EE6502 MICROPROCESSORS AND MICROCONTROLLERS

16 MARKS QUESTION AND ANSWERS

BY

ENSOLT ELAKKIA

EE6502 MICROPROCESSORS AND MICROCONTROLLERS

UNIT I 8085 PROCESSOR

Hardware Architecture, Pinouts – Functional Building Blocks of Processor – Memory organization – I/O ports and data transfer concepts– Timing Diagram – Interrupts.

UNIT II PROGRAMMING OF 8085 PROCESSOR

Instruction -format and addressing modes – Assembly language format – Data transfer, data manipulation& control instructions – Programming: Loop structure with counting & Indexing – Look up table - Subroutine instructions - stack.

UNIT III 8051 MICRO CONTROLLER

Hardware Architecture, Pintouts – Functional Building Blocks of Processor – Memory organization – I/O ports and data transfer concepts– Timing Diagram – Interrupts-Comparison to Programming concepts with 8085.

UNIT IV PERIPHERAL INTERFACING

Study on need, Architecture, configuration and interfacing, with ICs: 8255 , 8259 , 8254,8237,8251, 8279 ,- A/D and D/A converters &Interfacing with 8085& 8051 UNIT V MICRO CONTROLLER

PROGRAMMING & APPLICATIONS

Data Transfer, Manipulation, Control Algorithms& I/O instructions – Simple programming exercises key board and display interface – Closed loop control of servo motor- stepper motor control – Washing Machine Control.

TEXT BOOKS:

1. Krishna Kant, “Microprocessor and Microcontrollers”, Eastern Company Edition, Prentice Hall of India, New Delhi , 2007.

2. R.S. Gaonkar, ‘Microprocessor Architecture Programming and Application’, with 8085, Wiley Eastern Ltd., New Delhi, 2013.

3. Soumitra Kumar Mandal, Microprocessor & Microcontroller Architecture, Programming &Interfacing using 8085,8086,8051,McGraw Hill Edu,2013.

REFERENCES:

1. Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely ‘The 8051 Micro Controller and Embedded Systems’, PHI Pearson Education, 5th Indian reprint, 2003.

2. N.Senthil Kumar, M.Saravanan, S.Jeevananthan, ‘Microprocessors and Microcontrollers’, Oxford, 2013.

3. Valder – Perez, “Microcontroller – Fundamentals and Applications with Pic,” Yeesdee Publishers, Tayler & Francis, 2013.

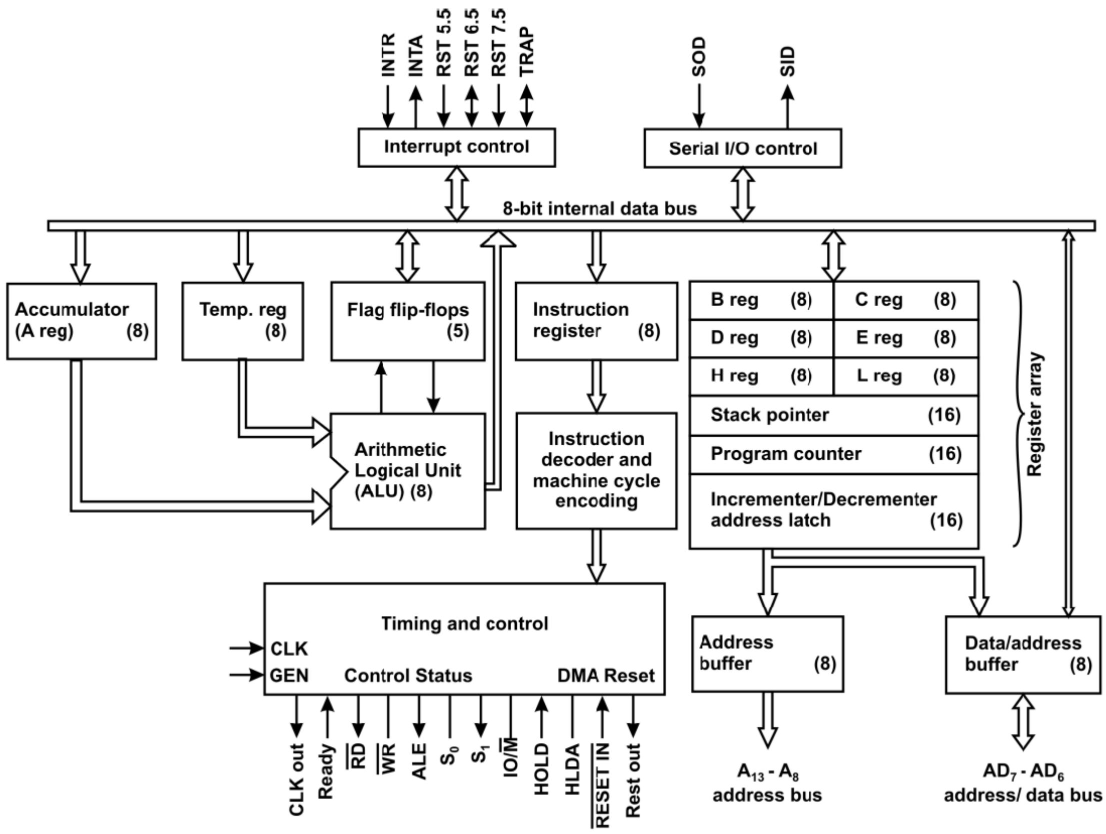
1. Explain with neat block diagram the architecture of 8085 microprocessor.

Architecture of 8085 microprocessor

8085 consists of various units and each unit performs its own functions. The various units of a microprocessor are listed below

Accumulator
Arithmetic and logic Unit
General purpose register
Program counter
Stack pointer
Temporary register
Flags
Instruction register and Decoder
Timing and Control unit
Interrupt control
Serial Input/output control
Address buffer and Address-Data buffer
Address bus and Data bus

8085 Architecture



Accumulator

Accumulator is nothing but a register which can hold 8-bit data. Accumulator aids in storing two quantities. The data to be processed by arithmetic and logic unit is stored in accumulator. It also stores the result of the operation carried out by the Arithmetic and Logic unit. The accumulator is also called an 8-bit register. The accumulator is connected to Internal Data bus and ALU (arithmetic and logic unit). The accumulator can be used to send or receive data from the Internal Data bus.

Arithmetic and Logic Unit

There is always a need to perform arithmetic operations like +, -, *, / and to perform logical operations like AND, OR, NOT etc. So there is a necessity for creating a separate unit which can perform such types of operations. These operations are performed by the Arithmetic and Logic Unit (ALU). ALU performs these operations on 8-bit data.

But these operations cannot be performed unless we have an input (or) data on which the desired operation is to be performed. So from where do these inputs reach the ALU? For this purpose accumulator is used. ALU gets its Input from accumulator and temporary register. After processing the necessary operations, the result is stored back in accumulator.

General Purpose Registers

Apart from accumulator 8085 consists of six special types of registers called General Purpose Registers. These general purpose registers are used to hold data like any other registers. The general purpose registers in 8085 processors are B, C, D, E, H and L. Each register can hold 8-bit data. Apart from the above function these registers can also be used to work in pairs to hold 16-bit data. They can work in pairs such as B-C, D-E and H-L to store 16-bit data. The H-L pair works as a memory pointer. A memory pointer holds the address of a particular memory location. They can store 16-bit address as they work in pair.

Program Counter and Stack Pointer

Program counter is a special purpose register. Consider that an instruction is being executed by processor. As soon as the ALU finished executing the instruction, the processor looks for the next instruction to be executed. So, there is a necessity for holding the address of the next instruction to be executed in order to save time. This is taken care by the program counter. A program counter stores the address of the next instruction to be executed. In other words the program counter keeps track of the memory address of the instructions that are being executed by the microprocessor and the memory address of the next instruction that is going to be executed.

Microprocessor increments the program whenever an instruction is being executed, so that the program counter points to the memory address of the next instruction that is going to be executed.

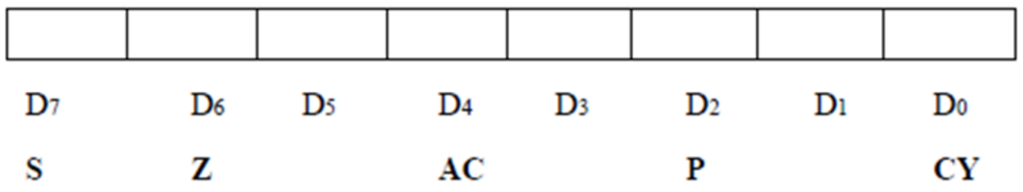
Program counter is a 16-bit register. Stack pointer is also a 16-bit register which is used as a memory pointer. A stack is nothing but the portion of RAM (Random access memory). So does that mean the stack pointer points to portion of RAM? Yes. Stack pointer maintains the address of the last byte that is entered into stack. Each time when the data is loaded into stack, Stack pointer gets decremented. Conversely it is incremented when data is retrieved from stack.

Temporary Register:

As the name suggests this register acts as a temporary memory during the arithmetic and logical operations. Unlike other 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. In the next article let us discuss about the FLAGS.

Flags

Flags are nothing but a group of individual Flip-flops. The flags are mainly associated with arithmetic and logic operations. The flags will show either a logical (0 or 1) (i.e.) a set or reset depending on the data conditions in accumulator or various other registers. A flag is actually a latch which can hold some bits of information. It alerts the processor that some event has taken place.+



Why are they called flags?

The possible solution is from the small flags which are found on the mail boxes in America. The small flag indicates that there is a mail in the mail box. Similarly this denotes that an event has occurred in the processor. Intel processors have a set of 5 flags.

Carry flag
Parity flag
Auxiliary carry flag
Zero flag
Sign flag

Consider two binary numbers. For example:

1100 0000

1000 0000

When we add the above two numbers, a carry is generated in the most significant bit. The number in the extreme right is least significant bit, while the number in extreme left is most significant bit. So a ninth bit is generated due to the carry. So how to accommodate 9th bit in an 8 bit register? For this purpose the Carry flag is used. The carry flag is set whenever a carry is generated and reset whenever there is no carry. But there is an auxiliary carry flag? What is the difference between the carry flag and auxiliary carry flag?

Let's discuss with an example. Consider the two numbers given below

0000 0100, 0000 0101 .

