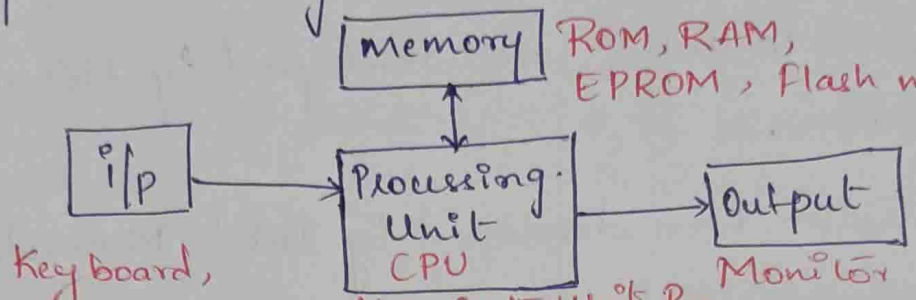
Unit - 1

Introduction to Micro Controller

Introduction :-

* What is a Computer ?

It is a programmable machine that reads binary instruction from memory, process the data given by input according to the instruction and provide o/p.



Block diagram of Computer

* What is Microprocessor? [general purpose dig Comp Central Process Unit [CPU]]
  - Processing Unit [Computer on chip]
- In 1960, CPU was designed using digital gates. Using Semiconductor technology.
- which helped them to fabricate a thousands of gate on a single chip.
- ∴ the entire CPU along with its timing function was built on a single chip and it was name as micro processor.
- Intel – 4 bit microprocessor (4004) – 1971.
  - 8 bit microprocessor (8080) – 1974.
  - 16 bit microprocessor (8086) – 1978.

Some of the limitation in microprocessor let the Companies which designed the processor to move on to micro controller.

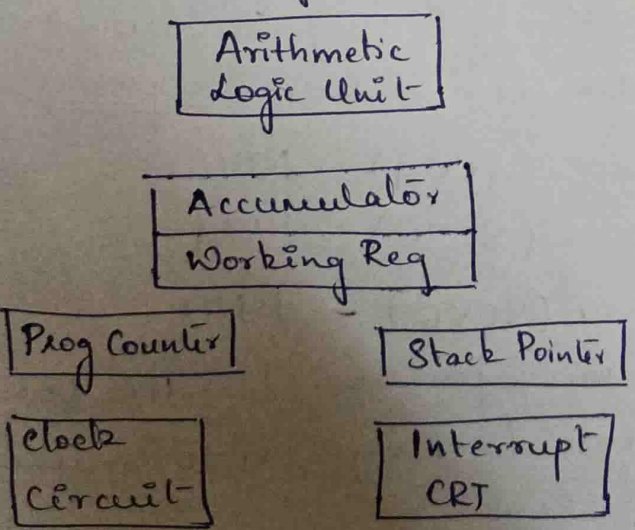
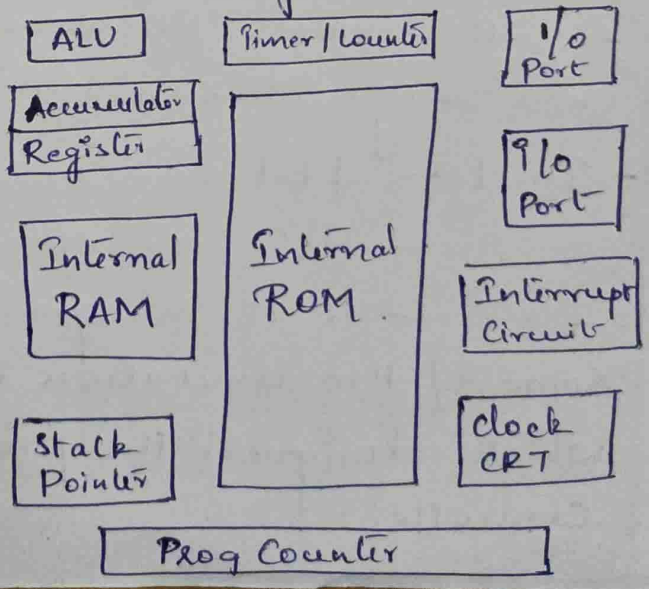
Main limitations are.

1. It requires external memory to execute a program.
2. It cannot be directly interfaced with i/o devices Peripheral chips are needed.

* What is a micro controller?
  - A device which contains microprocessor, memory, i/o devices on a single chip.

- first 4 bit micro controller was developed by Hitachi, National, Toshiba.
- 8bit microcontroller were developed by Intel, Motorola, Philips, Microchip technology.
- 8051 was introduced by Intel - 1981
  It is a 8bit controller.

* Difference between Micro Controller and Microprocessor

Microprocessor	Microcontroller
* Block diagram. 	* Block diagram. 

