Module I

MICROPROCESSORS

→ General Architecture of comp =
performs 4 ()s —

* Accepts the data & instructions as input through an input unit.
* Stores the data & ins in its memory & retrieves the same data when required.
* processes the data to convert into useful info.
* communicates the info as output.

* comp consist of 4 ()ally indpndnt main parts —
a) Input unit (b) output unit.
c) Memory (d) Control processing unit.

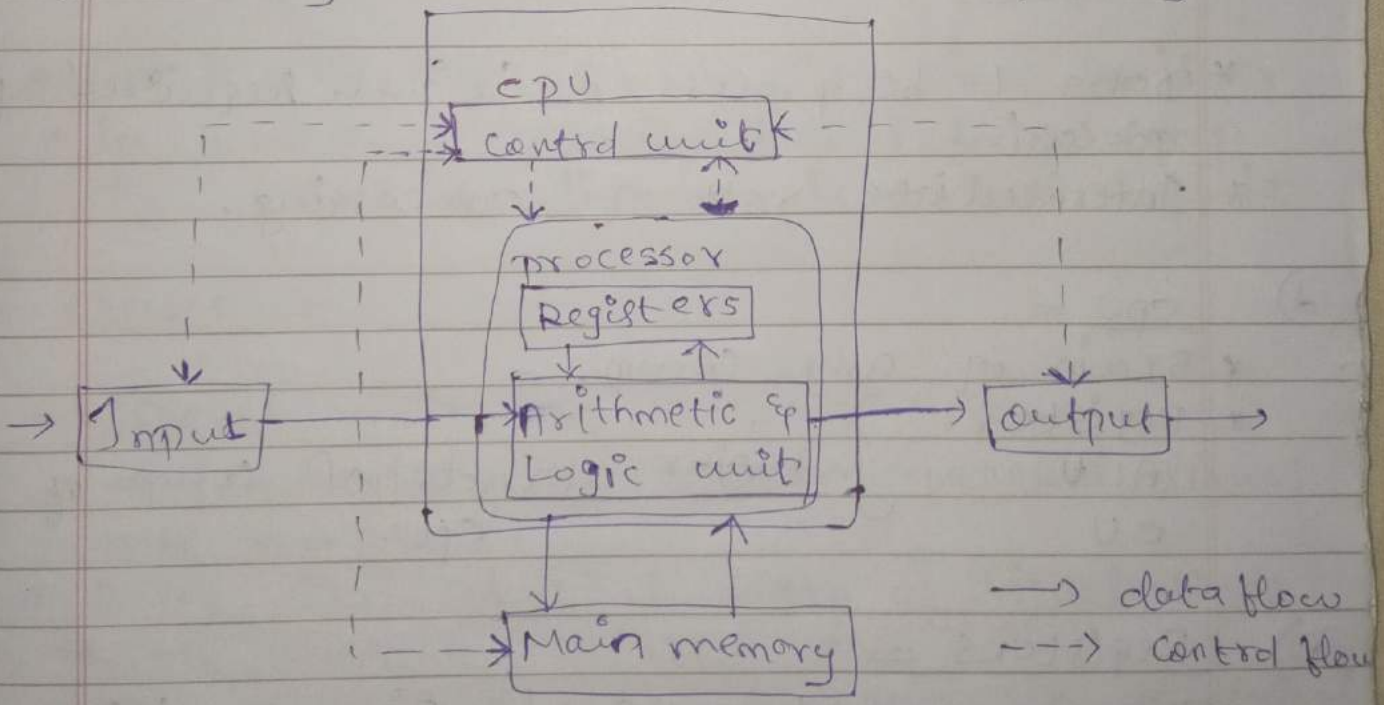
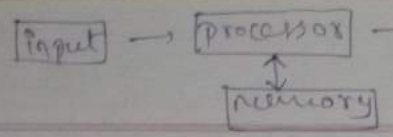


fig: Main units of comp.

a) Input unit =
4 ()s —



* Accepts the data & ins... from outside world
* converts the received analogue data into comp acceptable (digital) form.
* Supply the digital form of data & ins to comp for further processing.

b) output unit =
4 C)s —
* Accepts the rslts produced by the comp. (binary coded form).
* Convrt these binary coded rslts to human acceptable form.
* Supply the converted rslts to ouside world

c) memory =
* Data to be processed & ins required for processing.
* Intermediate rslt of processing.