When we add both the numbers a carry is generated in the third bit from the least significant bit. This sets the auxiliary carry f lag. When there is no carry, the auxiliary carry flag is reset. So whenever there is a carry in the most significant bit Carry flag is set. While an auxiliary carry flag is set only when a carry is generated in bits other than the most significant bit. Parity checks whether its even or add parity. This flag returns a 0 if it is odd parity and returns a 1 if it is an even parity. Sometimes they are also called as parity bit which is used to check errors while data transmission is carried out. Zero flag shows whether the output of the operation is 0 or not. If the value of Zero flag is 0 then the result of operation is not zero. If it is zero the flag returns value

1. Sign flag shows whether the output of operation has positive sign or negative sign. A value 0 is returned for positive sign and 1 is returned for negative sign. Instruction Register and Decoder Instruction register is 8-bit register just like every other register of microprocessor. Consider an instruction. The instruction may be anything like adding two data's, moving a data, copying a data etc. When such an instruction is fetched from memory, it is directed to Instruction register. So the instruction registers are specifically to store the instructions that are fetched from memory. There is an Instruction decoder which decodes the information present in the Instruction register for further processing.

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: Control Signals: READY, $\overline{RD}$, $\overline{WR}$, ALE

Status Signals: S0, S1, IO/$\overline{M}$

DMA Signals: HOLD, HLDA

RESET Signals: RESET IN, 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 I/O device. After collecting the input data the control is transferred back to main program.

Interrupt signals present in 8085 are:

INTR
RST 7.5
RST 6.5
RST 5.5
TRAP

Of 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.
SIM

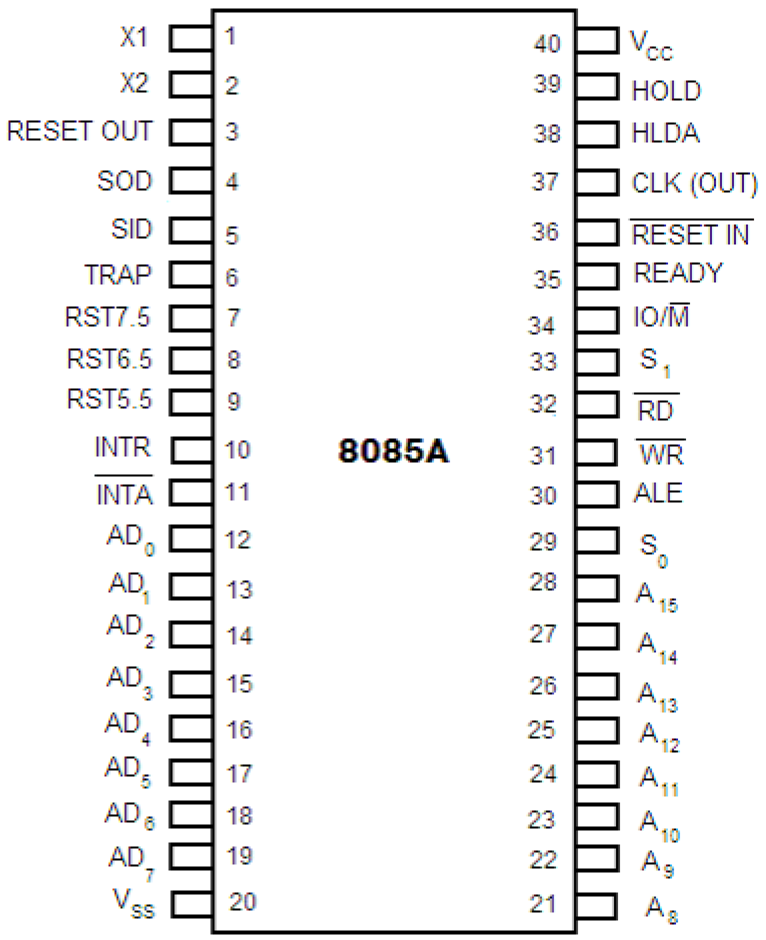
RIM

Address buffer and Address-Data buffer

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. Address bus and Data bus:

We know that 8085 is an 8-bit microprocessor. So the data bus present in the microprocessor is also 8-bits wide. So 8-bits of data can be transmitted from or to the microprocessor. But 8085 processor requires 16 bit address bus as the memory addresses are 16-bit wide. The 8 most significant bits of the address are transmitted with the help of address bus and the 8 least significant bits are transmitted with the help of multiplexed address/data bus. The eight bit data bus is multiplexed with the eight least significant bits of address bus. The address/data bus is time multiplexed. This means for few microseconds, the 8 least significant bits of address are generated, while for next few seconds the same pin generates the data. This is called Time multiplexing. But there are situations where there is a need to transmit both data and address simultaneously. For this purpose a signal called ALE (address latch enable) is used. ALE signal holds the obtained address in its latch for a long time until the data is obtained and so when the microprocessor sends the data next time the address is also available at the output latch. This technique is called Address/Data demultiplexing.

2. **Draw the pin diagram of 8085 microprocessor.**



3. Explain the types of memory.

Classification of ROM

ROM (Read Only Memory) The data is permanently stored in the memory and these devices are mask programmed during manufacturing. ROMs cannot be reprogrammed and are of nonvolatile type. These devices are cheaper than programmable memory devices. The applications of ROM are fixed programmed instructions, look-up tables, conversions, and some specific operations.

The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

PROM (Programmable Read Only Memory) The data can be electrically stored. It can be programmed by blowing built-in fuses. PROM can be reprogrammed and is of nonvolatile type. These memory devices are of very low memory density and occupy more space.

EPROM (Erasable Programmable Read Only Memory) These are strictly a MOS device and programmed by storing charge on insulated gates. These devices are erasable with ultraviolet rays and reprogrammable after erasing. These memory devices are of nonvolatile type.

EEPROM (Electrically Erasable Programmable Read Only Memory) These memory devices are electrically programmable by the programmer and the stored data can be erased by ultraviolet light. This is of nonvolatile type. This is also called Electrically Alterable Programmable Read Only Memory (EAPROM).

Classification of RAM

The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

Static RAMs These RAMs are built with static or dynamic cells. Five or six transistors are used to store a single bit. Data can be written and read in nanoseconds. Usually TTL, ECL, NMOS and CMOS technology are used to manufacture static RAMs. When the power is shut off, data stored in cells can be lost.

Dynamic RAMs In a dynamic memory, data can be stored on capacitors and to retain data every cell has to be refreshed periodically. One transistor is used to build a memory cell and requires less space. These memories consume less power compared to static RAMs.

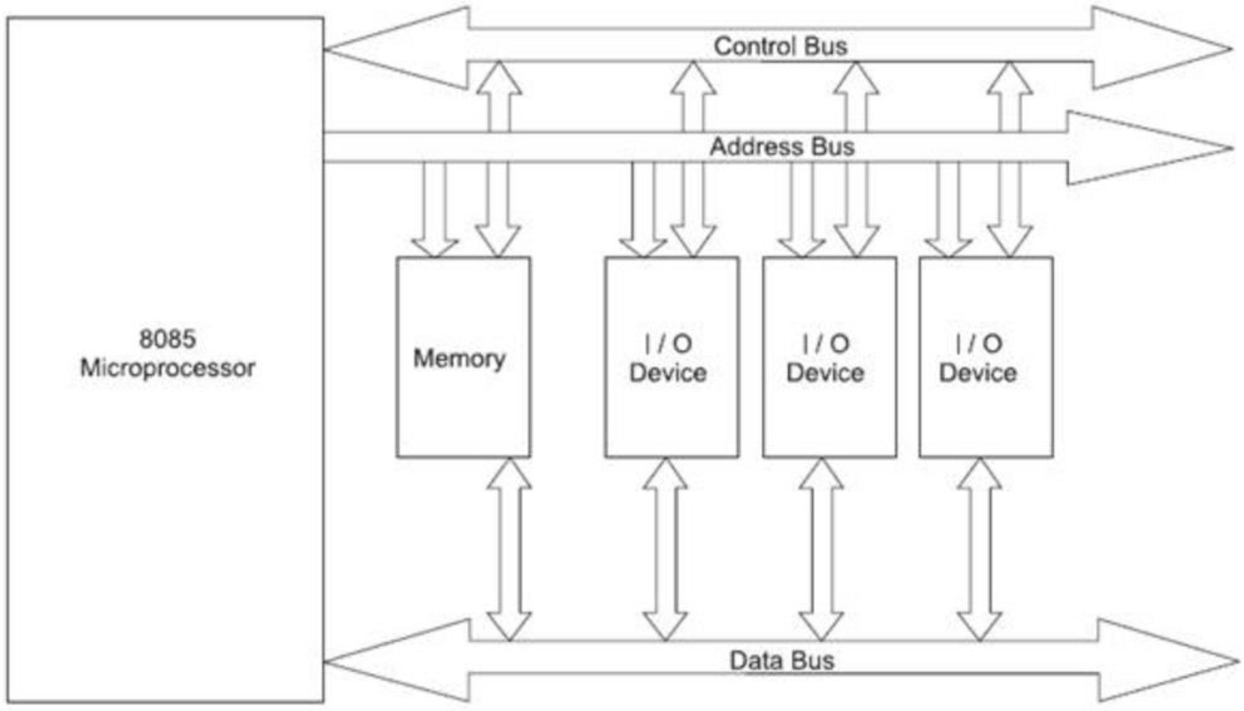
Static RAM	**Dynamic RAM**
Stored data is retained as long as power is ON.	Stored data will be erased and repeated refreshing is required to store data.
Stored data will not be changed with time.	Stored data will be changed with time.
Consume more power.	Consume less power than static RAM.
SRAM is expensive.	SRAM is less expensive.
SRAM has less packing density.	DRAM has high packing density.
These memories are not easy to construct.	These memories construction are simple.
No refreshing is required.	As refreshing is required, additional circuit is incorporated with memory.
No maintenance is required.	Maintenance is required.

