

They process information 8 bit at a time. That's why they are called 8 bit processor. They can handle large numbers but in order to process large numbers, they broke them into 8 bit pieces and process each bit of group separately.

Memory - It is the location where information is kept while not in current use. It is stored in memory. Memory is collection of storage device. Usually each storage device holds 1 bit, also, in most kind of memory, these storage devices are grouped into group of eight. These 8 storage locations can only be accessed together so one can read only or write in terms of byte to and from memory.

Assignment 1 - Q. Explain microcomputer, microcontroller and microprocessor. And write three examples of each.

31/1/24 : Master-Slave System

In this system, the microprocessor is the master and all peripheral devices are slave objects.

The master control and initiate all operations of peripheral devices. The buses are group of lines that carry data, address, control signals. The CPU interface is provided to a multiplex line to generate the chip select signals and additional control signals. The system bus has separate bus lines for each signal. All the slaves in the system are connected to the same system bus. At any time instant communication takes place between the master and one of the slaves. All the slave have tri state logic and hence normally remain in high impedance

state. The processor select a slave by sending an address. When a slave is selected, it comes to the normal logic, and communicate with the processor.

and data

The EPROM memory is used to store program permanently. The RAM memory is used to store temporary program and data. The input device is used to enter CPU program, data and to operate all system. The output device is used for examining the result. Since the speed of input output devices does not match with speed of microprocessors. An interface device is provided between system bus and input output devices.

CPU - CPU consists of ALU, CU, and MV (Registers Unit)

The CPU stores instruction and data word from memory. It also exports process data in memory.

2/120/1
7 5 6
2 3 0 0
2 1 5 1
2 7 1
2 3 1
1

ALU - This section perform computing functions on data. These functions are arithmetic operations such as +, - and so on, and logical operations, AND, OR, NAND, NOR etc. Results are stored either in memory or register or send to output devices.

MV - It contains various registers. The registers are used primarily to store data temporarily during execution of program. Some of the registers are accessible ~~to~~ to the user through instructions.

CU - It provide necessary timing and control signals. Necessary to all operations in the micro computer. It controls flow of data between the microprocessor and peripherals (I/O and memory). The CU gets a clock which determines the speed of the microprocessor.

2/2/2⁴ Machine language -

It is a low level programming language. It can be only represented by 0 and 1. In earlier when we have to create a picture or show data on the screen of the computer then it is very difficult to draw using only binary digit (0 and 1) For e.g. - To write 120 in the computer system, it represent as 1111000. So, it is very difficult to learn. To overcome this problem, the assembly language is introduced

Assembly language - It is more than low-level and less than high level. So, it is intermediary language. It uses numbers, symbols and abbreviations instead of 0 and 1. For e.g. - Addition, subtraction and multiplication, it uses symbols like ADD, SUB, MUL.

Q. What is the difference between Assembly, Machine and High level language on behalf of microprocessor.

Assembly language Program Development Tools -

1. Editor - It is a program which allows you to create a file containing the assembly language statement for your program. Eg. - PC Write and Word Star. When you type your program, the editor stores the ASCII code for the letters and numbers in successive RAM location. When you have typed in all your program, then you save the file on the hard-disc. This file is called source file and the extension is .asm
2. Assembler - It is used to translate assembly language mnemonics for instructions to corresponding binary codes. When you run the assembler it reads the source file of your program from the disc where you have saved it after editing.
3. Linker - A linker is a program used to join several object files into one large object file. It produces a link file which contains binary code for all combined modules. The linker also produces a link map file which contains the address information about the link file (.exe)
4. Locator - A program used to assign the specific address of object code where the segment of the object code is to be loaded into the memory.
5. Debugger - Program which allows you to load your object code program into system memory, execute the program and troubleshoot or debug it. It allows you to change the content of registers and memory locations and re-run program.
6. Emulator - Mixture of hardware and software. Used to test and debug hardware and software of an external system such as prototype of a microprocessor.

Temporary Register:

The register act as a temporary memory during the arithmetic and logical operation. Unlike other register this T.R. can only be accessed by the microprocessor and it is completely inaccessible to programmable memory. It is ~~is~~ an 8-bit register.