d) CPU =
* Brain of any comp.
* Register } major structural comp of
ALU } cpu.
CU }

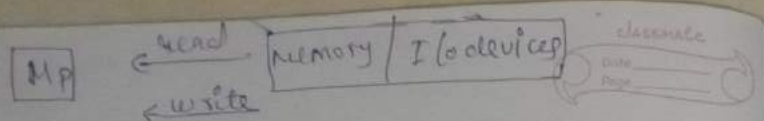
e) Registers =
* Temporary storage locations within cpu where the data fetched from memory can be held.
* Most comp use several types of registers each designed to perform a specific

* eg- Accumulator, program counter, memory buffer (R), Instruction (R) etc

f) ALU =
* It is the place where the actual execution of the ins... takes place during data processing operation.
* The type & no. of arithmetic & logic operations that a comp can perform is determined by engineering design of ALU.
* Almost all ALU's designed to perform — add, sub, multiply & ÷ & logical operations like <, >, = .

g) CU =
* Decodes the ins... in the ins... register.
* In general, controls the entire comp by means of timing & control circuit.

h) Buses =
* Different units of a comp system are interconnected by common parallel signal lines → buses.
* Buses are physical grp of signal lines that have a related C).
* It allow for the transfer of electrical signals b/w different parts of comp system.
* 3 principal buses in a comp system —
a) Data bus = Bus over which data going to & from I/O device/memory will flow (bi-dirn bus).



b) Address bus =
* Bus over which addresses going to memory / I/O devices will flow.
* Uni-dir.

c) Control bus =
* used to snd various control signals to & frm I/O devices / memory.

* The width of a bus is the no of signal lines dedicated to transferring info
* These 3 buses together → system bus.

=> Introduction to Mp. =

* Mp is a multipurpose, programmable, clock driven, registes based electronic that reads binary instn frm a storage device → memory, accepts binary data as input & processes data & provide rslt as output

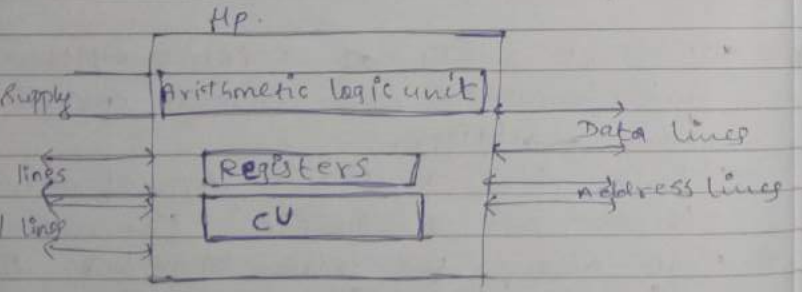


fig: Inside & outside of a Mp.

* fetch = Instn taken frm memory
* Decode = understand the instn
* Execution = execute the instn

* A mp is an imp part of a comp architecture which you will not be able to perform anything on your comp.
* It is a prgmble device that takes input performs some arithmetic & logical operation ovr it & produces the desired output.
* Mp performs 3 basic
  a) performs some arithmtic & logic operations
  b) Data in Mp can move frm 1 location to anthr.
  c) Has a prgrmble counter (PC) register that stores the address of nxt instn based on the value of PC. Mp jumps frm 1 location to anthr & takes decisions.

=> Micro comp & micro controllers =

a) micro comp
- defined as small sized & inexpensive
- capable of performing floating point arithmtic operations.
- Has the same architectural blk str that is prsnt on a comp.
- bcz of mass production, they are becoming still popular.

b) Micro Controllers =
* Has centre of embedded system
* memory & I/O components are internal to it.
* Lower cost
* More no. of registers
* required less instn

c) microprocessor =

* Has centre o/s comp system.
* Memory & I/O comp are external to it.
* Higher cost
* Less no. of registers
* Requires more instr~

=> n-bit MP =

* A MP is called n-bit MP depending on size of internal data bus.
* n-bit MP will have registers that can hold n-bit data.
* n-bit MP can also be used to perform on larger data.