Table : Comparison of memories

Types of Memory	**Category**	**Erasing property**	**Writing mechanism**	**Volatility**
Read only Memory (ROM)	Read only Memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)		Not possible	Electrically	Nonvolatile
Erasable PROM (EPROM)		Ultra-violet light and chip level	Electrically	Nonvolatile
Electrically Erasable PROM		Electrically and byte level	Electrically	Nonvolatile
(EPROM)				
Flash memory		Electrically and block level	Electrically	Nonvolatile
Random access memory (RAM)	Read and write memory	Electrically and byte level	Electrically	Volatile

4. Explain memory interfacing techniques used in 8085 microprocessor

Memory and I/O interfacing with microprocessor



Memory devices are used to store digital information. The simplest type of digital memory device is the flip-flop, which is capable for storing single bit data, and is volatile and very fast. This device is generally used to store data in the form of registers. Registers are also used as main memory of computers for internal computational operations. The basic goal of digital memory is to store and access binary data, which is a sequence of 1's and 0's.

Each memory chip like RAM, ROM, EPROM, E^2PROM and DRAM have numbers of pins and these pins are used to accept different kinds of signals. Normally every memory chip has pins for address, data, control signals and chip select signals.

Address Pins : Address pins are used to accept address from the system address bus transmitted by the microprocessor. 8085 has 16 bit address bus; hence it can access 216 no. of memory locations, which is equal to 64KB memory. For any microprocessor memory is required to store program as well as data. Since microprocessor doesn't have on-chip memory, we need to connect it externally. So it requires addressing mechanism.

CS (Chip Select) or CE (Chip Enable) Pin : This signal of the memory chip is ACTIVE LOW and acts as master enable pin for read or write operation. Hence, for every read or write operation, this signal must be low otherwise no operation will be performed.

WR (Write Control Signal) Pin :This is an active low input control signal used to write data to the memory location whose address is available on address lines if chip select signal is enable. This signal is available on system control bus and generated by the microprocessor or other master in the system such as DMA controller or co-processor.

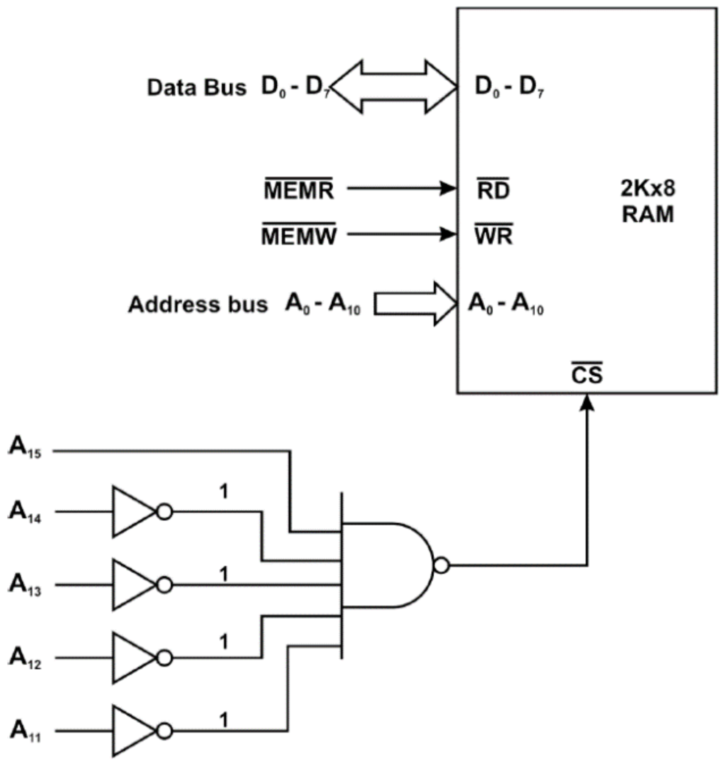
RD (Read Control Signal) Pin : This is an active low input control signal used to read data from the memory location whose address is available on address lines if chip select signal is enable. This signal is available on system control bus and generated by the microprocessor or other master in the system such as DMA controller or co-processor.

Chip Select Logic : Chip select logic can be developed using either combination of different gates such as AND, NAND, NOT etc. or decoders.

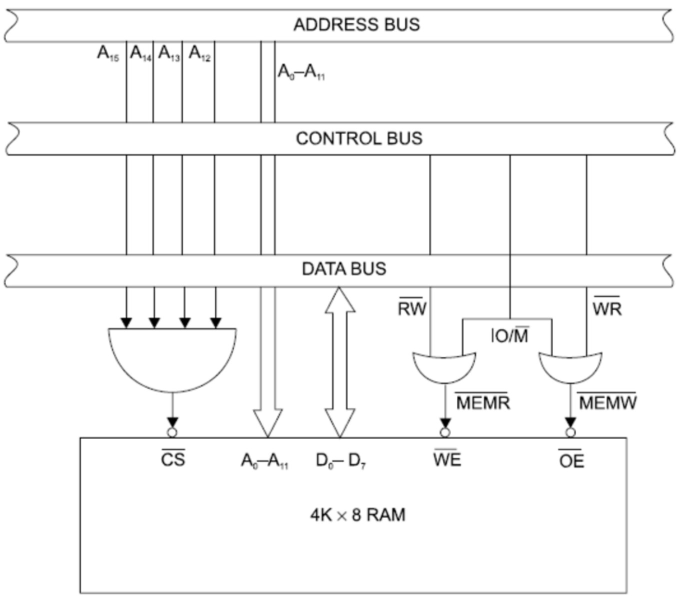
Using Logic Gates : Interfacing of 2K of RAM with the microprocessor 8085.

The 8085 is an 8 bit microprocessor. Hence all 8 lines of data bus can be directly connected after de-multiplexing to D0-D7 of the RAM memory. The eleven (11) address lines required to access any memory location within 2k memory, so out of 16 address lines (A0-A15) of 8085 microprocessor, the A0-A10 address lines can be connected directly to generate chip select signal using NAND and NOT gates depending on the addresses required.

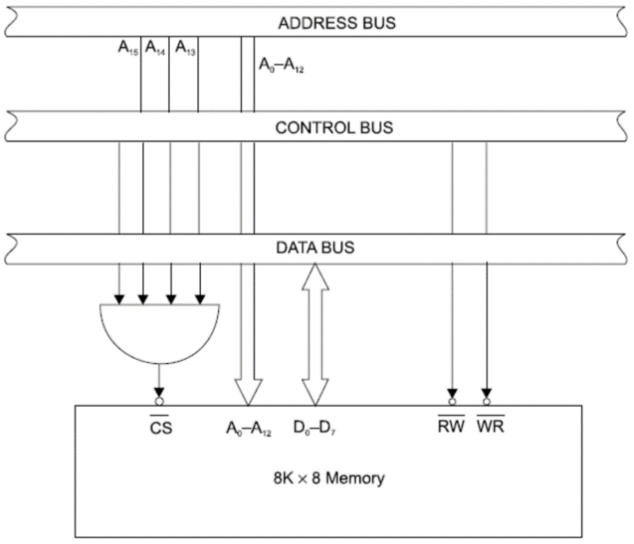
For generation of chip select we are using NAND gate. The chip select is active low signal, hence all inputs of the NAND gates must be logic 1 to generate chip select signal.



Interfacing 4K byte RAM with microprocessor

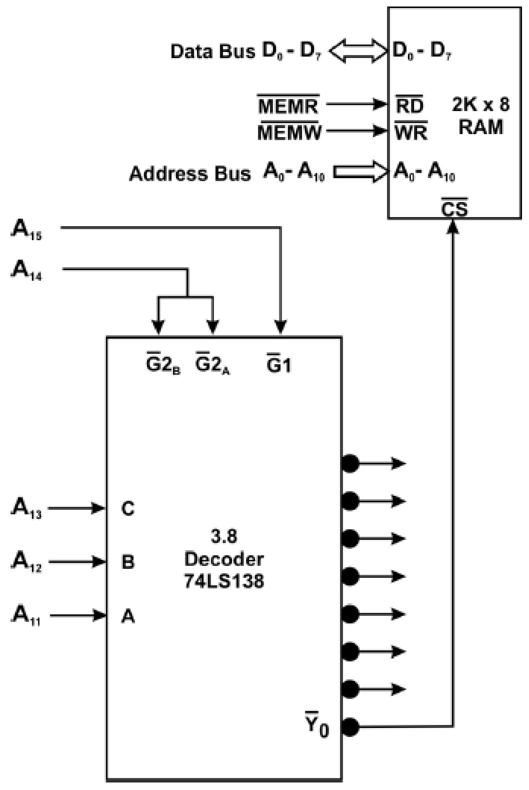


Interfacing 8K byte RAM with microprocessor



Using Decoder

Consider above example of memory interfacing, where remaining address lines i.e. A11-A15 can be connected to the decoder. Now connect A11, A12, A13 to A, B,C inputs of 3:8 decoder 74LS138 respectively, A14 to G2A and G2B enable pins and at last A15 to G1 pin of 3:8 decoder.



Here, the advantage of using decoder is minimum hardware is required as compared using NAND gates. When we use NAND gates, other logical devices are also required as per requirement as in above examples, NOT gates are used. Hence for numbers of devices, numbers of NAND and other logical devices are required to generate chip select signals.

But when we use decoder like 3:8 (74LS138), we can generate eight chip select signals using one decoder IC, as it has eight active low output pins.

The following are the steps 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 we 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.

5. Explain IO interfacing with 8085.

There are two method of interfacing I/O devices with the microprocessor , I/O mapped I/O, Memory mapped I/O.

I/O mapped I/O

In this technique, I/O device is treated as a I/O device and memory as memory.

Each I/Q device uses eight address lines.

If eight address lines are used to interface to generate the address of the I/O port, then 256 input and 256 output devices can be interfaced with the microprocessor.

The address bus of the 8085 microprocessor is 16 bit, so we can either use lower order address lines i.e. A0 – A7 or higher order address lines i.e. A8 – A15 to address I/O devices where the address available on A0 – A7 will be copied on the address lines A8 – A15.

In I/O mapped I/O, the complete 64 Kbytes of memory can be interfaced as all address lines can be used to address memory locations as the address space is not shared among I/O devices and memory and 256 input and /or output devices.

