows the Memory Interfacing in 8085 with absolute decoding. This addressing technique is normally used in large memory systems.

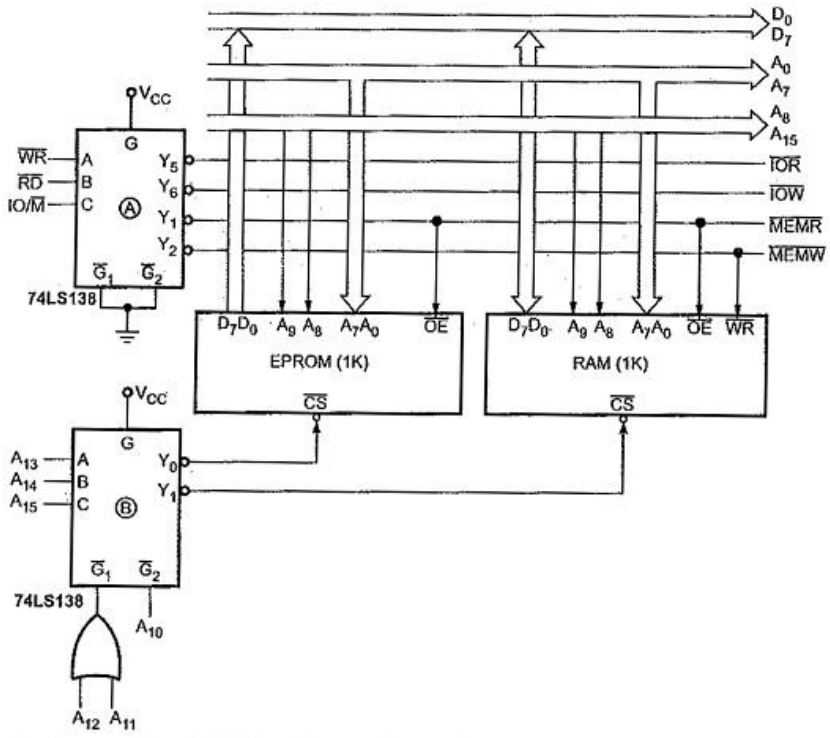


Fig.44 Absolute Decoding

Linear decoding:

In small systems, hardware for the decoding logic can be eliminated by using individual high-order address lines to select memory chips. This is referred to as linear decoding. Figure shows the addressing of RAM with linear decoding technique. This technique is also called **partial decoding**. It reduces the cost of decoding circuit, but it has a drawback of multiple addresses (shadow addresses).

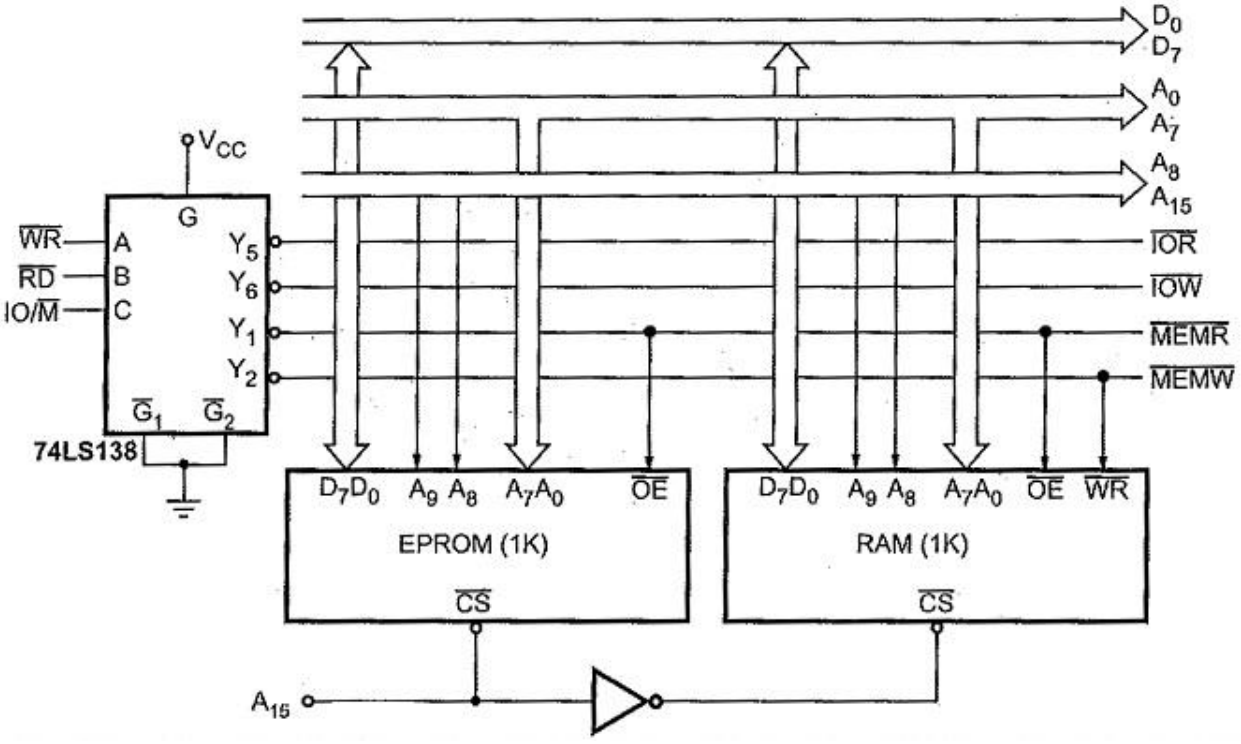


Fig.44 Linear Decoding

Figure 44 & 45 shows the addressing of RAM with linear decoding technique. A_{15} address line, is directly connected to the chip select signal of EPROM and after inversion it is connected to the chip select signal of the RAM. Therefore, when the status of A_{15} line is 'zero', EPROM gets selected and when the status of A_{15} line is 'one' RAM gets selected. The status of the other address lines is not considered, since those address lines are not used for generation of chip select signals.

1.6.8 APPLICATION OF MEMORY MAPPING

The 8085 microprocessor has 16 address lines. Therefore, it can access 2^{16} locations in the physical memory. If all these lines are connected to a single memory device, it will decode these 16 address lines internally and produces 216 different addresses from 0000H to FFFFH so that each location in the memory will have a unique address.

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	Hex Address
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000H
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0001H
0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0002H
- - - -	- - - -	- - - -	- - - -	- - - -
- - - -	- - - -	- - - -	- - - -	- - - -
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0	FFFEH
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0	FFFFH

Fig.45 Memory Address

Above diagram shows the various memories addresses used in Microprocessor. If more than one chip is used then some logic must be used to select one chip. This is done with the help of decoder.

74LS138 address decoder to generate the chip select signals for each memory block. In this decoder when the address lines A_{13}, A_{14} and A_{15} are 000, the output line Y_0 will be activated. This in turn selects the first memory block. Similarly, when these lines are 001 (C=0, B=0 and A=1) Y1 will be activated, and the second memory block will be selected.