=> Evolution of MP =

a) 1st generation (4-bit MP) =

* Introduced in the yr 1971 - 1972
* It was named Intel 4004 since it was a 4-bit processor.
* was a processor on a single chip.

b) 2nd generation (8-bit MP) =

* Introduced in 1973
* It was 1st 8-bit MP which could perform a. & logic operations on 8-bit word
* named Intel 8008. & another version was Intel 8088.

c) 3rd generation (16-bit MP) =

* Introduced in 1978, which were 16-bit processors with a performance like minicomp.

d) 4th generation (32-bit MP) =

* several diff~ companies introduced the 32-bit MP, but most popular 1 is the Intel 80386.

e) 5th generation (64-bit MP) =

* From 1995 to now we are in 5th gene~
* After 80856, Intel came out with a new processor namely pentium processor followed by pentium pro CPU.

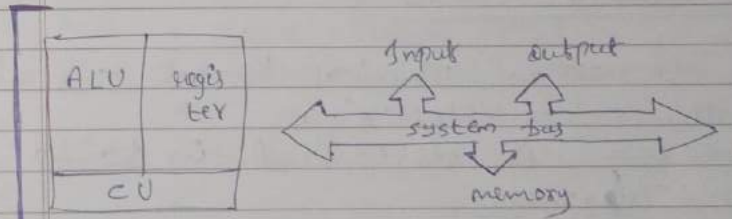
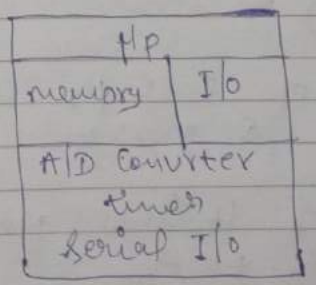


fig: Basic str of micro comp.



→ Basic diagram of micro controller

(Buzzer) → Intel 8085 = 8-bit Processor
Intel 8086 = 16-bit Processor

⇒ Intel 8085 =

* It is an 8-bit Mp introduced by intel in
* It was binary compatible with more famous Intel 8080 bt required less supporting hardware.
* The '5' in model now came from the fact that 8085 requires only a +5V (v) power supply rather than the +5V, -5V & +12V supplies the 8080 needed
* main features of 8085 Mp —
  - 8-bit Mp
  - Manufactured with N-MOS technology.
  - Data bus is a grp of 8 lines D_0-D_7
  - Supports external interrupt request.
  - Has 16-bit pc & 16-bit Stack pointer (SP)
  - requires a Signal +5V power Supply.

→ 8085 Architecture =

* refers to actual physical layout of diff elements of a Mp chip.
* Architecture is described in terms of
a) Pinout Diagram
b) Functional block diagram.

Pinout Diagram =

The 40 Signals of 8085 Mp can be classified into 6 grps —

a) Address Bus =

* It is a grp of 16 lines generally identified as A_0 to A_{15}.
* Address Bus = carries address info
* Data Bus = carries data
* control Bus = carries control signals

* Unidirectional - bits flow in 1 (dir). (from Mp to memory)
* 8085 use a 16-bit address bus to carry memory address & capable of addressing 64KB of memory.

b) Data Bus =

* Grp of 8 lines used for data flow.
* Bi dir~
* Identified as A_7 - A_0
* used both as low order address bus as well as the data bus in time multiplexed mode.

c) control & status Signal =

* Grp of Signals includes 2 control Signals ($\overline{RD}$ & $\overline{WR}$), 3 status Signals ($IO/\overline{M}$, S_1 & S_0) and 1 Sple Signal (ALE).

a) ALE (Address Latch Enable) =

the going pulse generated every time the 8085 begins an operations.

b) $\overline{RD}$ (read) =

This active low read Signal indicates that the selected memory address is to be read & the data is available on data Bus.

(c) $\overline{WR}$ (write) =

This active low write Signal indicates that the data on the bus is to be written on to selected memory location.

MP ⟷ Memory

(c) IO/$\overline{M}$ (Input-output/memory) = This status signal is used to differentiate b/w I/O & memory operations.