Microprocessor	Micro Controller
* Contains ALU, general purpose register, stack pointer, Prog Counter, clock timing CRT, Interrupt CRT	* Contains microprocessor, and built in ROM, RAM, i/o devices, timers and counter.
* Many instruction to move data b/w memory and CPU	* It has one/two instruction to move data b/w memory and CPU.
* It has 1/2 bit handling instruction	* It has many bit handling instruction.
* Less no of pins for multifunction	* More no of pins for multifunction.
* It has Single memory map for data and Code.	* It has seperate memory map for data and Code.
* Access time for memory & i/o device are more.	* Less Access time as it has built in memory & i/o devices.
* It requires more hardware	* less hardware hence PCB size is less and ↑ reliability.
* More flexible in design point of view. (designer can decide the amount of memory, i/o port)	* less flexible in design point of view. (fixed amount of ROM, RAM, i/o port on chip)
* Large no of instruction set	* Limited no of instruction set.

Difference b/w RISC and CISC CPU architecture

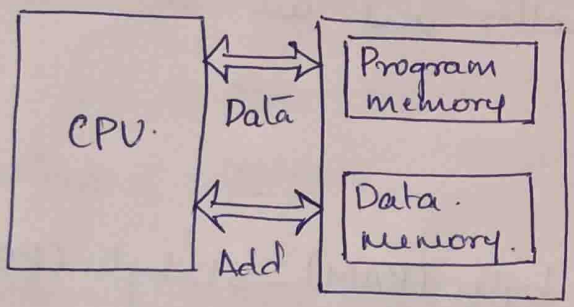
RISC	CISC
* Reduced instruction Set computer takes only 1 cycle for execution	* Complex instruction Set computers takes multiple cycle
* very few instruction's refer memory.	* MOST of instruction refer memory.
* Multiple register set	* Single register set.
* few addressing mode & most instruction have register to register.	* Many addressing mode.
* fixed formate instruction	* Variable formate instructio
* Instruction are executed by hardware.	* Instructions are executed by micro programs.
* Highly pipelined.	* less pipelined or Not pipelined.
* Complexity is in the compiler.	* Complexity is in the micro program.
* Code size is Large hence Low cycles/sec.	* Small code in size hence high cycles/sec.
* more transistors are used for memory register	* transistors are used for. Storing Complex instruction.
* instruction are of same length & only 8bit.	* instruction length varies & 8bit, 12bit, 32bit.

Von-Neumann Architecture

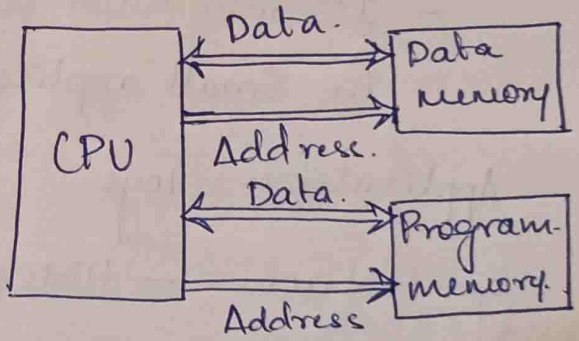
- Processor is having architecture with single bus & single memory to hold both instruction and data.
- this makes the program execution slow.

Harvard Architecture :

- Program memory and data memory are connected using seper-te address and data bus.
- this makes the program execution fast.
- It uses seperate memories for their instruction & data. and has dedicated buses for each of them.
- Instruction and operation, operands are fetched simulta-neously.



Von-Neuman Architecture.



Havard Architecture.

* Block diagram.
* It uses single memory space for both instructions and data. It is also called Stored Program Computer.
* It is not possible to fetch inst" Code and data simultaneously.

* Block diagram.
* It has seperate program memory & data memory.
* Instruction code & data are fetched simultaneously.

Von-Neumann	Harvard
4. Execution of Instruction takes more Instruction Cycle.	4. Execution of instruction takes less instruction cycle.
5. Uses CISC processor.	5. Uses RISC processor.
6. Main feature is pre-fetching.	6. Main feature is instruction parallelism.
7. Computer based on princeton architecture also know as Control flow or Control driven Computer.	7. they are also called as data flow or data driven processors.

* Micro Controller Surevy :-

1. 4bit micro Controller :-

- CPU can handle only 4bit of data at a time.
- 1st introduced micro controller and are still used in small applications.

Application : Toys.

Ex Hitachi – HMCS40 – 32 bytes (RAM) – 512 bytes (ROM)
Toshiba – TLCS47 – 128bytes – 2K bytes.

2. 8 bit Micro Controller :-

- CPU can handle 8 bits at a time.
- Proven to be very useful data size.
  [ASCII stored in 8bit formate hence a good choise for data Communication].
- Interfaced easily to data buses of 8 bits.

Application : TV, VCR, CamCoder, Videogames.

bit = 0 or 1
Nibble - 4 bits
1 Byte = 8 bits
1 Kilo Byte - 1024 by

Eg Intel - 8051 - 40 pins - 128 bytes of RAM.
Intel - 8052 - 40 pins - 256 bytes.