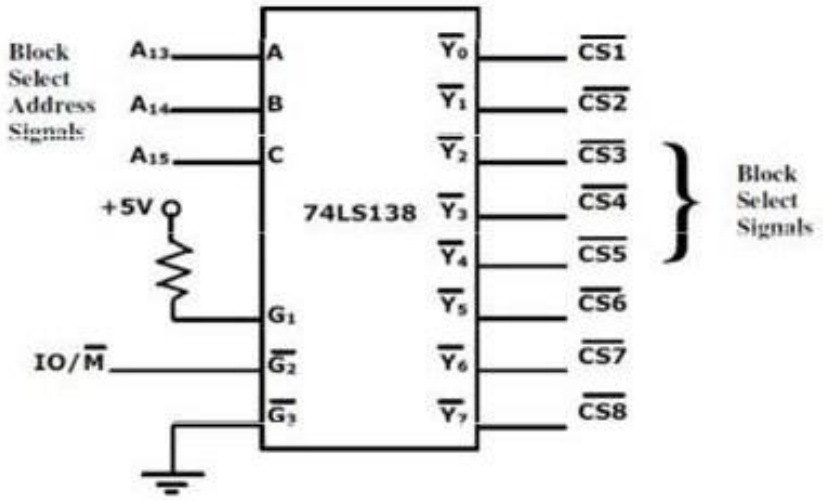


Fig.46 Memory block decoder

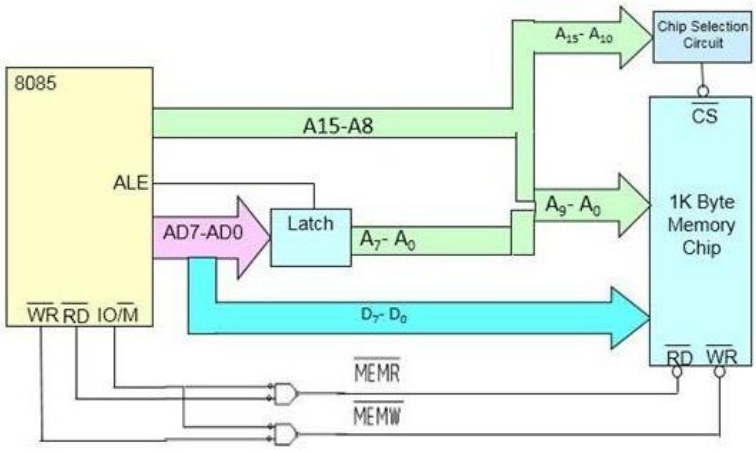


Fig.47 The complete interfacing diagram

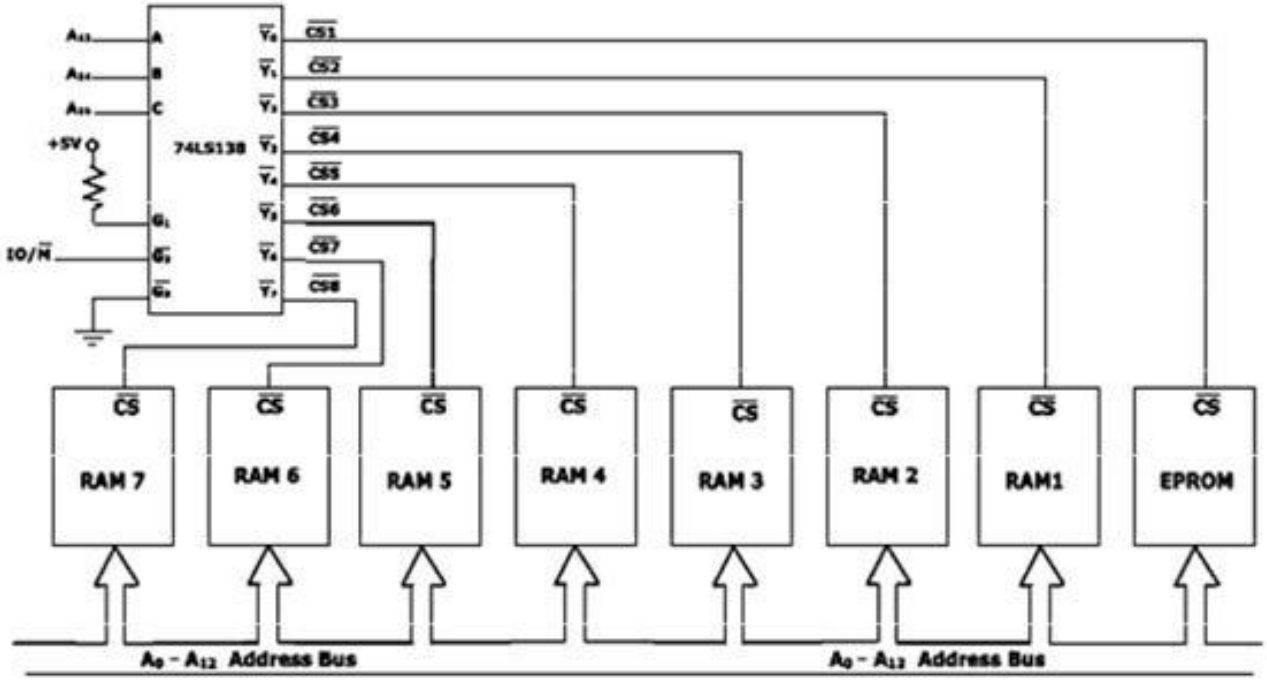


Fig.48 Memory Interface Diagram

Examples

i. Connect 1k x 8 ROM with microprocessor 8085.

- ✓ 1k location means 10 address pins from A_0 to A_9.
- ✓ x 8 (byte) means each location has 8 bits. So, there will be 8 data pins from D_0 to D_7.
- ✓ ROM has control signals$\overline{RD}$ *and* $\overline{CS}$.
- ✓ Power Supply of + 5V and ground are present in all the IC.
- ✓ The connection of different memory pins with microprocessor pins are given below:

Memory Interfacing Table:

S. No.	Memory Pin	Microprocessor/System Pins
1.	A_9 to A_0 address pin	A_9 to A_0 address pin
2.	D_7 to D_0 data pin	D_7 to D_0 data pin
3.	$\overline{RD}$	$\overline{MRD}$
4.	$\overline{CS}$	Reset of the six upper address pin A_{15} to A_{10} are decoded to generate $\overline{CS}$

Adress Decoding Table:

IC Number	Hexa Address	Binary Address	
		A_{15} A_{14} A_{13} A_{12} A_{11} A_{10}	A_9 ..... A_0
2k × 8 ROM	Starting address 4000H	0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0
	End address 47FFH	0 1 0 0 0 0	1 1 1 1 1 1 1 1 1 1
		Decoded to generate $\overline{CS}$	$A_9 - A_0$ of system to $A_9 - A_0$ of 1k × 8 ROM

Interfacing Diagram:

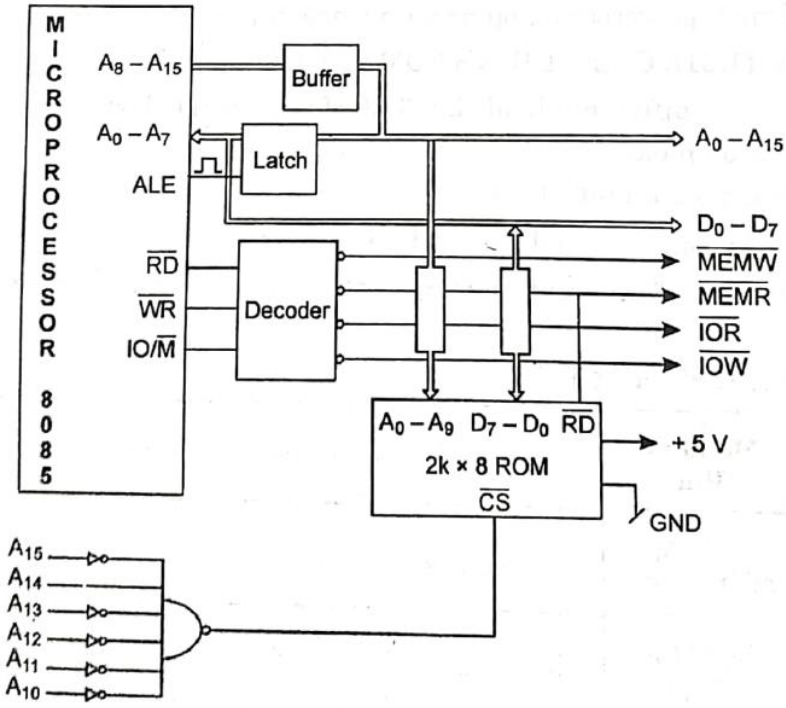


Fig.49 Interfacing Diagram

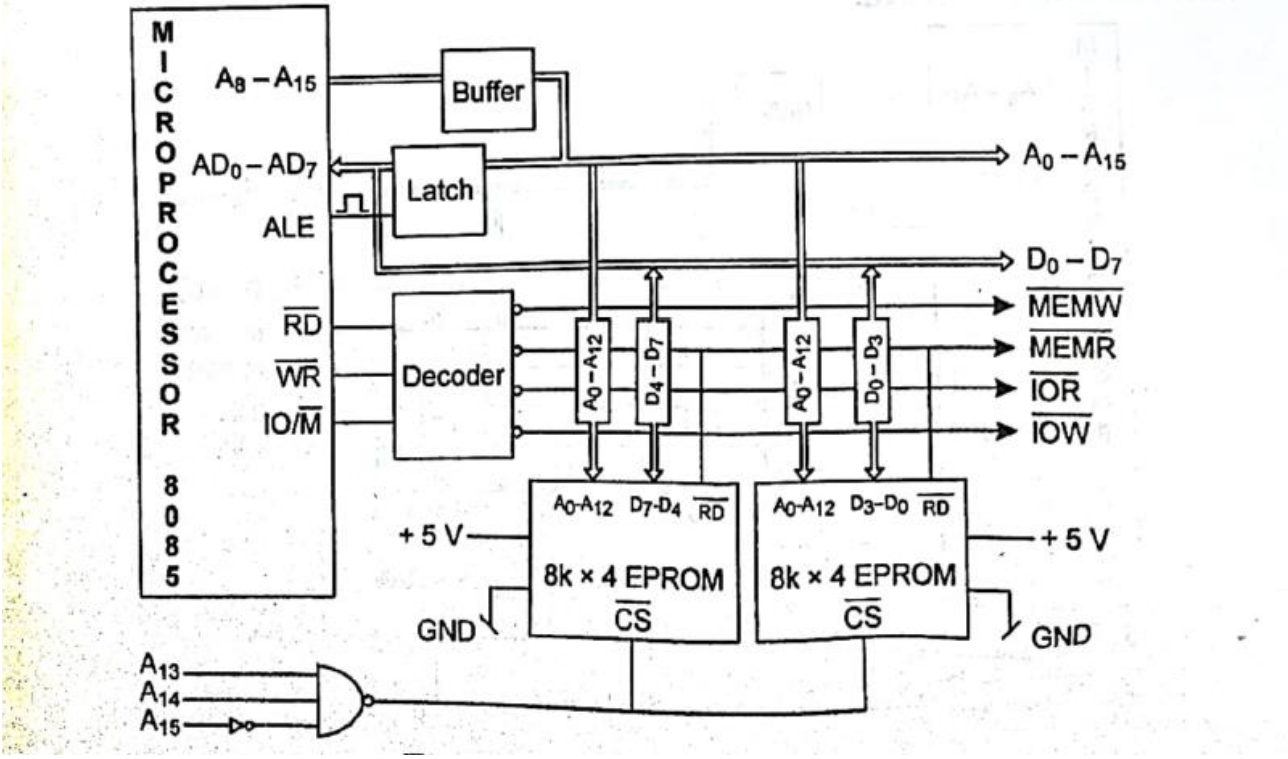
i. Connect 8k x 4 EPROM with 8085 CPU or with system bus with system line starting from address 6000H. The memory IC available is 8k nibble EPROM.

(a) 8k location means 13 address pin $A_{12} - A_0$.
(b) X4 means 4 FIF in each location. So there will be four data pin D_3 to D_0.
(c) EPROM has control Signal $\overline{RD}$.
(d) $\overline{CS}$, 5V, ground, V_{PP}.

Address Decoding Table:

Memory IC	Hexa Address	Binary Address	
		A_{15} A_{14} A_{13} A_{12} ..... A_0	
8k ×4 EPROM 2k in parallel = 2k × 5 EPROM	Starting address 6000	0 1 1	0 0 0 0 0 0 0 0 0 0 0 0
	End address 7FFF	0 1 1	1 1 1 1 1 1 1 1 1 1 1 1
		$A_{15} - A_{13}$ decoded to generate $\overline{CS}$	$A_{12} - A_0$ of system pin

Interfacing Diagram:



1.6.9 UNIVERSITY QUESTIONS RELATED TO THE TOPIC

Two-mark Questions

Q-1. Specify the memory addressing capacity of 8085 microprocessor.

Q-2. How many address lines are required to address 2MB memory?

Q-3. If a processor has 4 GB memory then how many address lines are required to access this memory?

Q-4. Distinguish between SRAM & DRAM.

Q-5. Write a short note on RAM.

Q-6. Explain different types of ROM.

Q-7. Distinguish between EPROM & EEPROM.

Q-8. Explain control signals WR, RD and chip select logic.

Five-mark Questions

Q-9. Connect 2k x 4 RAM two IC with 8085 microprocessors from address 9000H.

Q-10. Connect 1k x 8 ROM with the system line of 8085 from address 0400 H.

Q-11. Connect 16 K x 8 EPROM with system buses from address 8000H.