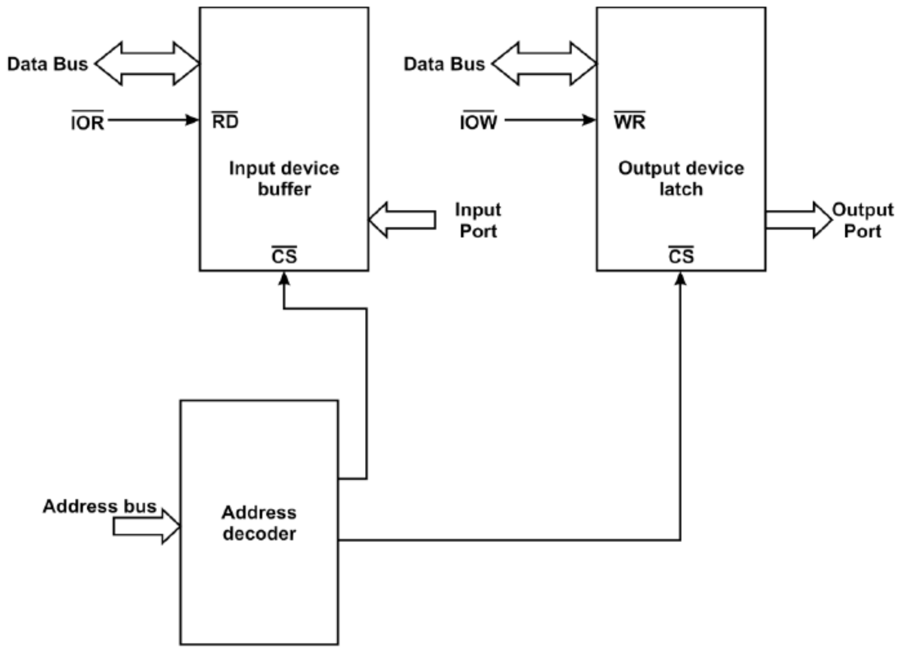
In this type, the data transfer is possible between accumulator A register and I/O devices only.

Address decoding is simple, as less hardware is required.

The separate control signals are used to access I/O devices and memory such as IOR, IOW for I/O port and MEMR, MEMW for memory hence memory location are protected from the I/O access.

But in this type, arithmetic and logical operation are not possible directly.

Also we cannot use other register for data transfer between I/O device and microprocessor accepts A register.



MEMORY MAPPED I/O

In this technique, I/O devices are treated as memory and memory as memory, hence the address of the I/O devices are as same as that of memory i.e. 16 bit for 8085 microprocessor.

The address space of the memory i.e. 64 Kbytes will be shared by the I/O devices as well as by memory.

All 16 address lines i.e. A0-A15 is used to address memory locations as well as I/O devices.

The control signals MEMR and MEMW are used to access memory devices as well as I/O devices.

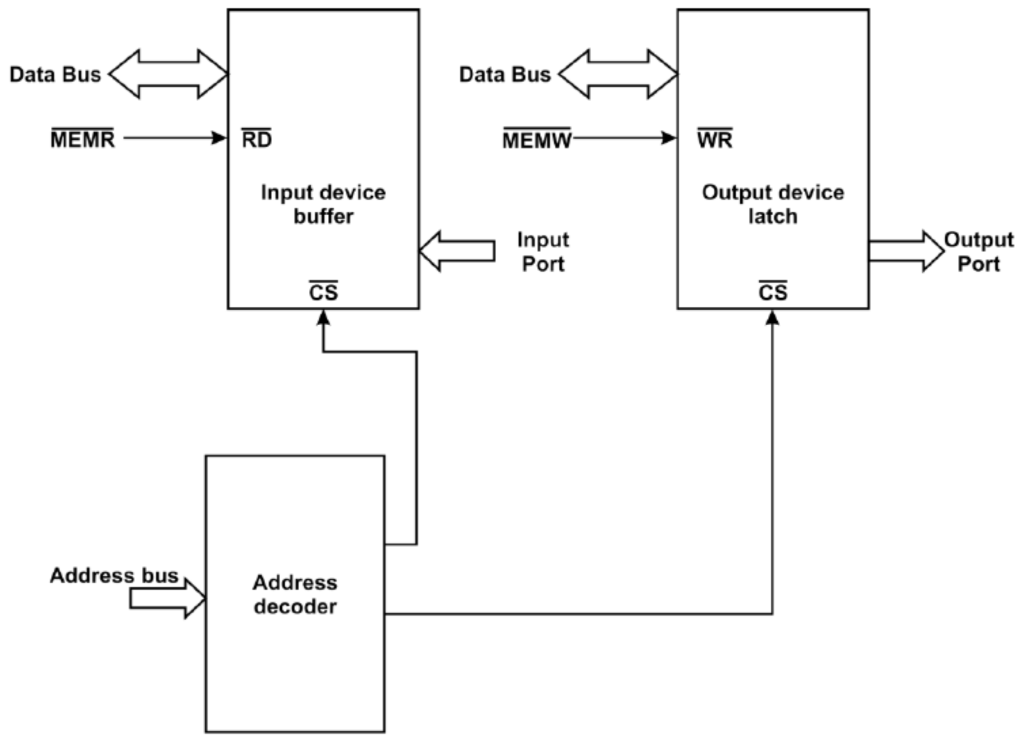
The data transfer is possible between any register of the microprocessor and I/O device or memory device.

Hence, all memory related instructions can be used to access devices as they are treated as memory devices.

Address decoding of the I/O devices and memory devices are complicated and expensive as more hardware is required.

The 8085 microprocessor can access either 64 K I/O ports or memory locations, hence the total numbers of the I/O ports and memory locations should not be greater than 64 K.

I/O devices and memory locations are distinguished by the addresses only.



I/O mapped I/O has the following advantages:

The total 256-address spaces are available for I/O devices.

Program writing will be easy as special instructions are used for I/O operations. In 8085 microprocessor, IN and OUT instructions are usually used for data transfer with I/O devices.

In I/O mapped I/O scheme, the I/O address length is usually one byte and instructions are two byte long. Therefore the program requires less memory and shorter execution time compared to memory mapped I/O.

The memory reference instructions can be easily distinguished from I/O reference instructions, which make program debugging easier.

I/O mapped I/O has the following disadvantages:

One microprocessor pin must be used to distinguish between memory and I/O operations. The additional control signals, IOR and IOW must be generated for read and write operations.

In data transfer with I/O devices and microprocessor, the data has to be transferred to the accumulator only to perform arithmetic or logical operations. Different addressing modes are not used in I/O mapped I/O.

Memory mapped I/O has the following advantages:

The Memory mapped I/O scheme can provide more than 256 input-output ports, as the port Addresses are 16-bit.

All the memory related instructions can be used in read and write operations of memory mapped I/O devices. The arithmetic and logical operations can be performed on available I/O data directly.

CPU registers can exchange transfer of data with I/O devices directly without accumulator.

Therefore memory mapped I/O simplifies and increases speed of data transfer.

Memory mapped I/O has the following disadvantages:

Memory mapped I/O scheme utilizes memory reference instructions, which are three byte instructions and is longer than I/O instructions.

Due to wider port address, the interface of hardware is also complicated.

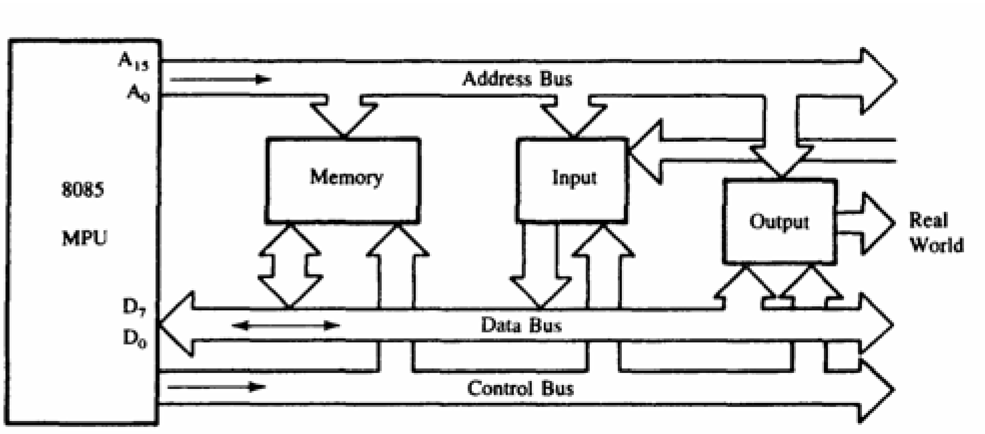
The complexity of the program is large.

Comparison between memory mapped I/O and I/O mapped I/O

Memory-Mapped I/O	**I/O Mapped I/O**
16-bit address	8-bit address
MEMR memory read, *MEMW* memory write	*IOR* I/O read, *IOW* I/O write
Memory related instructions are MOV M,R, MOV R,M, ADDM, ANA M, SUB M, STA, LDA, LDAX, STAX	I/O related instruction are IN and OUT
Data transfer between any register and I/O	Data transfer between accumulator and I/O
The memory map 64K is shared between I/Os and system memory	The I/O map is independent of the memory map; 256 input devices and 256 output devices
13 T states (STA, LDA)	The IN and OUT instructions are required
7 T states (MOV M,R)	10 T–states for execution
More hardware is needed to decode 16 bit address	Less hardware is needed to decode 8-bit address.
Arithmetic or logical operations can be directly performed with I/O data	Not available

5. Explain the bus structure of 8085 processor.

Typical system uses a number of buses i.e collections of wires, which transmit binary numbers, one bit per wire in term of voltage levels 0 volt or 5 volt for 0 and 1 respectively. The microprocessor MPU performs various operations with peripheral devices or a memory location by using three sets of communication lines called buses: the address bus, the data bus and the control bus. And these three combined lines is called as system bus.



Address bus: The address bus is a group of 16 lines generally called as A0 – A15 to carry a 16-bit address of memory location. In a computer system, each peripheral or memory location is identified by a binary number called an address. This is similar to the postal address of a house. The address bus is unidirectional, that means bit flow in only one direction from MPU to peripheral. MPU carries 16-bit address i.e. 216 = 65,536 or 64K memory locations.

Data Bus: The data bus is a group of eight bidirectional lines used for data flow in both the directions between MPH and peripheral devices. The 8 data lines are manipulating 8-bit data ranging from 00 to FF i.e. (28 = 256) numbers from 0000 0000 -1111 1111. This 8 bit data is called as word length and the register size of a microprocessor and MPH is called 8–bit microprocessor.