* 16 bit Microcontroller :-

- CPU can handle only two bytes at a time.
- Designed for high speed / high performance application.
- they provide large program & data memory space.
- More flexible i/o capabilities.
- Less cost

Eg Intel - 80C196 - 64 pins - 1Kbyte RAM.
Hitachi - H8/532 - 84 pins - 232Kbyte - RAM.

* 32 bit Microcontroller :-

- Can handle 32 bit data at a time.
- Application Robotic Control, image processing

Eg Intel 80960
Arm processor.

* Salient features of 8051 microcontroller :-

Jan-09 6M

1. 8bit CPU.
2. Internal ROM of 4 Kbytes
3. Internal RAM of 128 bytes
4. 32 i/o pins
5. two 16bit Timers / Counters (T_0 & T_1)
6. 8bit stack pointer (SP)
7. 8bit program status word (PSW)
8. 16 bit program Counter (PC) & Data pointer (DPTR)

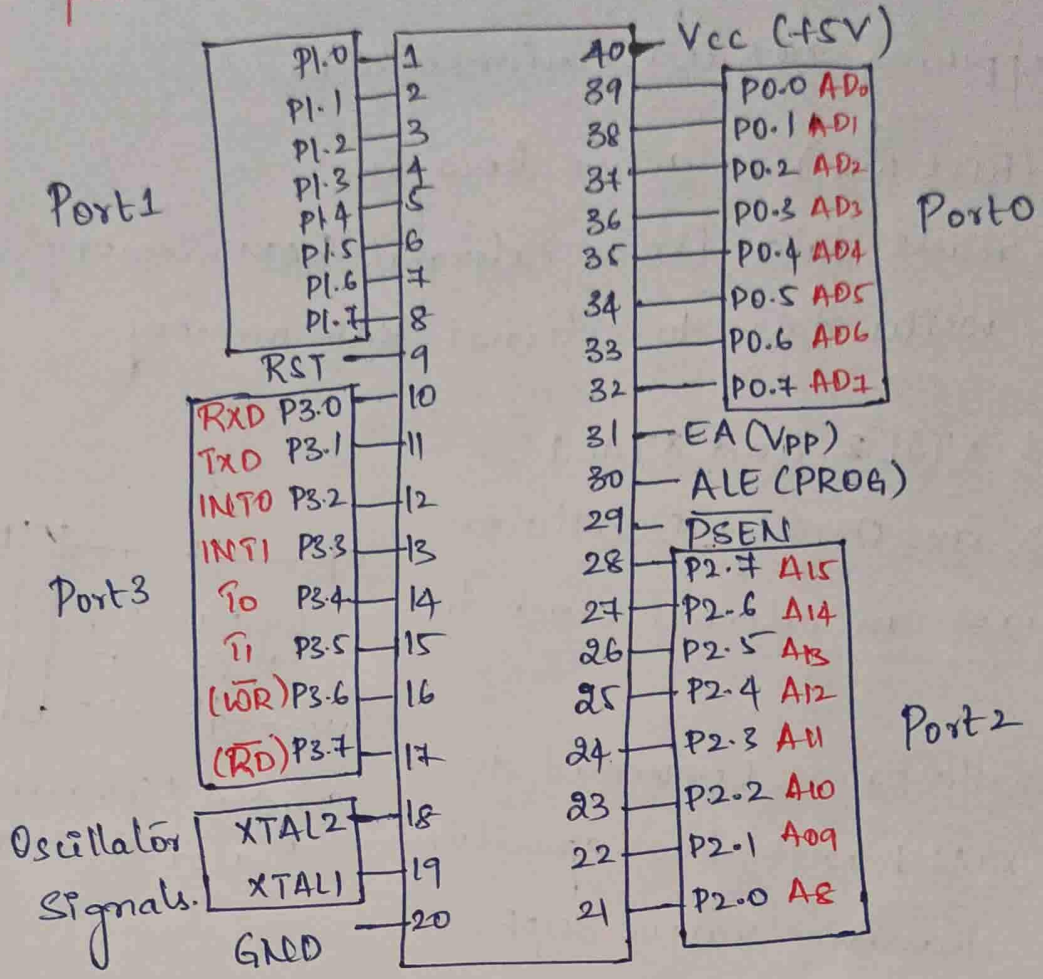
9. 6 interrupt Sources with priority levels.
10. Full duplex Serial data Trx / Rrx.
11. On chip Oscillator Circuits.
12. 4 ports $P_0 - P_3$.
13. 4 Register Bank [$R_0 - R_7$]

* Brief History of 8051:

- In 1981, Intel introduced an 8bit micro controller called 8051.
- It has 128 bytes of RAM, 4K bytes of onchip ROM, two timers, One Serial port and 4 ports (each 8 bit wide) all on Single chip.