Ten-mark Questions

Q-12. Connect the following memory IC with 8085 CPU:

1. 1k x 8 ROM
2. 1k x 8 RAM

1.7 INPUT OUTPUT INTERFACING

1.7.1 WHAT IS I/O INTERFACING

- The data transfer between input device (for example: keyboard, punch card etc.) and microprocessor and between microprocessor and display device is called input / output data transfer or I/O data transfer.
- The data transfer is done with the help of input/output port.
- Interfacing can be done in groups of 8 bits using the entire data bus. This is called parallel I/O.
- The other method is serial I/O where one bit is transferred at a time using the SID and SOD pins on the Microprocessor.
- There are two ways to interface 8085 with I/O devices in parallel data transfer mode:
    i. Memory Mapped I/O interfacing
    ii. IO Mapped I/O interfacing

1.7.2 NEED OF I/O INTERFACING

1.7.3 INPUT PORTS

- It is used to transfer data from the outside to microprocessor such as ADC. We have to use buffer between microprocessor and input device.
- This buffer is a tri-state and its output is available only when enable signal is active.
- When microprocessor wants to read data from the input device (keyboard), the control signals from the microprocessor activates the buffer by asserting enable input of the buffer.
- Once the buffer is enabled, data from the input device is available on the data bus. Microprocessor reads this data by initiating read command.

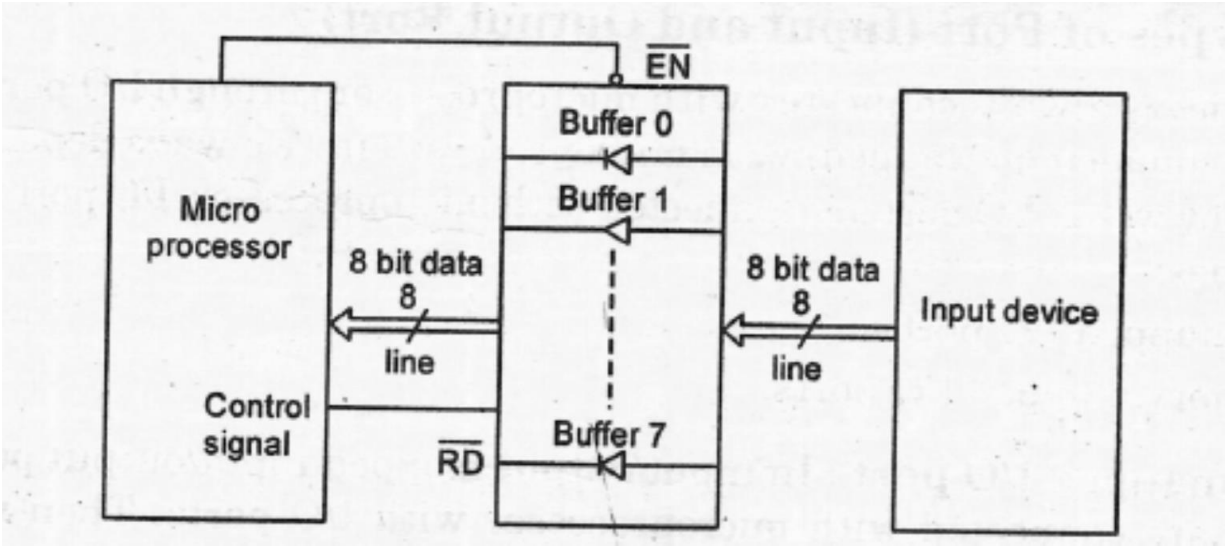


Fig.50 Input Port Interfacing

1.7.4 OUTPUT PORTS

- It is used to transfer data from microprocessor to outside and is called output port. The output device is connected to the microprocessor through a latch.
- When microprocessor wants to send data to the output device, it puts the data on the data bus and activates the clock signal of the latch, latching the data from the data bus at the output of latch. It is then available at the output of latch for the output device.

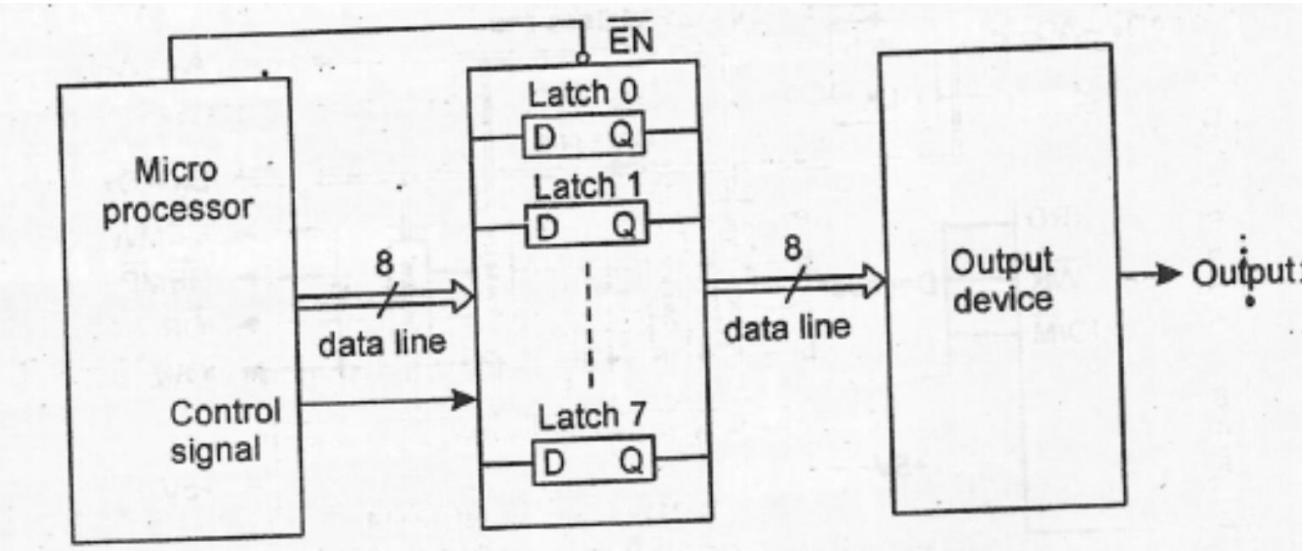


Fig.51 Output Port Interfacing

1.7.5 TYPES OF I/O PORTS

The I/O device is always connected with microprocessor through I/O port. This I/O port can be connected / mapped with microprocessor in two ways, depending upon the method by which I/O ports are connected with microprocessor..

I/O ports are divided into two types:

- I/O mapped I/O ports: Here the I/O port are connected with microprocessor with I/O ports. The control signal is used for I/O $\overline{\text{IORD}}$ for input port and $\overline{\text{IOWR}}$ for output port. Then, it is called I/O mapped I/O ports.
  The instruction which can be used to perform data transfer between microprocessor and I/O ports are IN and OUT.
- Memory mapped I/O ports: Here the I/O ports are connected with the microprocessor like one memory location. Address is assigned to each port from 0000H to FFFFH.
  The control signals used are $\overline{MRD}$ for input port and $\overline{MWR}$ for output port.