Flags - They are nothing but a group of individual flip-flops. The flags are mainly associated with arithmetic and logic operations. The flag will show either 1 or 0 (0 or 1) 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. Inter processor have a set of five flags:

Carry Flag, Sign Flag, Auxiliary Carry Flag, Parity Flag, Zero Flag.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S	Z	AC		P		ACY	

0001
MSB LSB

When we add two nos., a carry is generated in the MSB. The extremely right no. is least significant bit while the number on extremely left is most significant bit. For the purpose of carry flag is used to set whenever a carry is generated and rest if there is no carry is generated. When there is no carry the auxiliary carry flag is reset. So whenever a carry is in MSB carry flag is set. While an A.C.F. is set only when a carry is generated in bits other than the MSB.

Parity Flag - This flag is used while the data transmission is carried out. Parity check is checked whether it is even or odd parity. This flag is written a 1 if it is even parity and written a 0 if it is odd 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 output of operation is 0 or not. If the value of 0 flag is 0 then the result of operation is not zero. If it is zero, the flag written value 1.

Sign Flag - Shows whether the output of operation has + sign or - sign.
A value 0 is written for + sign and 1 is written for - sign.

Instruction Registers and Decoder - I.R. is 8 bit register just like every other register of micro processor. Consider an instruction, the instruction may be anything like adding two data, moving a data, copying data. When such an instruction is fetched from memory, it is directed to instruction register. So, the I.R. are specifically to store the instruction that are fetched from memory. There is an instruction decoder which decodes the information present in I.R. for further processing.

Timing and Control Unit - It is a very important unit as it synchronize the register and flow of data through various registers and other units. This unit consists of an auxilitor and controller, sequences which sends control signals needed for internal and external control of data and other unit. The auxilitor generates two phase lock signals which aids in synchronizing all the registers of 8085. Signals that are associated with timing and control unit are RD, WR, ALE. ALE is used for providing control signals to synchronize the component of ~~microprocessor~~ and timing for instruction to perform operation. RD (Active mode) and WR (Active High) are used to indicate whether the operation is reading data from memory or writing data into memory respectively.

7/2/24

Status signals - I/O machine is used to indicate whether the operation belongs to the memory or peripherals. It is a status signal which determines whether the address is from I/O or memory. (I/O/m). When it is high (1) the address on the address bus is for I/O device. When it is low (0) the address on address bus is for the memory.

PMA Signal - HOLD - Indicates that another master is requesting the use of address and data bus. The CPU, upon receiving the hold request, will relinquish the use of bus as soon as the completion of current bus transfer. Internal processing can continue, the processor can regain the bus only after the hold.

is removed. When the hold is acknowledged, the address, the data RD, WR and I/O/M lines, ~~are~~ are dry stated.

(ii) HLDA - It is a signal which indicates that the hold request has been received after removal of hold request, the HLDA goes low. Hold acknowledge indicates that the CPU has received the hold stack and that it will relinquish the bus in next clock cycles. HLDA goes low after the hold request is removed. The CPU takes the buses on half clock cycle after HLDA goes low, fast.

(iii) Ready - This signal synchronize the ~~fast~~ CPU and the slow memory, peripherals. If ready is high during read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If ready is low, the CPU will wait and integral no. of clock cycle for ready to go high before completing the read/write cycle, ready must confirm to specify set up and hold times.

Reset signals - (i) Reset - When the signal on this pin is low (0), the program counter is set to 0, the buses are dry states and the microprocessor unit is reset. (ii) Reset Out - This signal indicates that the microprocessor is being reset, the signal can be used to reset other devices.

Interrupt Control - It interrupts process. Consider that a microprocessor is executing the main program whenever the interrupt signal is enable 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 e.g. an I/O device may sent an interrupt signal to notify that the data is ready for input. The microprocessor temporarily stops the execution of main program and transfer control to I/O device. After collecting input data, the control transferred back to main program. Interrupt signals present in 8085 are - INTR, TRAP, RST 7.5, RST 6.5, RST 5.5.

9/2/24

TRAP - The trap interrupt is a non-maskable interrupt, that is generated by an external device, such as power failure or a hardware malfunction. The trap interrupt has the highest priority and cannot be disabled.

RST 7.5 - The RST 7.5 interrupt is a maskable interrupt that is generated by a software instruction, it has the highest priority.

RST 6.5 - It is maskable interrupt that is generated by software instruction. It has third highest priority.

RST 5.5 - The RST 5.5 interrupt is a maskable interrupt that is generated by a software instruction. It has fourth highest priority.

INTR - The INTR interrupt is maskable interrupt that is generated by an external device, such as keyboard or a mouse. It has lowest priority, and can be disabled.

Assignment Ques - What is the difference between maskable and non-maskable interrupt? What is the difference between hardware and software interrupt and vector and non-vector interrupt?