- It is also refered as a "System on a chip".
- the 8051 is an 8bit processor. Data larger than 8bit has to be broken into 8bit pie.
- Although the 8051 can have a maximum of 64K bytes of on chip ROM, many manufactures have put only 4K bytes on the chip.
- Hardware Circuitry is implemented using a Crystal & timing Capacitors to generate high frequency Signals. this Circuitry is called the Oscillator Circuit.

*PIN diagram of 8051 - has 32 I/O pins, 4 - 8bit parallel port (P0 - P3). 4 ports are bidirectional (i.e I/O ports)



* Port 1 (Pins 1-8) - can be configured as i/o pins.

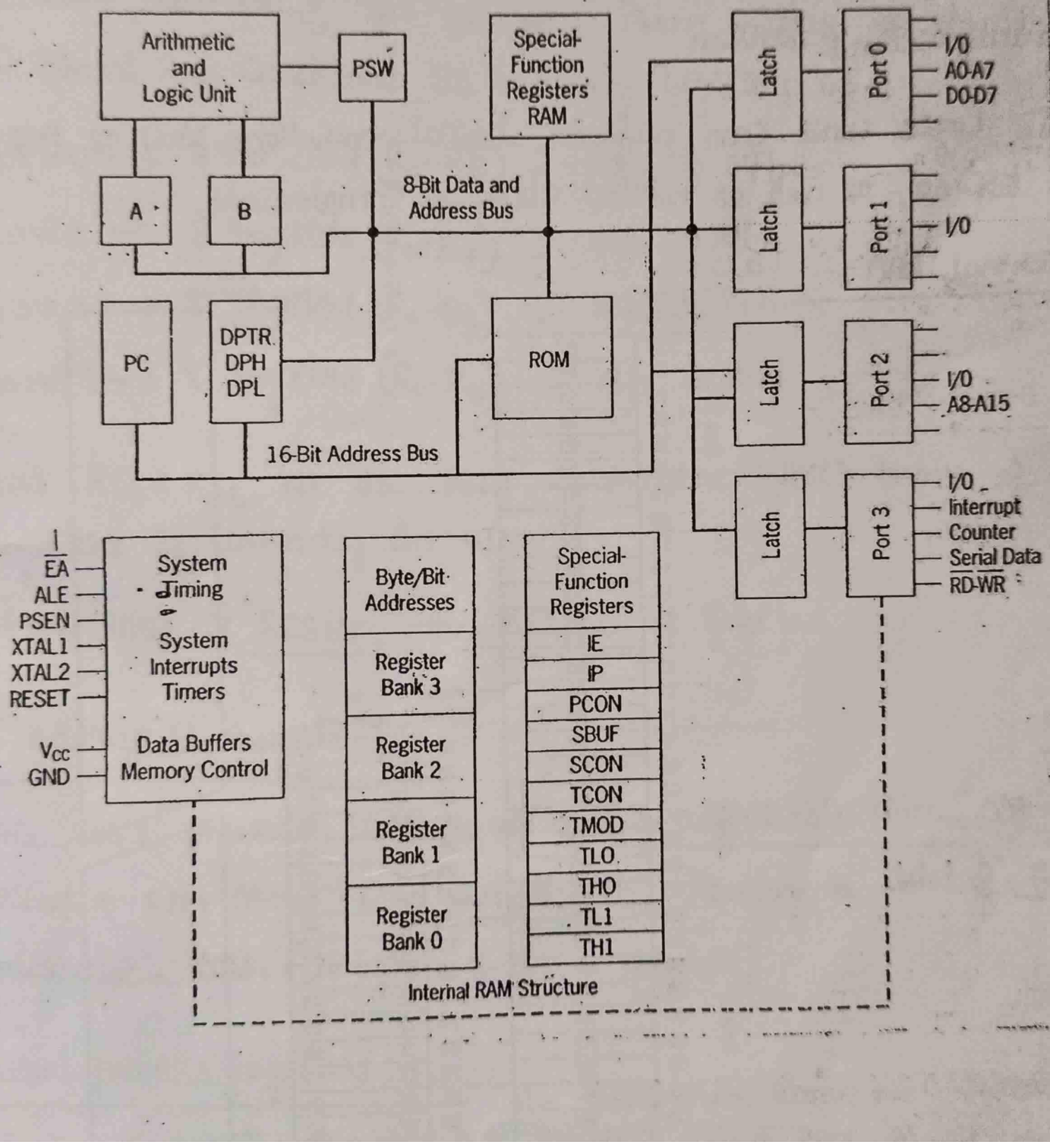
* PIN 9 (RST) - when active high is applied to the pin micro controller will terminate all its activities and. Resets and load 0000 to Program Counter.