Control bus: Control bus is having various single lines used for sending control signals in the form of pulse to the memory and I/O devices. The MPU generates specific control signals to perform a particular operations. Some of these control signals are memory read, memory write, I/O read and I/O write.

6. Describe the different types of interrupts used in 8085 microprocessor.

Interrupt Structure in 8085

Interrupt is signals send by an external device to the processor, to request the processor to perform a particular task or work.
Mainly in the microprocessor based system the interrupts are used for data transfer between the peripheral and the microprocessor.
The processor will check the interrupts always at the 2nd T-state of last machine cycle.
If there is any interrupt it accept the interrupt and send the INTA (active low) signal to the peripheral.
The vectored address of particular interrupt is stored in program counter.
The processor executes an interrupt service routine (ISR) addressed in program counter.
It returned to main program by RET instruction.

Types of Interrupts:

It supports two types of interrupts.

Hardware

Software

Software interrupts:

The software interrupts are program instructions. These instructions are inserted at desired locations in a program. The 8085 has eight software interrupts from RST 0 to RST 7. The vector address for these interrupts can be calculated as follows.

Interrupt number * 8 = vector address

For RST 5,5 * 8 = 40 = 28H

Vector address for interrupt RST 5 is 0028H

The Table shows the vector addresses of all interrupts.

Interrupt	Vector Address
RST 0	0000_H
RST 1	0008_H
RST 2	0010_H
RST 3	0018_H
RST 4	0020_H
RST 5	0028_H
RST 6	0030_H
RST 7	0038_H
Hardware	Address
RST 7.5	$003C_H$
RST 6.5	0034_H
RST 5.5	$002C_H$
TRAP	0024_H

Hardware interrupts:

An external device initiates the hardware interrupts and placing an appropriate signal at the interrupt pin of the processor. If the interrupt is accepted then the processor executes an interrupt service routine. The 8085 has five hardware interrupts

(1) TRAP (2) RST 7.5 (3) RST 6.5 (4) RST 5.5 (5) INTR

<u>TRAP:</u>

This interrupt is a non-maskable interrupt. It is unaffected by any mask or interrupt enable.

TRAP bas the highest priority and vectored interrupt.

TRAP interrupt is edge and level triggered. This means hat the TRAP must go high and remain high until it is acknowledged.

In sudden power failure, it executes a ISR and send the data from main memory to backup memory.

The signal, which overrides the TRAP, is HOLD signal. (i.e., If the processor receives HOLD and TRAP at the same time then HOLD is recognized first and then TRAP is recognized).

There are two ways to clear TRAP interrupt.

By resetting microprocessor (External signal)
By giving a high TRAP ACKNOWLEDGE (Internal signal)

RST 7.5:

The RST 7.5 interrupt is a maskable interrupt. It has the second highest priority. It is edge sensitive. ie. Input goes to high and no need to maintain high state until it recognized.

Maskable interrupt. It is disabled by,

DI instruction
System or processor reset.
After reorganization of interrupt.
Enabled by EI instruction.

RST 6.5 and 5.5:

The RST 6.5 and RST 5.5 both are level triggered. . ie. Input goes to high and stay high until it recognized.

Maskable interrupt. It is disabled by,

DI, SIM instruction
System or processor reset.
After reorganization of interrupt.
Enabled by EI instruction.

The RST 6.5 has the third priority whereas RST 5.5 has the fourth priority.

INTR:

INTR is a maskable interrupt. It is disabled by,

DI, SIM instruction
System or processor reset.
After reorganization of interrupt.
Enabled by EI instruction.

Non- vectored interrupt. After receiving INTA (active low) signal, it has to supply the address of ISR.

It has lowest priority.

It is a level sensitive interrupts. i.e. Input goes to high and it is necessary to maintain high state until it recognized.

The following sequence of events occurs when INTR signal goes high.

1. The 8085 checks the status of INTR signal during execution of each instruction.

2. If INTR signal is high, then 8085 complete its current instruction and sends active low interrupt acknowledge signal, if the interrupt is enabled.

3. In response to the acknowledge signal, external logic places an instruction OPCODE on the data bus. In the case of multibyte instruction, additional interrupt acknowledge machine cycles are generated by the 8085 to transfer the additional bytes into the microprocessor.

4. On receiving the instruction, the 8085 save the address of next instruction on stack and execute received instruction.

SIM and RIM for interrupts:

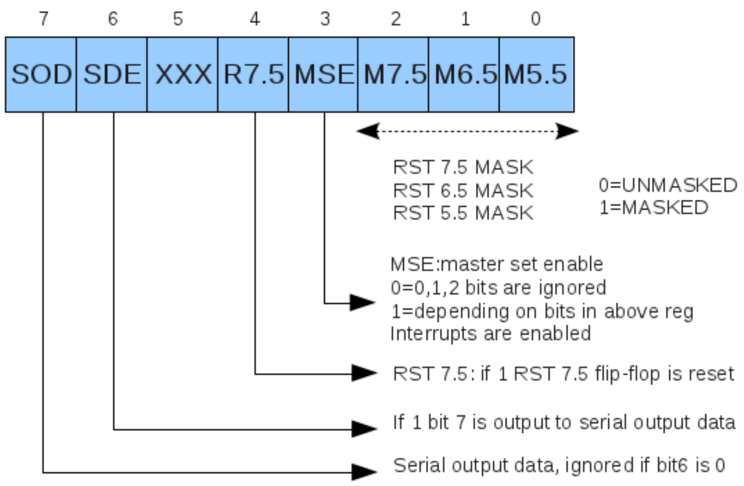
The 8085 provide additional masking facility for RST 7.5, RST 6.5 and RST 5.5 using

SIM instruction.

The status of these interrupts can be read by executing RIM instruction.

The masking or unmasking of RST 7.5, RST 6.5 and RST 5.5 interrupts can be performed by moving an 8-bit data to accumulator and then executing SIM instruction.

The format of the 8-bit data is shown below.


7 6 5 4 3 2 1 0
SOD SDE XXX R7.5 MSE M7.5 M6.5 M5.5
RST 7.5 MASK
RST 6.5 MASK
RST 5.5 MASK
0=UNMASKED
1=MASKED
MSE:master set enable
0=0,1,2 bits are ignored
1=depending on bits in above reg
Interrupts are enabled
RST 7.5: if 1 RST 7.5 flip-flop is reset
If 1 bit 7 is output to serial output data
Serial output data, ignored if bit6 is 0

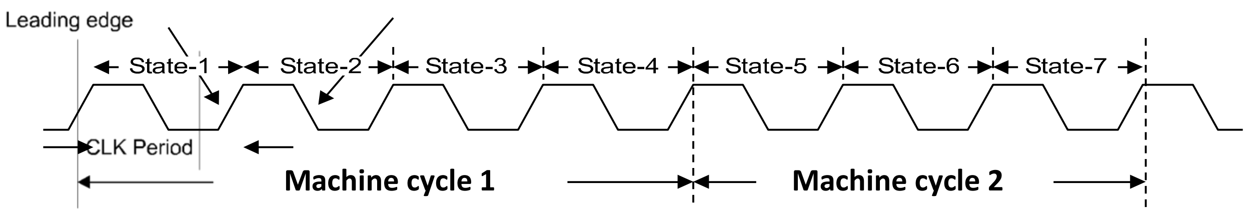
INTA is not an interrupt. INTA is used by the Microprocessor for sending the acknowledgement.

7. What is timing diagram, explain.

Timing diagram is the display of initiation of read/write and transfer of data operations under the control of 3-status signals IO / M , S1, and S0. 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. If I ask a man to bring 6-bags of wheat, each weighing 100 kg, he may take 6-times to perform this task in going and bringing it. A stronger man might perform the same task in 3- times only. Thus, it depends on the strength of the man to finish the job quickly or slowly. Here, we can assume both weaker and strong men as machine. The weaker man has taken 6-machine cycle (6-times going and coming with one bag each time) to execute the job where as the stronger man has taken only 3-machine cycle for the same job. Similarly, a machine may execute one instruction in as many as 3-machine cycles while the other machine can take only one machine cycle to execute the same instruction. Thus, the machine that has taken only one machine cycle is efficient than the one taking 3-machine cycle. Each machine cycle is composed of many clock cycle. Since, the data and instructions, both are stored in the memory, the µP performs fetch operation to read the instruction or data and then execute the instruction. The µP in doing so may take several cycles to perform fetch and execute operation. The 3-status signals : IO / M , S1, and S0 are generated at the beginning of each machine cycle. The unique combination of these 3-status signals identify read or write operation and remain valid for the duration of the cycle. Table shows details of the unique combination of these status signals to identify different machine cycles.

	Status			*Controls*		
Machine cycle	IO / M	S_1	S_0	RD	WR	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
Acknowledge of INTR (INTA)	1	1	1	1	1	0
BUS Idle (BI) : DAD	0	1	0	1	1	1
ACK of RST, TRAP	1	1	1	1	1	1
HALT	Z	0	0	Z	Z	1
HOLD	Z	X	X	Z	Z	1
X ® Unspecified, and Z ® High impedance state						

Thus, time taken by any µP to execute one instruction is calculated in terms of the clock period. The execution of instruction always requires read and writes operations to transfer data to or from the µP and memory or I/O devices. Each read/ write operation constitutes one machine cycle (MC1) as indicated in Fig. 5.1 (a). Each machine cycle consists of many clock periods/ cycles, called T-states. The heartbeat of the microprocessor is the clock period. Each and every operation inside the microprocessor is under the control of the clock cycle. The clock signal determines the time taken by the microprocessor to execute any instruction. The clock cycle shown in Fig.has two edges (leading and trailing or lagging). State is defined as the time interval between 2-trailing or leading edges of the clock. Machine cycle is the time required to transfer data to or from memory or I/O devices.