Address Buffer and address Data Buffer - The content 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 bus and address data bus. As a memory and input / output chips are connected to these buses, and the CPU can exchange the 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.

D₀-D₇, A₀-A₇

Pin Diagram of 8085 - : The signals can be grouped as follows:

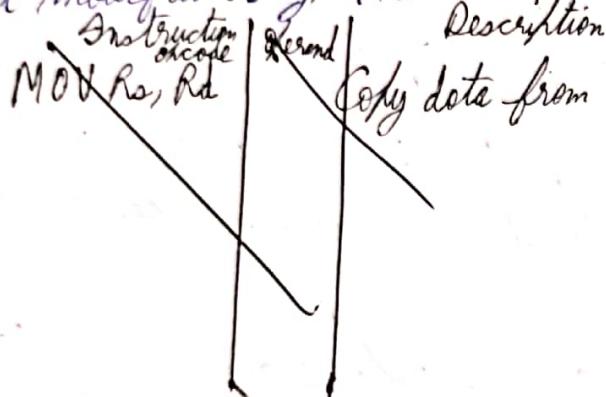
1. Power supply and clock signals
2. Address bus
3. Data bus
4. Control and status signals

5. Interrupt and externally initiated signals,
6. Senior I/O output ports

12/2/24

Instruction Sets — The 8085 instruction set can be classified into the following five functional headings.

Data Transfer Instruction — It includes that MOV data between registers or memory location and registers. In all data transfer operation the content of source register is not altered. Hence, the data transfer is copy operation. These instruction move data between register or M/L and registers. Copy data from source to destination. By copying, the content of source are not modified. Eg. $MOV\ R_1, R_2$

 $MVI\ R_1, 0H$

Opcode	Operand	Description
MOV	R_1, R_2	Move Immediate
	M, R_1	
	R_1, M	
MVI	R_1, data	Move Immediate
	M, data	
LDA	16bit address	load accumulator
LDA X	8 bit register pair	load accumulator indirect
LXI	Register pair, 16 bit data	load register pair immediate
STA	16bit address	Store accumulator direct
STAX	16bit address, register pair	Store accumulator indirect
XCHG	none	exchange HL with DE

~~Arithmetic Instructions~~ - It includes the instruction which performs addition, subtraction, multiplication, increment or decrement. The flag condition are altered after execution of an instruction in this group.

These include	Opcode	Operand	Description
	ADD	R M	Add register or memory to accumulator
	ADC	R M	Add register or memory to accumulator with carry
	ADI	8 bit data	Add immediate to accumulator
	ACI	8 bit data	Add immediate to accumulator with carry
	SUV	R M	Add register or memory to accumulator
	SUI	R M	Add register or memory to accumulator with carry
	INR	R M	Increment register or memory by 1
	INX	R M	Increment register A by 1
	DCR	R M	Decrement register or memory by 1
	DCX	R M	Decrement register A by 1

14/2/24

Logical Instruction - These instruction which perform logical operation like AND, OR, EXCLUSIVE NOR, COMPARE AND ROTATE instructions are grouped under this heading. The flag conditions are altered after execution of instruction in this group.

Opcode	Operand	Description
CMP	R M	Compare register or memory with accumulator
CPI	8 bit data	Compare immediate with accumulator
ANI	R M	Logical AND register or memory with accumulator
ANI	8 bit data	Logical and immediate with accumulator
XRA	R M	Exclusive OR reg. or memory with accumulator
ORA	R M	logical OR register or memory with accumulator
ORI	8 bit data	Logical OR immediate with accumulator
XRI	8 bit data	Exclusive OR immediate with accumulator

Branching Instructions - The instructions that are used to transfer the program control from one memory location to another memory location are grouped under this heading.

Opcode	Operand	Description
JMP	16-bit address	Jump unconditionally
JX	16-bit address	Jump conditionally

Machine Control Instruction - Include the instruction ~~created~~ related to interrupt and the instruction used to halt program execution. These instruction control machine functions such as halt, interrupt or do nothing.

Opcode	Operand	Description
HLT	None	Halt
NOP	None	No operation
SIM	None	This is a multipurpose instruction used to implement the 8085 interrupt I-S, 6-S, S-S and serial data output.
RIM	None	This is a multipurpose instruction used to read the status of interrupt I-S, 6-S, S-S and serial data input bit

Assign - 1

Q: Explain RIM and SIM instructions. Write difference between them.