Diffecrence between I/O Mapped I/O Interfacing & Memory Mappd I/O Interfacing

Memory mapped I/O	I/O mapped I/O
1. In this device address is 16 bit. Thus A_0 to A_{15} lines are used to generate device address.	1. In this I/O device address is 8 bit. Thus A_0 to A_7 or A_8 to A_{15} lines are used to generate device address.
2. $\overline{MEMR}$ and $\overline{MEMW}$ control signals are used to control read and write I/O operations.	2. $\overline{IOR}$ and $\overline{IOW}$ control signals are used to control read and write I/O operations.
3. Instructions available are LDA addr, STA addr, LDAX rp, STAX rp, MOV M,R, MOV R,M ADD M, CMP M etc.	3. Instructions available are IN and OUT.
4. Data transfer is between any register and I/O device.	4. Data transfer is between accumulator and I/O device.
5. Maximum number of I/O devices are 65536 (theoretically).	5. Maximum number of I/O devices are 256.
6. Execution speed using LDA addr, STA addr is 13 T-state and 7 T-states for MOV M, r and MOV r, M instructions.	6. Execution speed is 10 T-states.
7. Decoding 16 bit address may require more hardware.	7. Decoding 8 bit address will require less hardware.

1.7.6 APPLICATION OF INTERFACING INPUT AND OUTPUT DEVICES

Interfacing of Input Devices

- The address lines are decoded to generate a signal that is active when the particular port is being accessed.
- An IORD signal is generated by combining the IO/M and the RD signals from the microprocessor.
- A tri-state buffer is used to connect the input device to the data bus.
- The control (Enable) for these buffers is connected to the result of combining the address signal and the signal IORD.
- The microprocessor 8085 accepts 8 bit data from the input device such as keyboard, sensors, transducers etc.
- Fig.39 below shows the circuit diagram to Input Output Interfacing Techniques (buffer) which is used to read the status of 8 switches.
- The address for this input device is 80H as device select signal goes low when address is 80H.
- When the switch is in the released position, the status of line is high otherwise status is low. With this information microprocessor can check a particular key is pressed or not.
- The following program checks whether the switch 2 is pressed or not.

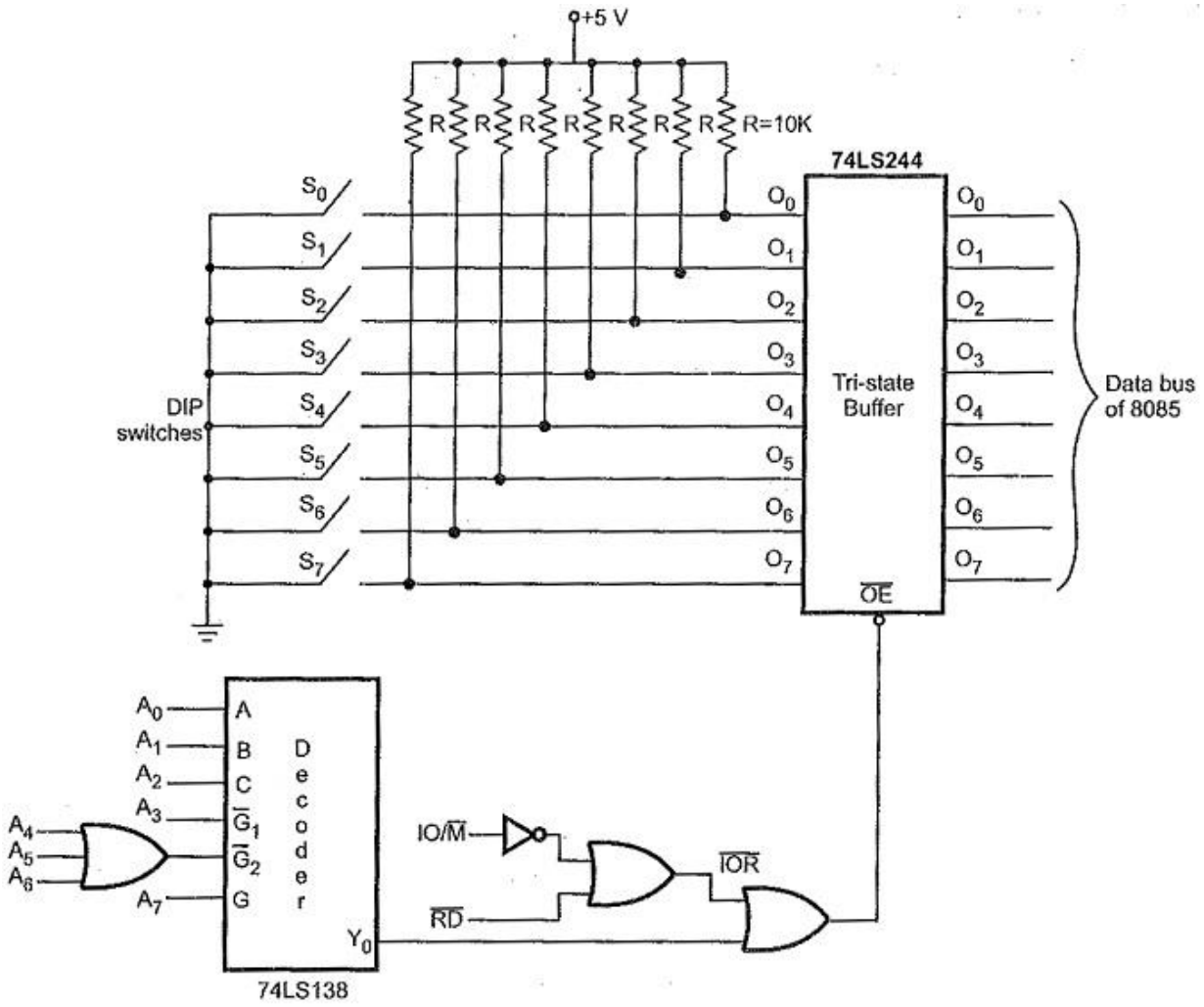


Fig.52 Input Port Interfacing Techniques

Interfacing of Output Devices

- Output devices are usually slow.
- Also, the output is usually expected to continue appearing on the output device for a long period of time.
- Given that the data will only be present on the data lines for a very short period (microseconds), it has to be latched externally.
- To do this the external latch should be enabled when the port's address is present on the address bus, the IO/M signal is set high and WR is set low.
- The resulting signal would be active when the output device is being accessed by the microprocessor.
- Decoding the address bus (for memory-mapped devices) follows the same techniques discussed in interfacing memory.
- The microprocessor 8085 sends 8-bit data to the output device such as 7 segment displays, LEDs, printer etc.
- Figure 40 below shows the circuit diagram to interface output port (latch) which is used to send the signal for glowing the LEDs. LED will glow when output pin status is low.
- The IC 74LS138 and 3 input OR gate is used to generate device select signal.

- The latch enable signal is active high. So, NOR gate is used to generate latch enable signal, which goes high when Y_1 and IOW both are low.