(e) S_1 & S_0 = These status signals indicates the type of machine cycle

S_1	S_0	operation
0	0	Halt
0	1	write
1	0	read
1	1	Fetch

d) Externally initiated signals =

* TRAP (input) = non-maskable RESTART interrupt
* RST 5.5, RST 6.5 and RST 7.5 (restart interrupts - Input) = transfer program control to specific memory location. priority order → 7.5, 6.5 & 5.5.
* INTR (Interrupt request - Input) = Has least priority among the interrupts
* $\overline{INTA}$ (Interrupt Acknowledge - output).
* HOLD (Input) = indicates that an external device is requesting the use of address & data buses.
* HLDA (Hold Acknowledge - output) = Indicates that cpu has received the HOLD request.
* $\overline{RESET\ IN}$ (Input) = when the signal on this line goes low, PC & set 0.
* RESET OUT (output) = Indicates MP is being reset

e) Serial I/O ports = 2 serial transmissions
① SID (Signal Input data)
② SOD (Signal output data)

In serial transmission, data bits are sent over a single line, 1 bit at a time.

→ ~~Functional block diagram of 8085~~

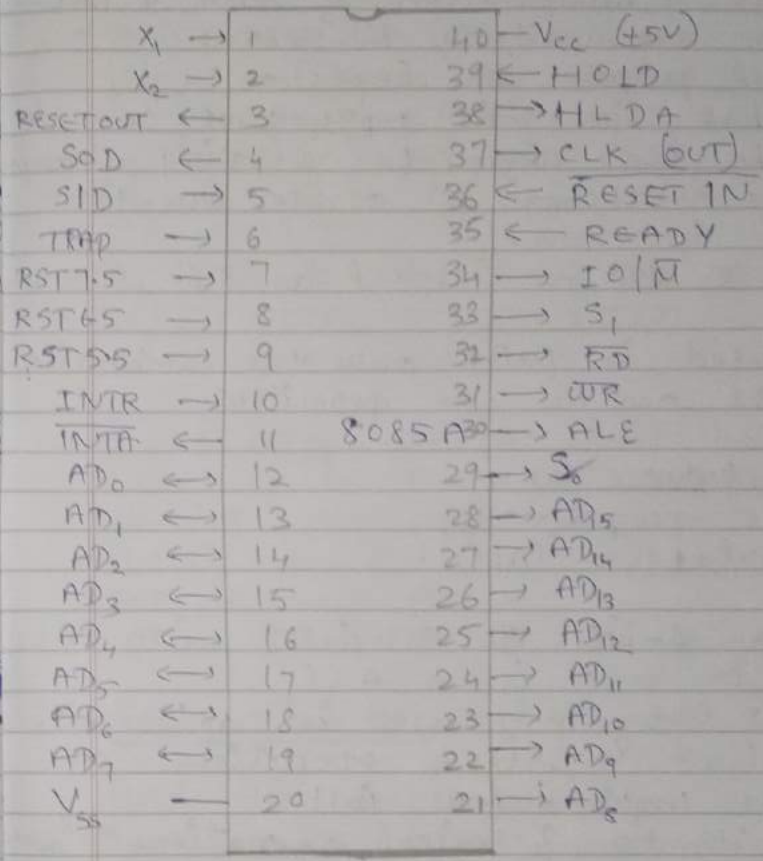


fig: Pin out diagram of 8085.

II Functional block Diagram of 8085 =

Internal architecture of 8085 -

I Registers =

* registers are used by MP for temporary storage & manipulation of data & instn.
* registers are available in 8085 :

1) General purpose (R) (GPRs) =

* 8085 has six 8-bit GPRs that can be used by the programr for a variety of purpo
* Also → Scratchpad (R) as user can store data in them.
* These (R) are labelled as B, C, D, E, H & L
* When used in pairs only the combinations BC, DE and HL are permitted.

2) Sple purpose (R) =

3 Sple purpose (R).

(i) Accumulator = (ACC).

* It is an 8-bit (R) associated with ALU
* (R) A in 8085 is an ACC.
* ACC is extensively used in arithmtic, logic, load & store operations as well as input/output instn
* All arithmtic & logical operations are performed on ACC contnts.
(i.e) 1 of the oprand is always taken into the ACC.
* Other operand for an arithmtic / logic operation may be stored either memory / 1 of the GPRs.
* Final rslt of an arithmtic / logical operation is also placed in ACC.

(ii) Flag registers =

* It is an 8-bit (R), in which 5 individual flip-flops serve as status flags : S (Sign flag), Z (zero flag), AC (Auxiliary carry flag), P (parity flag) and CY (carry flag)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0 ← bit position
S	Z	X	AC	X	P	X	CY

X → unspecified

* The 5 flags are set/reset individually according to the condtn which arise during an arithmtic / logical oprtions.
* If a flip-flop for a particular flag is set, it indicates 1, when it is reset, it indicates 0.