Addressing modes - These are the instructions used to transfer data from one register to another, from memory to the register and from register to memory without any alteration in the content. Addressing modes in 8085 is classified into 5 modes :-

1. Immediate Addressing mode - In this mode, the data is specified in the instruction itself. The data will be a part of the program instruction. E.g. - MVI B, 3EH; move the data 3EH given in the instruction to B register. LXI SP 2700H^(16bit)
2. Direct Addressing mode - Data is directly copied from given address to the register. For e.g., LD B 5000K^(16bit); The data at address 5000K is copied to register B

3. ~~Indirect~~ Addressing Mode - Data is transferred from one register to another by using the address pointed by register. For e.g.:
MOV R₁, B ; Means data is transferred from the memory address pointed by the register to register R.

4. Register Addressing Mode - Data is copied from one register to another. E.g. - MOV R₁, B ; means data in register B is copied to register R.
5. Implied Addressing mode - Does not require any operand. The data is specified by opcode itself. For e.g. - CMP.

16/2/24
Timing Diagram - Display of the initiation of read/write and transfer of data operation under the control of 3 status signals, I0/M, S0, S1.
Each machine cycle is composed of many clock cycles. Since the data and the instructions both are stored in the memory, the microprocessor performs fetch operation to read the instruction or data and then execute the instruction.

A₀-A₁₅ → Address lines = 16, D₀-D₇ → Data lines = 8
A D₀-A D₇ → Address data lines (8) (Lower order) A₈-A₁₅ (Higher order)

It is graphical representation of step with respect of time. The timing diagram represent clock cycle and duration delay, content of address bus and data bus, type of operation read/write /status signals

Instruction Cycle - This term is defined as no. of ~~signals~~ steps required by the CPU to complete entire process i.e., fetching and execution of one instruction. Instruction cycle is completed when fetching and execution cycle is completed. IC = FC + EC

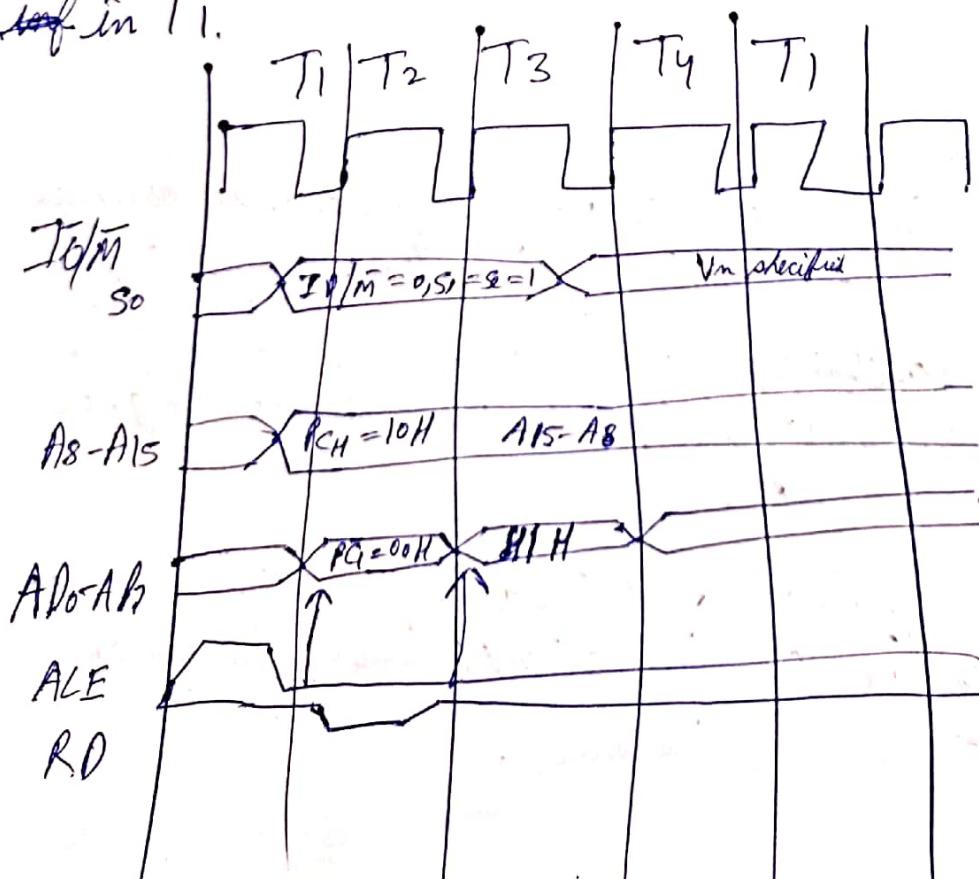
Machine Cycle - Term required by microprocessor to complete ~~operations~~ of accessing the memory device or I/O devices. In machine cycle, various operations like opcode fetch, memory read, memory write, I/O read, I/O write are performed. In 8085 microprocessor, has five basic machine cycle: They are: 1. Opcode Fetch Cycle (4T)