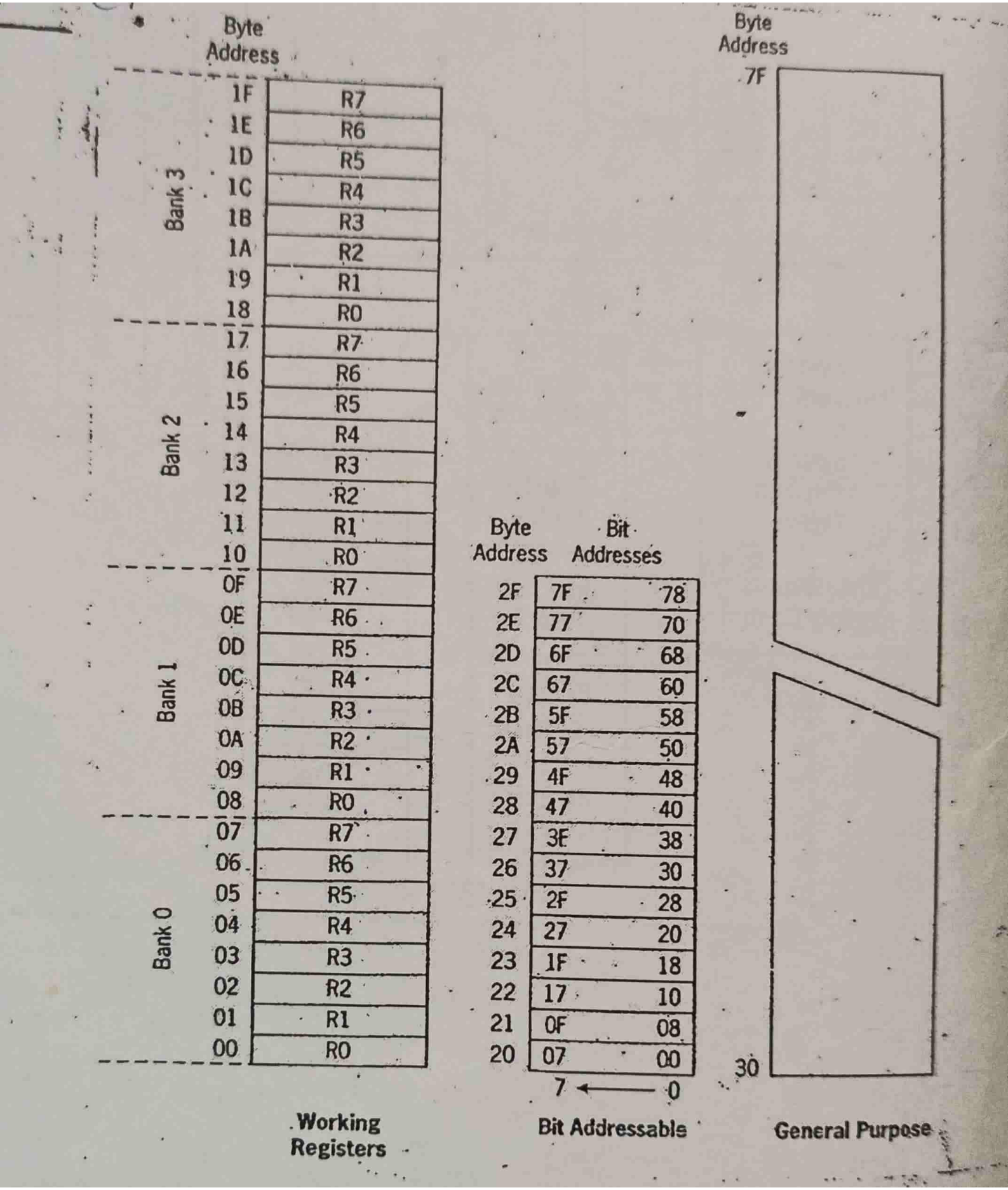
Pin 10-17 Port 3 - each of these pins can be configured as i/o pins.

Pin 10-11 (RXD & TXD) - 8051 has Serial data Communication
TXD line used to transmit data out of 8051.
RXD line used to receive data into 8051

Pin 12-13 (INT0 & INT1) - they are interrupt pins that are triggered by external circuit.

8051 Block Diagram




Byte Address
Bank 3
1F R7
1E R6
1D R5
1C R4
1B R3
1A R2
19 R1
18 R0
Bank 2
17 R7
16 R6
15 R5
14 R4
13 R3
12 R2
11 R1
10 R0
Bank 1
0F R7
0E R6
0D R5
0C R4
0B R3
0A R2
09 R1
08 R0
Bank 0
07 R7
06 R6
05 R5
04 R4
03 R3
02 R2
01 R1
00 R0
Working Registers
Byte Address
Bit Addresses
2F 7F 78
2E 77 70
2D 6F 68
2C 67 60
2B 5F 58
2A 57 50
29 4F 48
28 47 40
27 3F 38
26 37 30
25 2F 28
24 27 20
23 1F 18
22 17 10
21 0F 08
20 07 00
7 ← 0
Bit Addressable
Byte Address
7F
30
General Purpose

* Working register :-

- 1st 32 byte from address 00h to 1Fh of Internal RAM consist of 32 working register.

Bank0 → 8 register (R0-R7) : 00h to 07h.