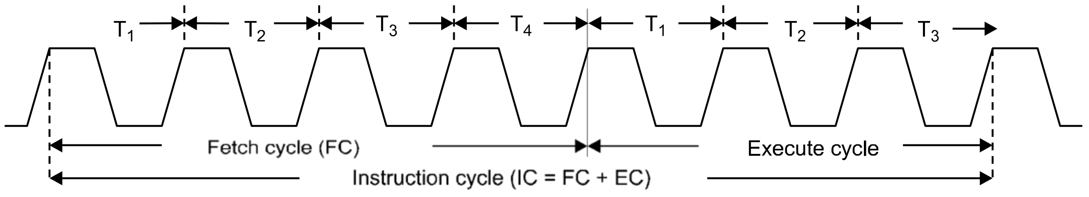


PROCESSOR CYCLE

The function of the microprocessor is divided into fetch and execute cycle of any instruction of a program. The program is nothing but number of instructions stored in the memory in se- quence. In the normal process of operation, the microprocessor fetches (receives or reads) and executes one instruction at a time in the sequence until it executes the halt (HLT) instruction. Thus, an instruction cycle is defined as the time required to fetch and execute an instruction. For executing any program, basically 2-steps are followed sequentially with the help of clocks. Fetch, and Execute.

The time taken by the µP in performing the fetch and execute operations are called fetch and execute cycle. Thus, sum of the fetch and execute cycle is called the instruction cycle as indicated in Fig.

Instruction Cycle (IC) = Fetch cycle (FC) + Execute Cycle (EC)



Each read or writes operation constitutes a machine cycle. The instructions of 8085 require 1–5 machine cycles containing 3–6 states (clocks). The 1st machine cycle of any instruction is always an Op. Code fetch cycle in which the processor decides the nature of instruction. It is of at least 4-states. It may go up to 6-states.

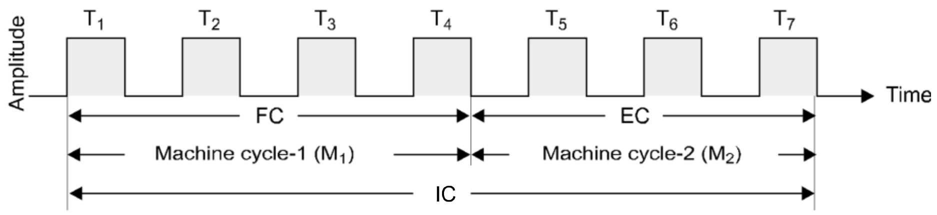


Fig. Ideal wave shape relationship for FC, EC, MC, and IC.

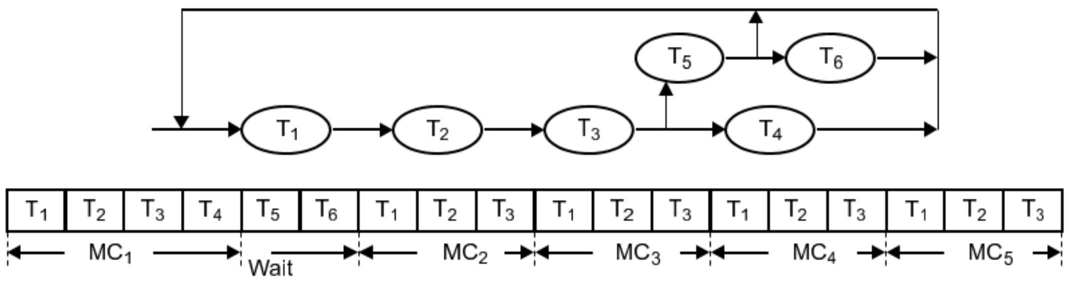
It is well known that an instruction cycle consists of many machine cycles. Each machine cycle consists of many clock periods or cycles, called T-states. The 1st machine cycle (M1) of every instruction cycle is the opcode fetch cycle. In the opcode fetch cycle, the processor comes to know the nature of the instruction to be executed. The processor during (M1 cycle) puts the program counter contents on the address bus and reads the opcode of the instruction through read process. The T1, T2, and T3 clock cycles are used for the basic memory read operation and the T4 clock and beyond are used for its interpretation of the opcode. Based on these interpretations, the µP comes to know the type of additional information/data needed for the execution of the instruction and accordingly proceeds further for 1 or 2-machine cycle of memory read and writes.

The Op. code fetch cycle is of fixed duration (normally 4-states), whereas the instruction cycle is of variable duration depending on the length of the instruction. As an example, STA instruction, requires opcode fetch cycle, lower-order address fetch cycle and higher order fetch cycle and then the execute cycle. Thus opcode fetch cycle is of one machine cycle in this example. A particular microprocessor requires a definite time to performing a specific task. This time is called machine cycle. Thus, one machine cycle is required each time the µP access I/O port or memory. A fetch opcode cycle is always 1-machine cycle, whereas, execute cycle may be of one or more machine cycle depending upon the length of the instruction.

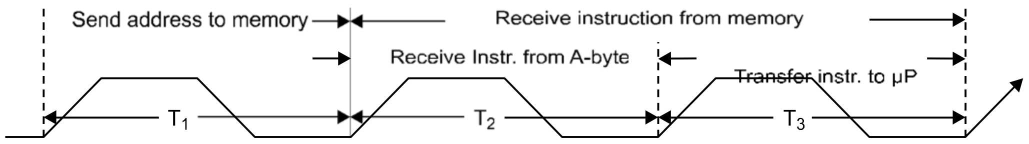
Instruction Fetch (FC) An instruction of 1 or 2 or 3-bytes is extracted from the memory locations during the fetch and stored in the µP's instruction register.

Instruction Execute (EC) The instruction is decoded and translated into specific activi- ties during the execution phase. Thus, in an instruction cycle, instruction fetch, and instruction execute cycles are related. Every instruction cycle consists of 1, 2, 3, 4 or 5-machine cycles as indicated in Fig. One machine cycle is required each time the µP access memory or I/O port. The fetch cycle, in general could be 4 to 6-states whereas the execute cycle could of 3 to 6-states. The 1st machine cycle of any instruction is always the fetch cycle that provides identification of the instruction to be executed.

The fetch portion of an instruction cycle requires one machine cycle for each byte of instruction to be fetched. Since instruction is of 1 to 3 bytes long, the instruction fetch is one to 3-machine cycles in duration. The 1st machine cycle in an instruction cycle is always an opcode fetch. The 8-bits obtained during an opcode fetch are always interpreted as the Opcode of an instruction. The machine cycle including wait states is shown in Fig.



Some instructions do not require any machine cycle other than that necessary to fetch the instruction. Other instructions, however, require additional machine cycles to write or read data to or from memory or I/O devices.



A typical fetch cycle is explained in Fig. only two clock cycles have been shown as the requirement to read the instruction. Since the access time of the memory may vary and it may require more than 2-clock cycles, the microprocessor has to wait for more than 2-clocks duration before it receives the opcode instruction. Hence, most of the microprocessors have the provisions of introducing wait cycle within the fetch cycle to cope up with the slow memories or I/O devices.

Opcode Fetch

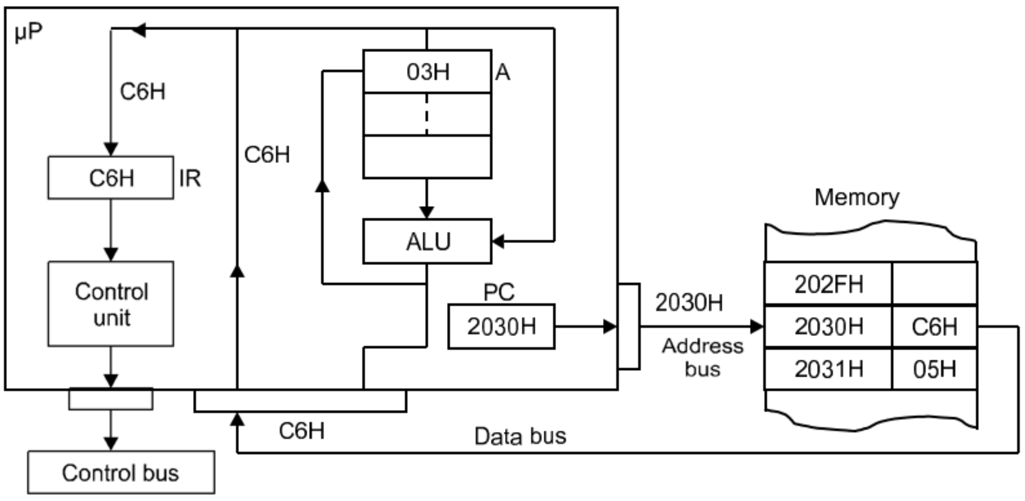
A microprocessor either reads or writes to the memory or I/O devices. The time taken to read or write for any instruction must be known in terms of the μP clock. The 1st step in communicating between the microprocessor and memory is reading from the memory. This read- ing process is called opcode fetch. The process of opcode fetch operation requires minimum 4- clock cycles T1, T2, T3, and T4 and is the 1st machine cycle (M1) of every instruction.

In order to differentiate between the data byte pertaining to an opcode or an address, the machine cycle takes help of the status signal IO / M , S1, and S0. The IO / M = 0 indicates memory operation and S1 = S0 = 1 indicates Opcode fetch operation.

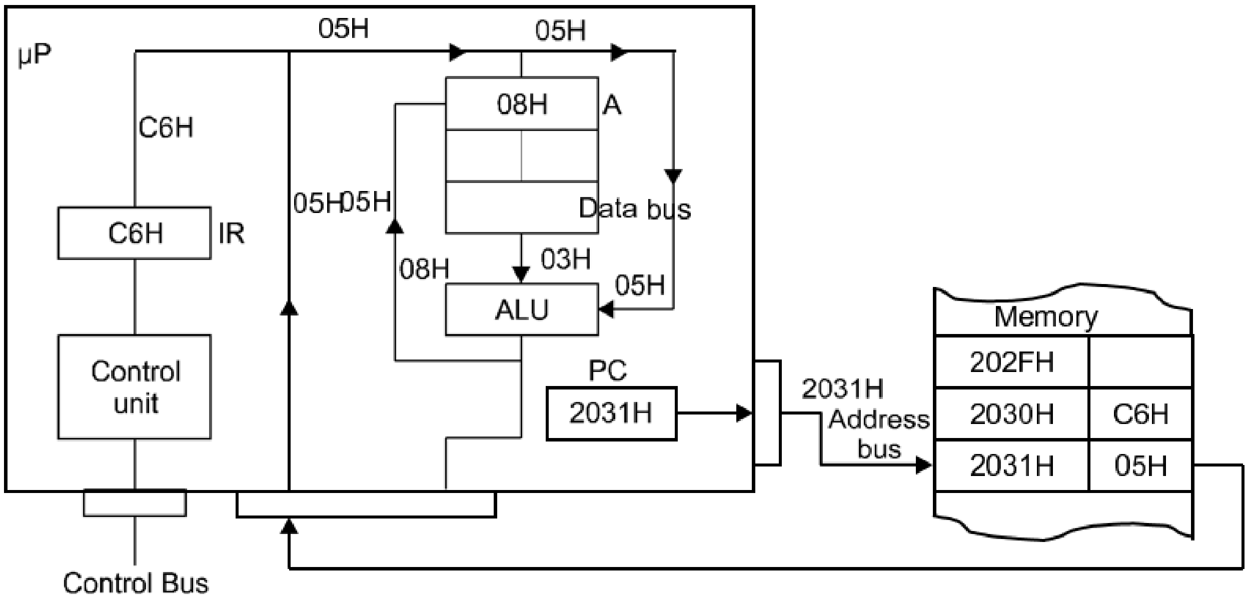
The opcode fetch machine cycle M1 consists of 4-states (T1, T2, T3, and T4). The 1st 3- states are used for fetching (transferring) the byte from the memory and the 4th-state is used to decode it.