(iii) Instruction (R) =

Holds the opcode (operation code) of the instn which is being decode & executed.

3) 16-bit registers =

a) PC (prgm counter) =

* A prgm is a sequence of instn.
* PC is a sple purpose 16-bit (R) which at a gvn time, store the address of the nxt instn to be fetched.

* PC register act as a pointer to the next inst..
* MP increments the contnt of PC (R) duri tne execution of an inst.. So tha it points to the address of nxt inst.. in the prgm at the end of the execution of an inst..
* pc depends on the ntre of the inst

b) SP (Stack pointer) =

* A stack is a reserved area of the memory where temporary info may be stored.
* 16-bit SP (R) is used by the prgmer to maintain a stack in the memory
* SP (R) always hold the address of the top elemnt of data stored in the stack.

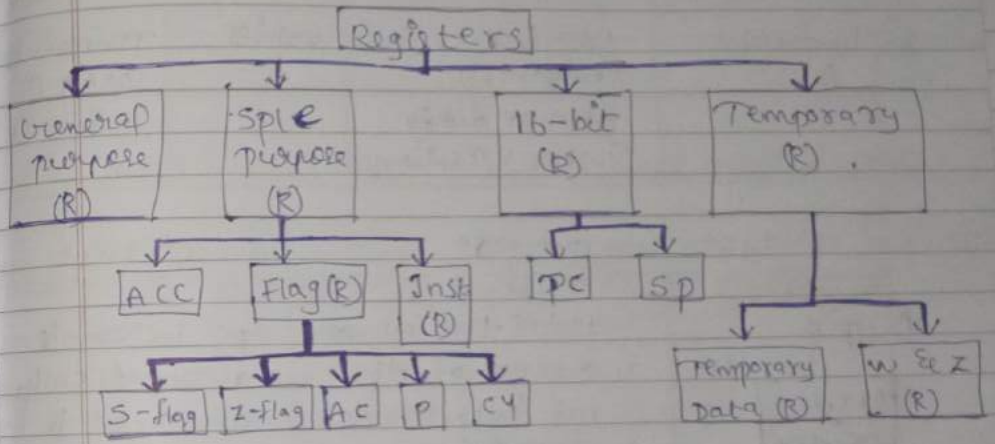
4) Temporary (R) = (2 types)

a) Temporary data (R) =

* ALU has 2 input -
  1 input is supplied by the Acc & other from temporary data (R).
* The prgmer cannot access this temporary data (R).
* It is internally used for execution of most of the arithmtic & logic inst..

b) W and Z (R) =

* Are temporary (R)
* used to hold 8-bit data during execution of some inst..
* These (R) are not available for prgmrs.



~~II Arithmetic & logic unit~~ =

(Short differences b/w regis type) →

①	(R) type	purpose	Size	eg.
	1) GPR	used for storing data & inst.. temporarily	varies	AX (Acc (R) BX (Base (R) CX (count (R) DX. (Data (R)
	2) SPR	used for specific ops within the processor.	varies	Acc (R) Flag (R) Inst.. (R).

3) 16-bit (R) (memory (R))	used for storing 16-bit data	16 bits	AX, DX BX, SP CX, BP (Base point SI (Source i DI (Destination index
4) Temporary (R)	used for temporary storage during instn execution.	varies	Temporary Data (R) w&z (R

2) Flag (R)	used for storing Status info about arithmetic & logical operations	Contains info about the outcome of a. & logical operations.
3) Instn (R)	used for storing the instn currently being executed.	Contains the opcode of the instn being executed

II ALU

III Instn Decoder & Machine cycle encoder =

* processor 1st fetches the opcode of the instn from memory & stores this opcode in the instn (R).
* It is then sent to instn decoder.
* Instn decoder decodes it & gives the timing and control signals.
* 8085 executes 7 different types of machine cycles.
* Gives the info about which machine cycle is currently executing in the encoded form on S_0, S_1 & IO/$\overline{M}$ lines.

IV Address Buffer =

* 8-bit unidirn buffer
* used to drive extend high-orders address bus ($A_{15} - A_8$)

V Address / Data buffer =

* 8-bit bi-dirn buffer.
* used to drive multiplexed address/data bus (i.e) low-order address bus ($A_7 - A_0$) & data bus ($D_7 - D_0$).