2. Memory Read Cycle (3T) 3. Memory Write Cycle (3T)
 4. I/O Read Cycle (3T) 5. I/O Write Cycle (3T)

Machine Cycle	Status Signal			Control		
	I _{O/M}	S ₀	S ₁	$\bar{R}P$	$\bar{W}R$	INTA
Opcode fetch	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	1	1	0	0	1	1
I/O write	1	0	1	1	0	1
INTA Ack.	1	1	1	1	1	0

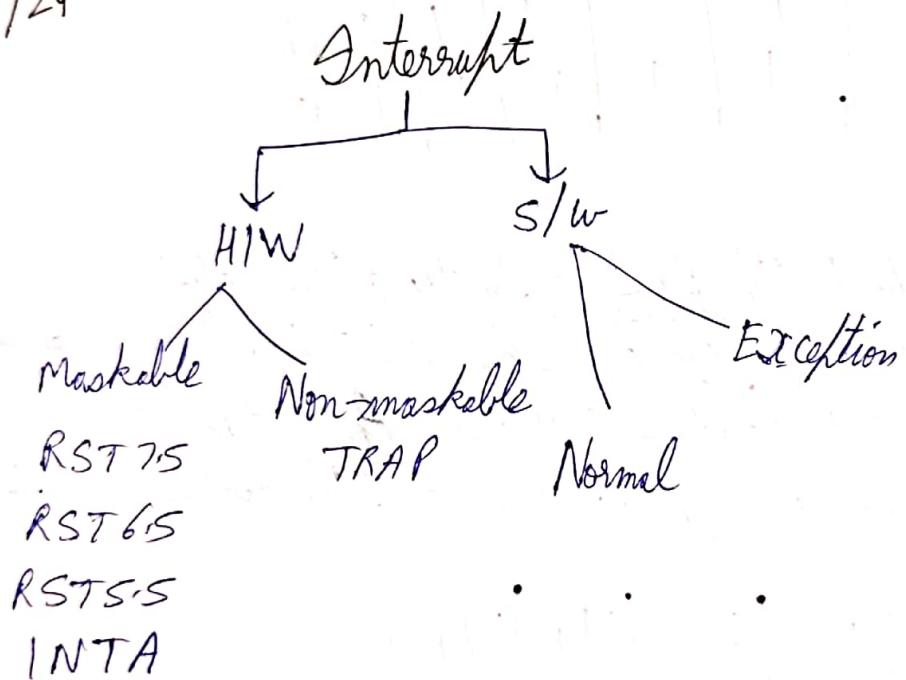
17/2/24

Opcode Fetch MOV B,C - In the given states:
 T₁ - The first clock cycle makes ALE higher indicating ~~ALE~~^{of first machine} ALE which loads low-order address 00H and on A₇ to A₀ and high order address 10H simultaneously on A₁₅ to A₈. The address 00H is less ~~off~~ in T₁.



- T₂: During T₂ clock, the microprocessor issue RD control signal to enable the memory and memory places 41H from 1000H.
- T₃: During T₃, the 41H is placed in instruction register and RD = 1(H)₁₆.
Disabled signals. It means memory is disabled in T₃ clock cycle.
The opcode cycle is completed by end of T₃ clock cycle.
- T₄: The opcode is decoded in T₄ clock and action as per 41H is taken accordingly in other words the content of C registers is copied from register B.

19/2/24



An interrupt is a signal that temporarily suspends the normal execution of a program and redirect the control to a specific interrupt service routine. Interrupt allow the microprocessor to respond a external event such as user input, system events, hardware signals without the need of constant polling.

- c) ISR - (Interrupt Service Routine) - Also called device driver in case of the device and called exceptional OR signal or TRAP handler in case of software interrupt. (Interrupt Handler).

Vector Interrupt - The processor automatically branches to the specific address in response to an interrupt

Non-vector Interrupt - The interrupt device should give the address to the interrupt service routine.

~~From~~ an INTR, is a non-vector interrupt. Hence, when a device interrupt through INTR, it has to supply the address of ISR after receiving interrupt acknowledge signal.

I.S.R. → ~~Instruction Set~~ → Processor → Interrupt Handler

↓
Instruction Cycle → Fetch + Decode + Encode + Execute + Read / Write