Bank1 → 8 register (R0-R7) : 08h to 0Fh.

Bank2 → 8 register (R0-R7) : 10h to 17h.

Bank3 → 8 register (R0-R7) : 18h to 1Fh.

- Bit RS0 & RS1 in PSW determines which Bank is used currently.
- When 8051 is RESET, Bank 0 is selected.

* Bit addressable register :-

- 8051 provides 16-byte of bit addressable area.
- It occupies RAM area from 20h to 2Fh
- forming a total of 128 bit i.e (16 byte x 8 bit = 128 bit).

* General purpose register :-

- addressable area from 30h to FFh.
- addressed by bytes.

* Internal ROM :-

8051 has 4K byte of Internal ROM.

* A register (Accumulator)

- It's a 8bit register, widely used for many operation like addition, Subtraction, multiplication, division.
- Also used for data transfer b/w 8051 and external memory.

* B register :-

Used with A-register for multiplication and division.

* Data pointer (DPTR):-

- 16 bit register, holds 16-bit address.
- Can be split into two parts.

DPH - data pointer high byte.

DPL - data pointer low byte.

* PC program Counter

- 16 bit register which holds the address of the next instruction to be executed.
- Only register that does not have an internal add"

* Program status word (PSW)

- Also called as flag register.

CY	AC	F0	RS1	RS0	OV	-	P

- Carry flag [CY]

After performing arithmetic and logic operation if there is a carry out from MSB then CY=1 else CY=0.

- Auxiliary Carry Flag (AC)

After performing arithmetic and logic operation if the carry is generated from D_3 and D_4 then.

AC=1, else AC=0.

- RS1 and RS0 — Register Bank Selector.

00 — Bank0 10 — Bank2

01 — Bank1 11 — Bank3.

- Overflow Flag (OV):-
  - Set to 1 if either of the following Condition occurs
  1. Carry generated from D_6 to D_7 but no carry out of D_7 (CY=0)
  2. there is a carry from D_7 bit (CY=1) but no carry from D_6 to D_7.

* Parity flag (P):-
  - Indicate the number of 1's present in accumulator
  1. If no of 1's in accumulator is odd P=1.
  2. If no of 1's in accumulator is even P=0.

* Special function register (SFR)
  - they are just used to perform operations on micro controllers.

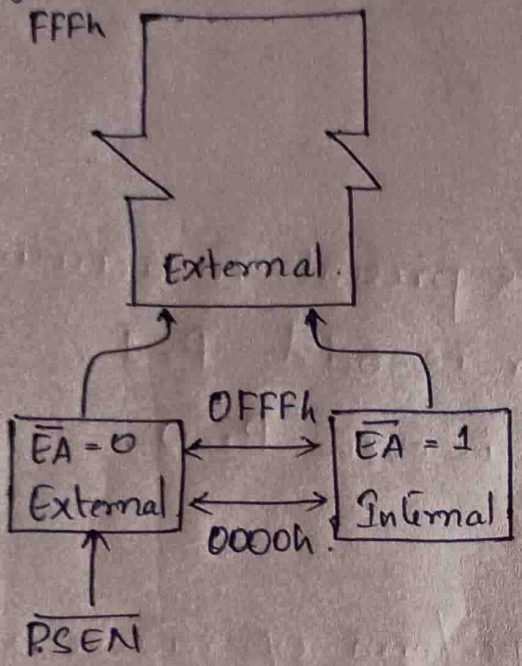
* Memory Organization:-
  - 8051 devices have seperate address space for program and data memory.
  - this allows the 8bit CPU to generate 8-bit address for accessing data.
- 8051 has a program memory which is 64 K byte long.
  Basic version of 8051 has a program memory of 4K bytes inside the chip.
- Internal memory of 8051 is implemented using ROM.
- Some version of 8051 does not have internal memory they are ROM less version.

In Such case all the program memory is implemented outside the chip.

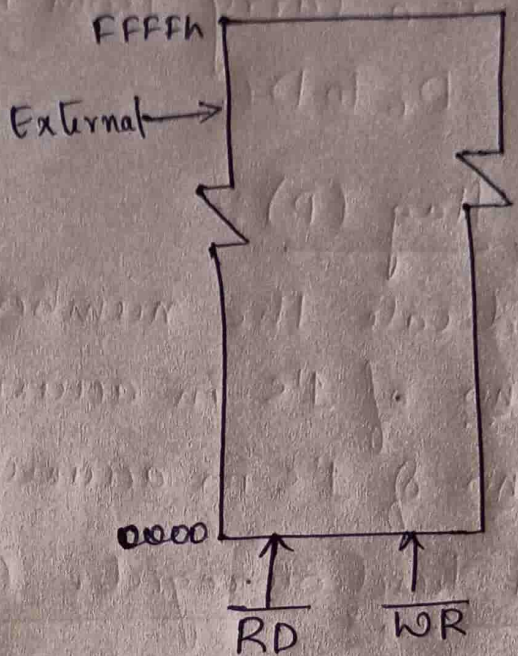
- the memory used to implement is EPROM (Electrically Programmable ROM.