(2)

(R) type	purpose	Function.
1) PC [Prgm counter]	Holds the memory address of the nxt instn to be executed.	PC increments automatically after each instn execution in memory.
2) SP [Stack pointer]	Hold memory address of the top of the stack	SP increments/decrements automatically data is pushed popped frm stack

(3)

(R) type	purpose	Function.
1) ACC.	used for storing intermediate results of arithmetic & logical operations.	used as a default register for arithmetic & logical operations on mp.

VI Increment / decrement Address Latch =

* 16-bit (R) is used to increment / decrement the contents of pc / sp as a part of execution of inst.

VII Interrupt control.

* processor fetches, decodes & executes inst. in a sequence.
* After execution of the spl routine, the prgm control must be transferred to the prgm which processor was executing before the occurance of spl condition. The occurance of this spl condn → interrupt
* I. control blk has 5 interrupt input → RST 5.5, RST 6.5, RST 7.5, TRAP & INTR & 1 acknowledge signal INTA.

VIII Serial I/o Control =

* Data transmission over long distence, it is necessary to transmit data bit by bit to reduce the cost of calding.
* In serial Communication 1 bit is transferred at a time over a single line.
* 8085's serial I/o controls provides 2 lines — SOD & SID.
a) SOD = used to transmit serial data fsm Mp to ex. device
b) SID = used to receive serial data fsm ex. dev

IX Timing & Control circuitry =

Timing unit is responsible for synchronizing all mp operations with the clock & the control circuitry along with control/ of fetching & decoding operations.

⇒ Memory Interfacing =

* while executing a prgm, the Mp need to access memory to read inst. codes & data stored in memory.
* Memory has certain signal requirements to write into & read fsm its registr.
* Memory Interfacing → process involves designing a circuit that will match the memory requirements with Mp signals.
* eg → R/w memory chip.

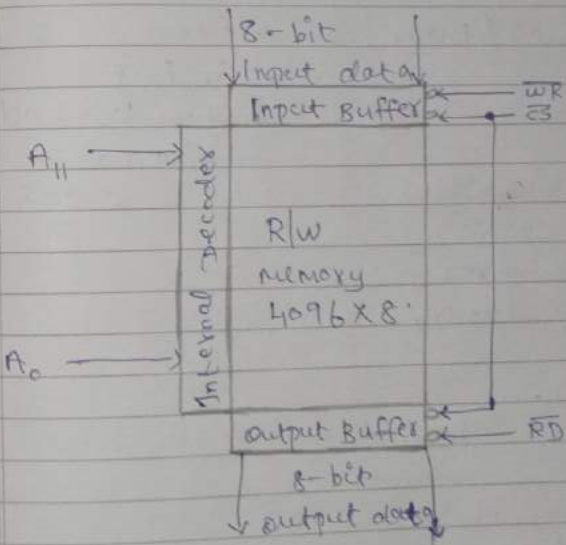


fig: R/w memory Chip.

* primary () of Memory Interfacing is that Mp should be able to read fsm & write into a gvn registes of a memory chip.

[volatile → its contents are lost when the power is turned off]

To enable this, MP should –
a) be able to select the chip
b) identify the (R)
c) enable the appropriate buffer.

I Memory classification =

1) primary (M) =
* volatile
* Refers to (M) that can be accessed directly by the CPU
* Also → main (M) / internal (M).
* 2 types :

a) RAM =
* Read from & written to by the CPU.
* used for temporary storage of data & inst~
* volatile (M).

b) ROM =
* that contains data that cannot be modified by the CPU.
* used to store permanent prgm that are required to start the comp / MP.
* Non-volatile

(2) Secondary (M) =
* Non-volatile
* (M) that is external to the CPU
* used to long-term storage of data & inst~
* eg → hard disk drives, memory cards.

II Memory Addressing =

* process of specifying the location of data / inst~ in the memory that the MP needs to access / retrieve.
* It is essential for the MP to perform various operations like reading & writing data, & executing inst~
* It can be vary depending on the architecture & design of the MP.

III Memory Mapping =

* process of assigning memory locations to various components of comp system.
* It enables the MP to access data & inst~ stored in these components by referring to their corresponding memory locations.
* Common m. mapping is the linear / flat memory model., where all the memory locations are treated as a contiguous blk of memory.