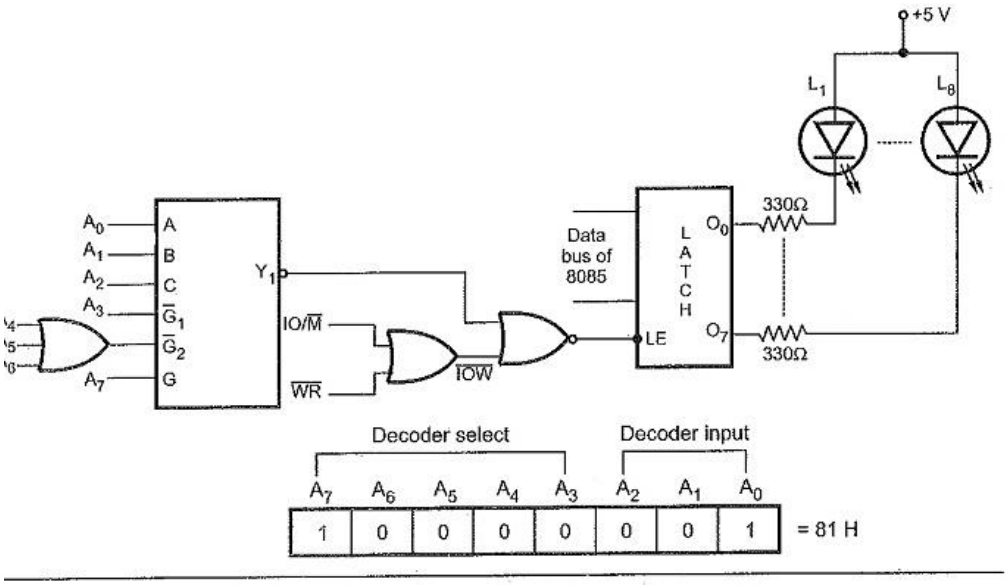


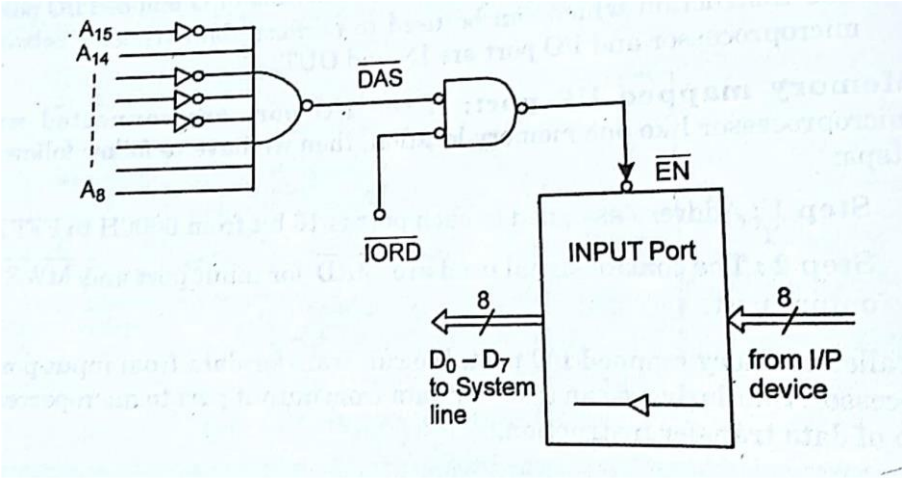
Fig.53 Output Port Interfacing Techniques

Examples

i. Connect one input port with microprocessor of address 47H.

Address Decoding Table:

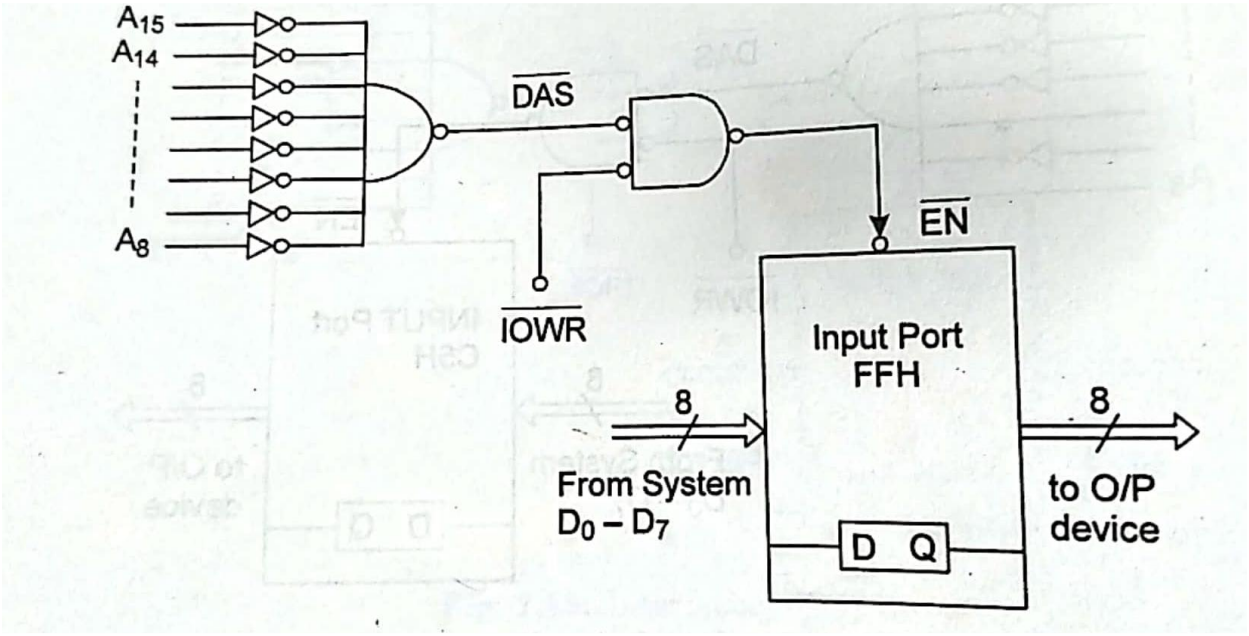
Memory IC	Hexa Address	Binary Address
		A_{15} ..... A_8 or A_7 ..... A_0
I/P Port	47H	0 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1
		Decoded to generate $\overline{DAS}$



ii. Connect one output port with system bus at address space FFH.

Address Decoding Table:

Port	Hexa Address	Binary Address
		A_{15} ..... A_8 or A_7 ..... A_0
Output port	FFH	1 1 1 1 1 1 1 1 Decoded to generate $\overline{DAS}$



1.8 MEMORY MAPPED I/O

1.8.1 WHAT IS MEMORY MAPPED I/O

- I/O interfaces can be input only, output only or bidirectional (input and output).
- Generally, an I/O interface includes of a set of registers through which the CPU can read or write data.
- These I/O registers can be classified into one of three types: data, control and status.
- A data register is used for exchanging data with the interface, a control register is for configuring or controlling the operation of the interface, and a status register indicates information about the state or condition of the interface.