Basic memory structure of 8051 is shown below.

Program memory (ROM)



Data memory (Read/Write)



* It can be implemented as combination of internal and External memory. or exclusively external memory.

Data memory and signal RD & WR are used to write and read data to and from the memory.

$\overline{EA}$ = 0 , 8051 can access external memory i.e from 0000h to FFFFh.

$\overline{EA}$ = 1, 8051 can access internal memory i.e from 0000h to OFFFFh.

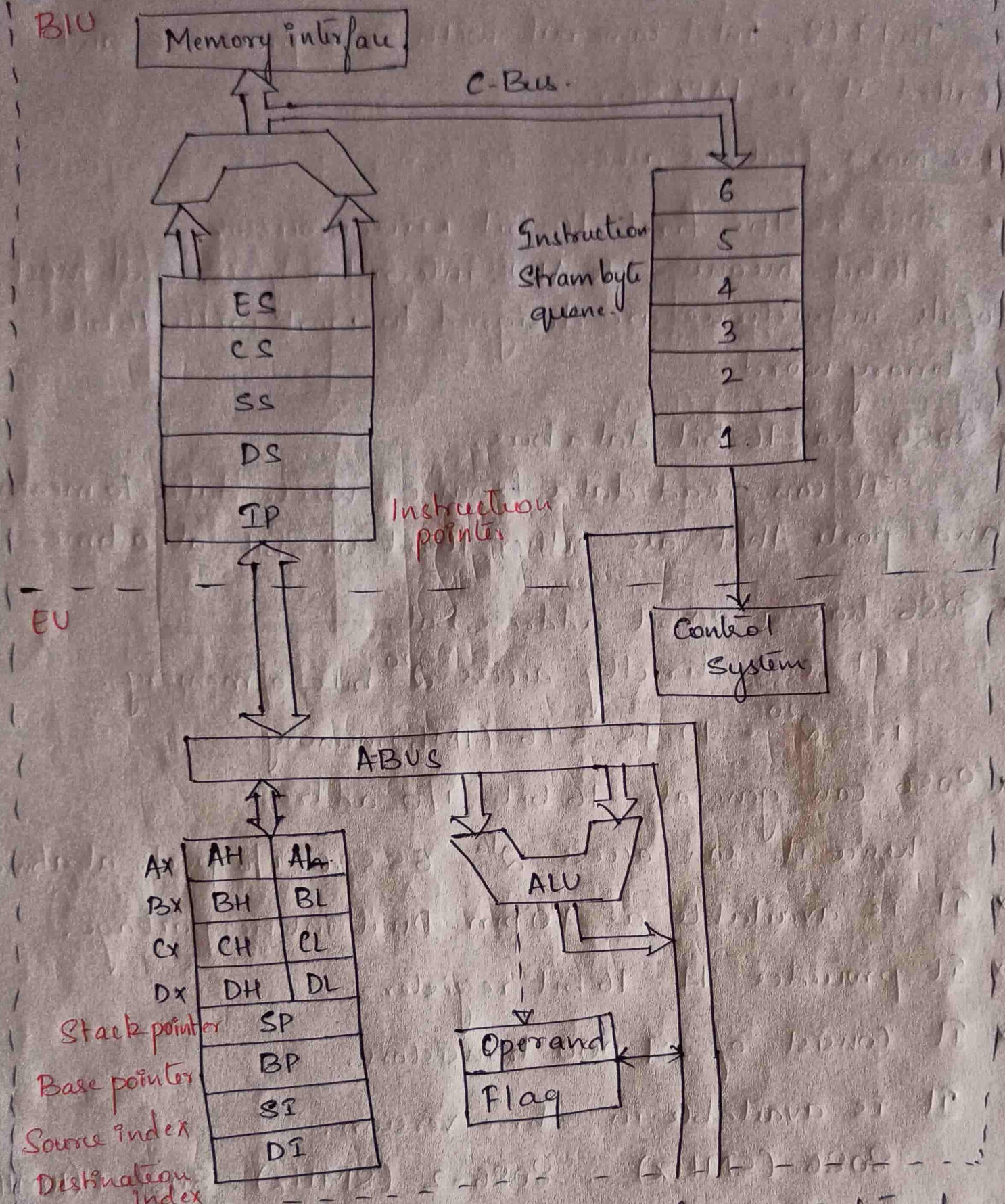
(two control pins PSEN and EA these determine whether External memory / internal memory is accessed

* Architecture of 8086:-

In 1978, Intel came out with 8086 processor and called it as iAPX 86.

the main features are:-

* 8086 is the first 16-bit microprocessor.
  16 bit means - its ALU, internal register and most of its instructions are designed to work with 16 bit binary word.
* 8086 has 16 bit data bus.
  It can read data from or write data to memory and ports that are either 16 bit or 8 bit at a time.
* 8086 has a 20 bit address but.
  It can directly access 2^{20} bits of memory.
  i.e 2^{20} = 10,48,576 (1Mb) memory.
* 8086 can generate 16 bit i/o address.
  hence it can access 2^{16} - 65536 i/o ports at a time
* It is available as 40 pin Dual inline package.
* It provides 14, 16 bit register.
* It consist of 29,000 transistors.
  It is available in three version.
  8086 (5MHz), 8086-2 (8MHz), 8086-1 (10MHz)
* It can perform bit, byte, word, block operation.