Thus, thorough understanding about the communication between memory and microprocessor can be achieved only after knowing the processes involved in reading or writing into the memory by the microprocessor and time taken w.r.t. its clock period.

The process of implementation of each instruction follows the fetch and execute cycles. In other words, first the instruction is fetched from memory and then executed. Figs.depict these 2-steps for implementation of the instruction ADI 05H. Let us assume that the accumulator contains the result of previous operation i.e., 03H and instruction is held at memory locations 2030H and 2031H.



The fetch part of the instruction is the same for every instruction. The control unit puts the contents of the program counter (PC) 2030H on the address bus. The 1st byte (opcode C6H in this example) is passed to the instruction register. In the execute cycle of the instruction, the control unit examines the opcode and as per interpretation further memory read or write operations are performed depending upon whether additional information/ data are required or not. In this case, the data 05H from the memory is transferred through the data bus to the ALU. At the same time the control unit sends the contents of the accumulator (03H) to the ALU and performs the addition operation. The result of the addition operation 08H is passed to the accumulator overriding the previous contents 03H. On the completion of one instruction, the program counter is automatically incremented to point to the next memory location to execute the subsequent instruction.



8. Draw the timing diagram of opcode fetch machine cycle & I/O read cycle.

1. Instruction cycle: this term is defined as the number of steps required by the cpu to complete the entire process ie. Fetching and execution of one instruction. The fetch and execute cycles are carried out in synchronization with the clock.

2. Machine cycle: It is the time required by the microprocessor to complete the operation of accessing the memory devices or I/O devices. In machine cycle various operations like opcode fetch, memory read, memory write, I/O read, I/O write are performed.

3. T-state: Each clock cycle is called as T-states.

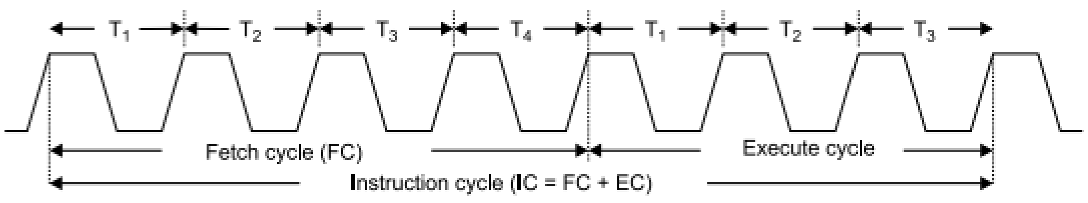
<u>Rules to identify number of machine cycles in an instruction:</u>

1. If an addressing mode is direct, immediate or implicit then No. of machine cycles = No. of bytes.

2. If the addressing mode is indirect then No. of machine cycles = No. of bytes + 1. Add +1 to the No. of machine cycles if it is memory read/write operation.

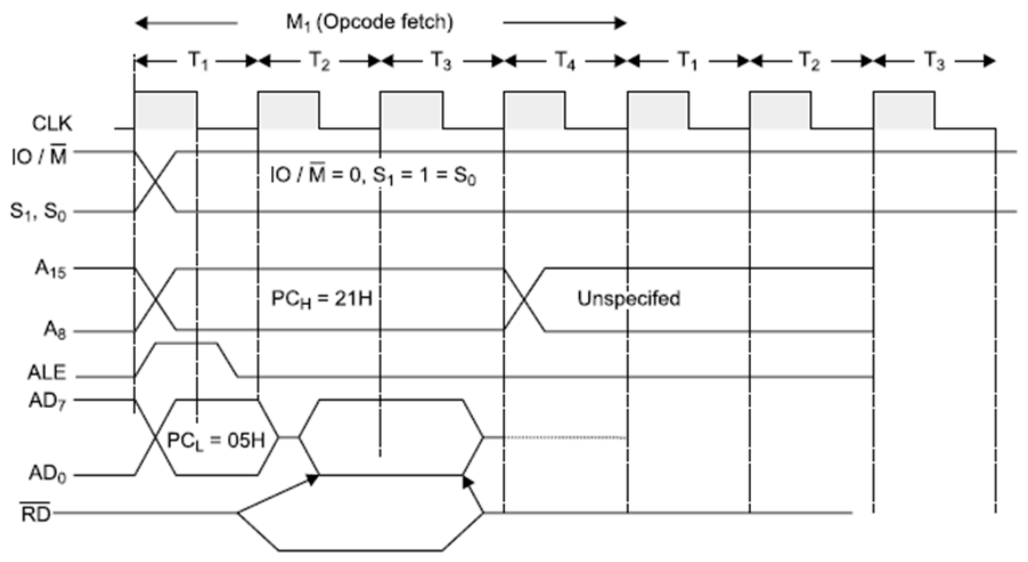
3. If the operand is 8-bit or 16-bit address then, No. of machine cycles = No. of bytes +1.

4. These rules are applicable to 80% of the instructions of 8085.



Where, Instruction cycle= Fetch Cycle(FC) + Execute cycle(EC).

The microprocessor requires instructions to perform any particular action. In order to perform these actions microprocessor utilizes Opcode which is a part of an instruction which provides detail(ie. which operation µp needs to perform) to microprocessor.



During T1 state, microprocessor uses IO/M(bar), S0, S1 signals are used to instruct microprocessor to fetch opcode.

Thus when IO/M(bar)=0, S0=S1= 1, it indicates opcode fetch operation.

During this operation 8085 transmits 16-bit address and also uses ALE signal for address latching.

At T2 state microprocessor uses read signal and make data ready from that memory location to read opcode from memory and at the same time program counter increments by 1 and points next instruction to be fetched.

In this state microprocessor also checks READY input signal, if this pin is at low logic level ie. '0' then microprocessor adds wait state immediately between T2 and T3.

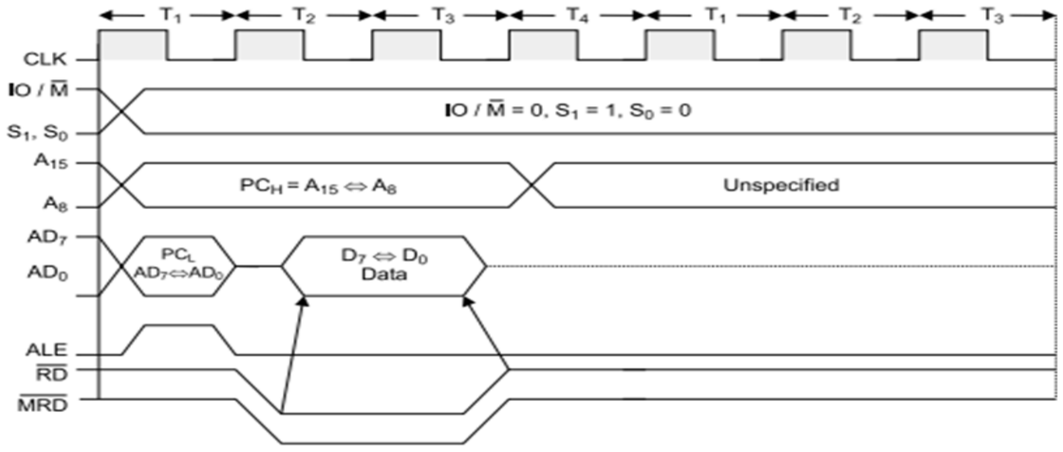
At T3, microprocessor reads opcode and store it into instruction register to decode it further.

During T4 microprocessor performs internal operation like decoding opcode and providing necessary actions.

The opcode is decoded to know whether T5 or T6 states are required, if they are not required then µp performs next operation.

Read and write timing diagram for memory and I/O Operation

Memory Read:



Operation:

It is used to fetch one byte from the memory.

It requires 3 T-States.

It can be used to fetch operand or data from the memory.

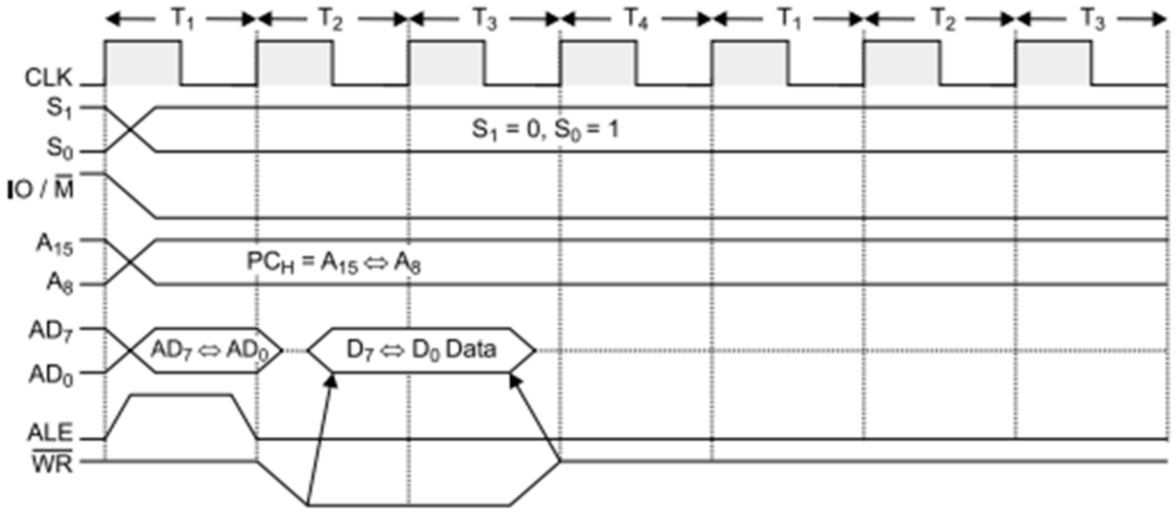
During T1, A8-A15 contains higher byte of address. At the same time ALE is high. Therefore Lower byte of address A0-A7 is selected from AD0-AD7.