- An I/O interface can include any number and combination of these registers, from a simple I/O interface consisting of a single data register to a complex I/O interface with several data, control and status registers.
- Some registers are subdivided into individual bits that serve as status, control or data bits individually

1.8.2 NEED OF MEMORY MAPPED I/O

- To communicate with the I/O interface, the CPU must have the ability to read and write I/O interface registers.
- Just as with memory, each of these I/O registers is assigned a unique identifier through which it is addressed by the CPU.
- These I/O addresses form an address space that can either be part of the CPU's memory address space or be separate from it.
- When memory and I/O share the same address space, the CPU is said to use memory-mapped I/O; otherwise, it is said to use separate I/O. Since memory-mapped I/O interface registers are mapped into the CPU's memory address space, these registers are manipulated by software in the same way as memory bytes, using existing CPU instructions and addressing modes. This allows much flexibility when working with I/O registers, but reduces the amount of the address space that can be mapped to RAM or ROM.

1.8.3 HOW MEMORY MAPPED I/O WORKS?

We must use the following steps:

- Select 16-bit port address (0000H – FFFFH).
- Control signal for memory mapped input/output are $\overline{\text{MRD}}$ and $\overline{\text{MWR}}$.
- Consider an example in which address bit A15 designates whether instructions reference memory or an I/O device.
- If A15= 0, a memory register is addressed; If A15= 1, than a memory mapped I/O device is address.
- This assignment elevates the first 32k of memory address space to memory and second 32k to memory mapped I/O devices.
- External logic generates devices select pulses for memory mapped I/O only when $\text{IO}/\overline{\text{M}}$ = 0, the appropriate address is on the address low and a $\overline{\text{WR}}$ or $\overline{\text{RD}}$ strobe occurs.
- Input and output transfer using memory mapped I/O are not limited to the accumulator. For example, same of 8085 A instructions that can be used for input from memory mapped I/O ports.
- MOV r, m move the contents of input port whose address is available in (H, L) reg pair to any internal register.
- Other instructions include, ANA M, ADD M, 1HD add (input from two ports and store the contents is reg pair (L) and (H) ADD M and ANA M provide input data transfer and

computation in a single instruction. same instruction that out the data from memory mapped ports are

- LHLD and SHLD carry out 16- bit I/O transfers with single instructions which reduce program executive time considerably.
- The price paid for this added capability is a reduction in directly addressable main memory and the necessity of decoding a 16- bit rather than an 8-bit address.
- When a microprocessor puts out an address and generates a control strobe for a memory read, it has no way of determining whether the device that responds with data is a memory device or an I/O device; nor does it care.
- It only requires that the devices that respond do so with in the allowable access time or uses the READY line to request enough WAIT states.

1.8.4 APPLICATIONS OF MEMORY MAPPED I/O

i. Interface one input port and one output port with 8085 in memory mapped configuration.

We know that in memory mapped configuration, input/output port is like memory location. Let the address assigned to input/output port is 8000H.

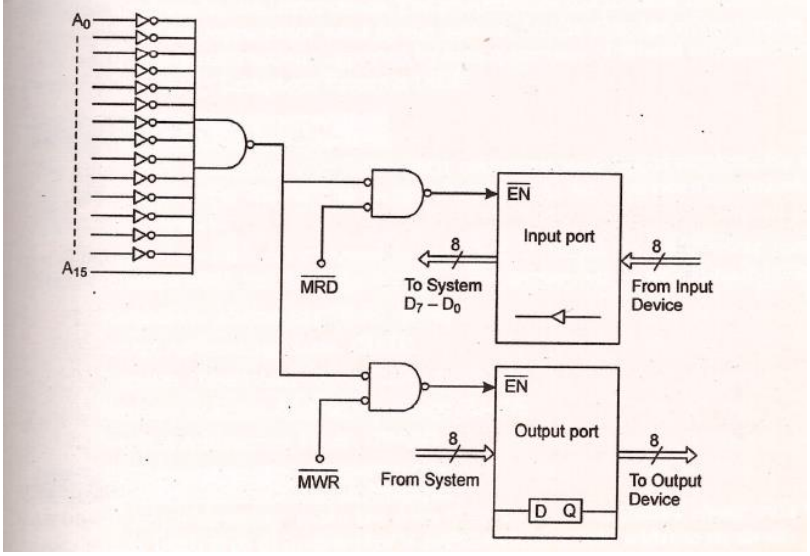
Address Decoding Table:

Port	Hexa Address	Binary Address
		A_{15} ..... A_0
One input port one output port	8000H	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
		Decoded to generate $\overline{DAS}$

Control Signal:

- Connect control signal for input port will be $\overline{MRD}$.
- Connect control signal for output port will be $\overline{MWR}$.

Interfacing Diagram:



1.8.5 UNIVERSITY QUESTIONS RELATED TO THE TOPIC

Two-mark Questions

Q-1. Differentiate between I/O Mapped I/O and Memory Mapped I/O.
Q-2. What is a memory device?
Q-3. Explain Memory Interfacing.
Q-4. Explain various data pins.

Five-mark Questions

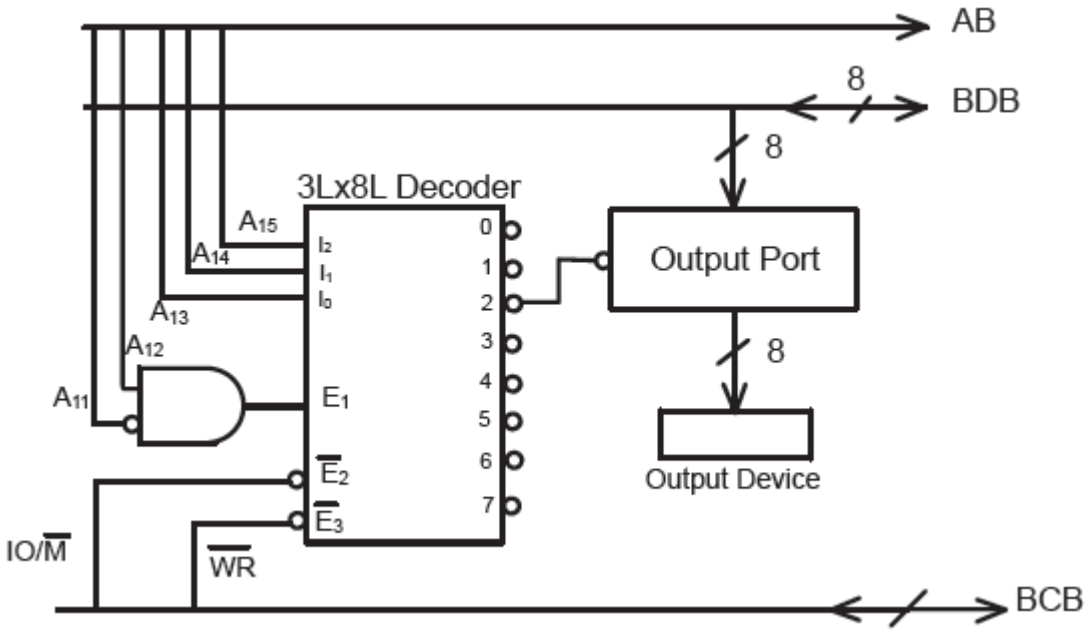
Q-5. Interface one output port with microprocessor with system line.
Q-6. Connect one input port with microprocessor 8085.
Q-7. Interface one input and one output port with microprocessor 8085.