Since it is memory ready operation, IO/M(bar) goes low.

During T2 ALE goes low, RD(bar) goes low. Address is removed from AD0-AD7 and data D0-D7 appears on AD0-AD7.

During T3, Data remains on AD0-AD7 till RD(bar) is at low signal.

Memory Write:



Operation:

It is used to send one byte into memory.

It requires 3 T-States.

During T1, ALE is high and contains lower address A0-A7 from AD0-AD7.

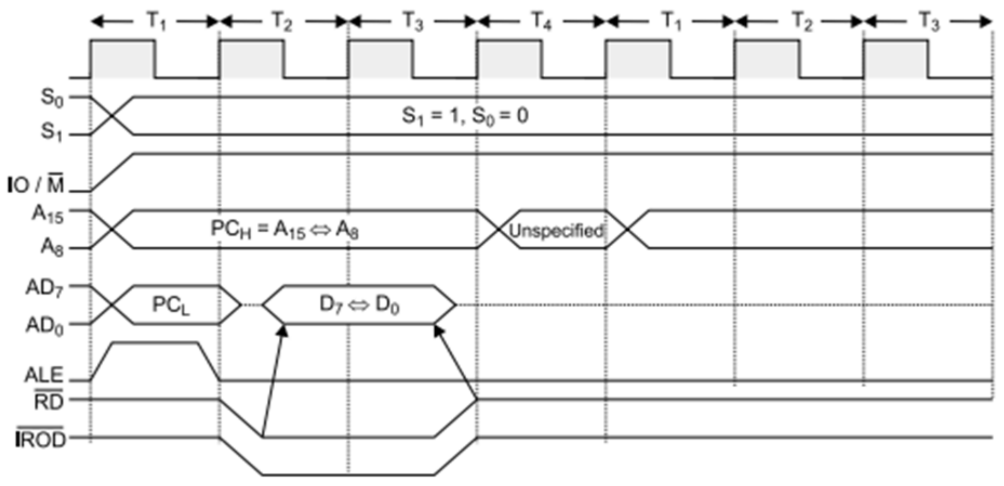
A8-A15 contains higher byte of address.

As it is memory operation, IO/M(bar) goes low.

During T2, ALE goes low, WR(bar) goes low and Address is removed from AD0-AD7 and then data appears on AD0-AD7.

Data remains on AD0-AD7 till WR(bar) is low.

IO Read:



Operation:

It is used to fetch one byte from an IO port.

It requires 3 T-States.

During T1, The Lower Byte of IO address is duplicated into higher order address bus A8-A15.

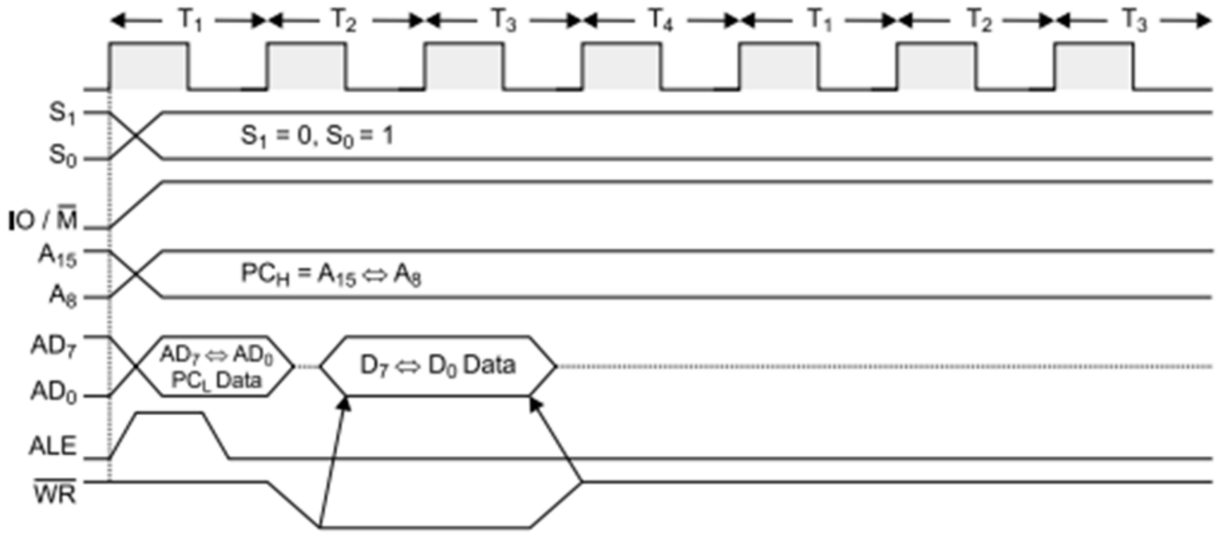
ALE is high and AD0-AD7 contains address of IO device.

IO/M (bar) goes high as it is an IO operation.

During T2, ALE goes low, RD (bar) goes low and data appears on AD0-AD7 as input from IO device.

During T3 Data remains on AD0-AD7 till RD(bar) is low.

IO Write:



Operation:

It is used to writ one byte into IO device.

It requires 3 T-States.

During T1, the lower byte of address is duplicated into higher order address bus A8-A15.

ALE is high and A0-A7 address is selected from AD0-AD7.

As it is an IO operation IO/M (bar) goes low.

During T2, ALE goes low, WR (bar) goes low and data appears on AD0-AD7 to write data into IO device.

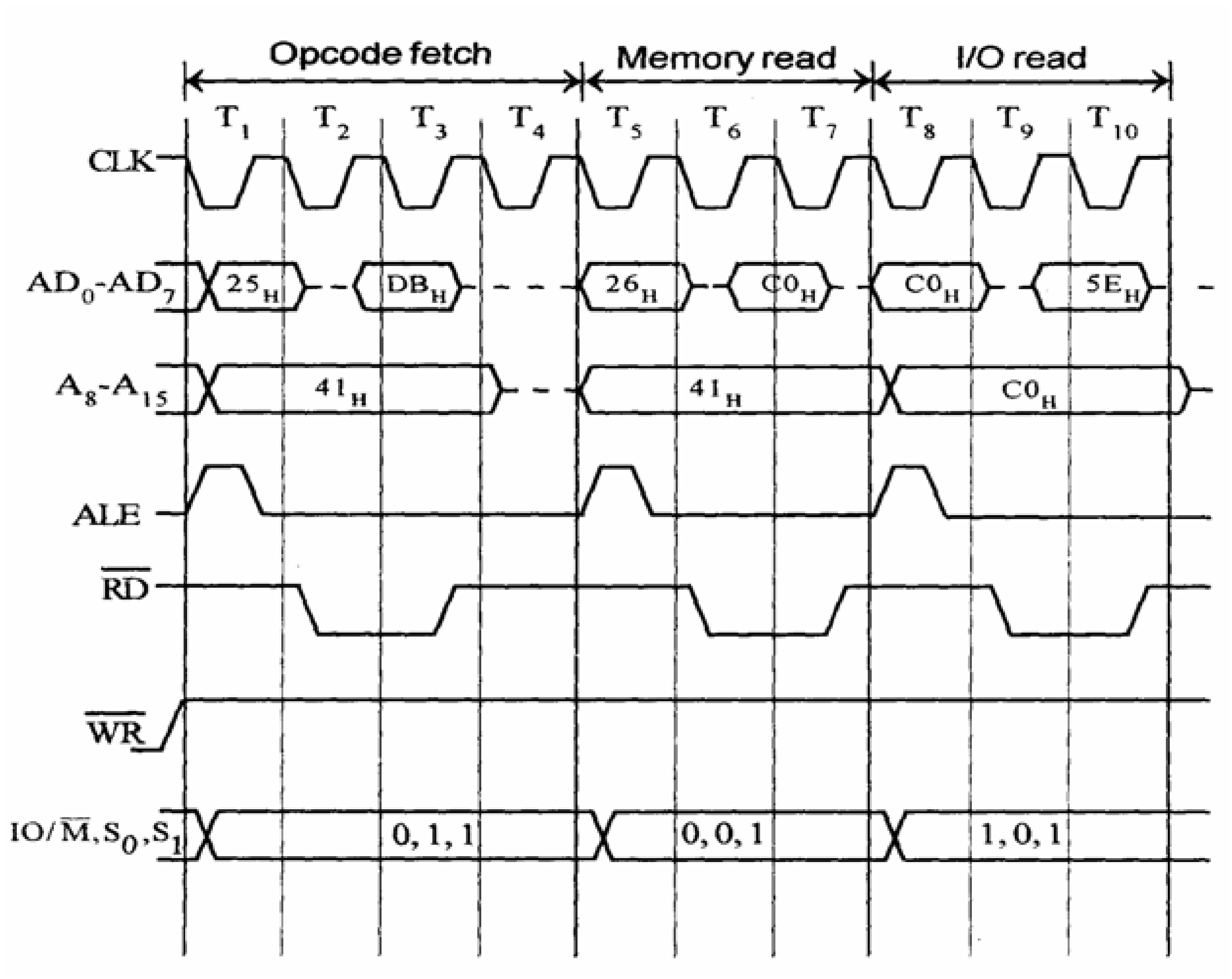
During T3, Data remains on AD0-AD7 till WR(bar) is low.

9. Draw the timing diagram of IN & OUT instruction of 8085 & explain.

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



Opcode fetch
Memory read
I/O read
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10
CLK
AD0-AD7
25H
DBH
26H
C0H
C0H
5EH
A8-A15
41H
41H
C0H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
1, 0, 1