Ten-mark Questions

Q-8. Explain how chip select logic can be used using GATES.
Q-9. Explain how chip select logic can be used using decoders.

GATE Questions

Q-10. An output device is interfaced with 8-bit microprocessor 8085A. The interfacing circuit is shown in figure

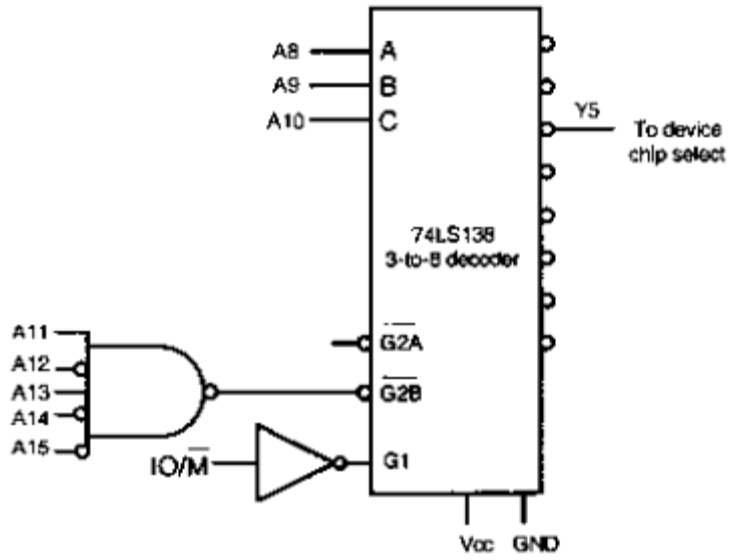


The interfacing circuit makes use of 3 Line to 8 Line decoder having 3 enable lines E1, E2' & E3'. The address of the device is

a. 50_H
b. 5000_H
c. $A0_H$
d. $A000_H$

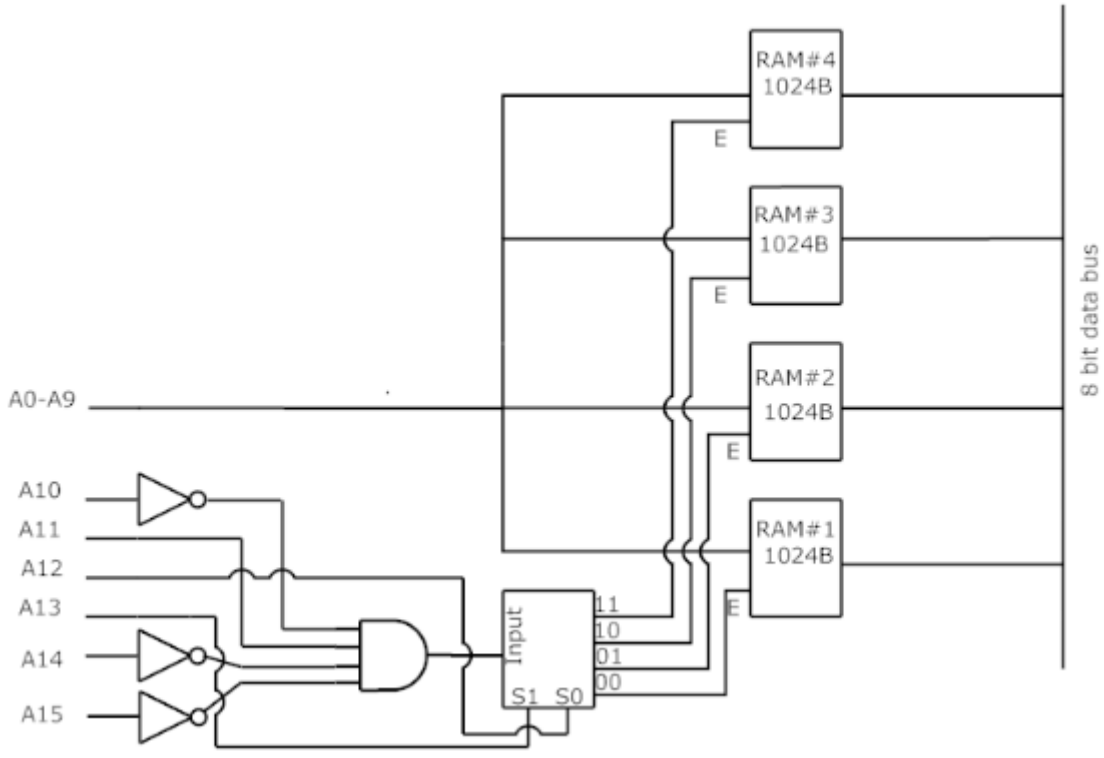
Answer: b

Q-11. In the circuit shown, the device connected to Y5 can have address in the range



Answer: 2D00 – 2DFF

Q-13. There are four chips each of 1024 bytes connected to a 16 bit address bus as shown in the figure below. RAMs 1, 2, 3 and 4 respectively are mapped to addresses



Answer: 0800H-0BFFH, 1800H-1BFFH, 2800H-2BFFH, 3800H-3BFFH

List of Terms Used in a Microprocessor

- **Bit**: A bit is a single binary digit.
- **Word**: A word refers to the basic data size or bit size that can be processed by the arithmetic and logic unit of the processor. A 16-bit binary number is called a word in a 16-bit processor.
- **Bus**: A bus is a group of wires/lines that carry similar information.
- **System Bus**: The system bus is a group of wires/lines used for communication between the microprocessor and peripherals.
- **Memory Word**: The number of bits that can be stored in a register or memory element is called a memory word.
- **Address Bus**: It carries the address, which is a unique binary pattern used to identify memory location or an I/O port. For example, an eight bit address bus has eight lines and thus it can address 2^8 = 256 different locations. The locations in hexadecimal format can be written as 00H – FFH.
- **Data Bus**: The data bus is used to transfer data between memory and processor or between I/O device and processor. For example, an 8-bit processor will generally have an 8-bit data bus and a 16-bit processor will have 16-bit data bus.
- **Control Bus**: The control bus carry control signals, which consists of signals for selection of memory or I/O device from the given address, direction of data transfer and synchronization of data transfer in case of slow devices.
- **Instruction Set**: It is the set of instructions that the microprocessor can understand. The instruction set of a microprocessor is provided in two forms: *binary machine code and mnemonics*.
- **Clock Speed**: It determines the number of operations per second the processor can perform. It is expressed in megahertz (MHz) or gigahertz (GHz).It is also known as Clock Rate.
- **Word Length**: It depends upon the width of internal data bus, registers, ALU, etc. An 8-bit microprocessor can process 8-bit data at a time. The word length ranges from 4 bits to 64 bits depending upon the type of the microcomputer.
- **Data Types**: The microprocessor has multiple data type formats like binary, BCD, ASCII, signed and unsigned numbers.