- figure shows the Block diagram of 8086 internal Architecture.
- It is divided into two seperate function unit
  - BIU (Bus interface Unit) } Work simultaneously to increase System Speed.
  - EU (Execution Unit

* Bus Interface Unit :-

- Used to interface to the Outside World.
- provides a full 16 bit bidirectional data bus and 20 bit address bus.
- Responsible for performing all external bus operation as listed below.

1. Sends address of memory or I/O port.
2. Fetches instruction from memory.
3. Reads data from port / memory.
4. Writes data from port / memory.
5. Supports instruction queuing.
6. provides address relocation facility.

* Execution Unit :-

- EU tell BIU from where the instruction has to be fetched / data to be fetched.
- decode the instruction and executes the instruction.
- It consist of Control Circuitry, Instruction decoder, ALU, flag register, General purpose register, Pointer

* Register Organization :-

It has 4 sets of register.

General Purpose

	15 8	7 0
AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

Segment register

CS
DS
ES
SS

Flag Register

Flag

Pointer & Index register

SP
BP
SI
DI
IP

* Execution Unit

- gives instruction to BIU starting from fetch the data and decode and execute those instruction.

functional parts of 8086 microprocessor.

1. ALU - handles all arithmetic and logical operation

2. Flag register - 16 bit register behaves as flip flop

* Changes its status according to result stored in accumulator.

* It has 9 flags and they are divided into 2 groups - Conditional flags and Control flags.

→ Conditional flags:-

- these flags represent the results of last arithmetic & logical instruction executed.

- It has 6 flags that are listed under it.

1. Carry flag: indicates an overflow condition for arithmetic operation.

2. Auxiliary flag: When an operation is performed at ALU, it results in a carry / borrow from lower nibble to upper nibble then this flag is set.

- the processor uses this flag to perform binary to BCD Conversion.

3. Parity flag: flag is used to indicate parity of result When 8bits of the result contain even no of 1's then Parity flag is set, for odd no of 1's then Parity flag is reset.

4. Zero flag:-

- flag set to 1 when the result of arithmetic or logical operation is zero. else flag is set to 0.

5. Sign flag:-

- this flag holds the sign of result.
- when result of arithmetic or logical operation is negative, then the sign flag is set to 1 else set to 0.

6. Overflow flag:-

Represents the result when the system capacity is exceeded.

→ Control Flag:-

- Controls the operation of execution unit
- 3 flags are listed under it.

1. Trap flag:-

- used for single step control and allows the user to execute one instruction at a time for debugging.
- If the flag is set, then the program can be run in a single step mode.

2. Interrupt flag:-

- It is an interrupt enable / disable flag.
- Used to allow / prohibit the interruption of a program.
- It is set to 1 for interrupt enable condition.
  set to 0 for interrupt disable condition.

3. Direction flag:

- It is used in string operation.
- It is set to 1 when string bytes are accessed from higher memory address to lower memory address and vice-versa.

* General Purpose register:-

there are 8 general purpose register

- AH, AL, BH, BL, CH, CL, DH, DL → these can be used individually to store 8bit data or can be used in pairs to store 16 bit data.
- A valid register pairs are represented as AX, BX, CX, DX

1. AX register - Accumulator register

they are used to store operads for arithmetic operation.

2. BX register - Base register

Used to store starting base address of the memory area within the data segment.

3. CX register - Counter

Used to store / Used in loop instruction to store loop Counter.

4. DX register - Used to hold i/o port address for i/o instruction.

* Stack pointer register :-

- Its a 16 bit register
- holds the address from the start of Segment to the memory location, where a word was most recently stored on the stack.

* Bus interface Unit [BIU] :-

- It takes care of all data and addresses transfer on the Buses for the EU like sending address, fetching instruction from the memory, read data from the ports and the memory as well as writing data to the ports and memory.

→ Functional Parts

1. Instruction queue :-

- BIU gets upto 6 bytes of next instructions and stores them in instruction queue.
- EU executes instructions and is ready for its next instruction, then it simply reads the instruction for the queue and hence increase the speed of execution.

2. Segment register :

- BIU has 4 segment buses CS, DS, SS & ES.
- It holds the address of instruction and data in memory, which are used by the processor to access memory location.
- It contains 1 pointer register i.e IP, which holds the address of next instruction to executed by EU.

1. CS – Code Segment:

Used for addressing a memory location in the code Segment of the memory, where executable program is stored.

2. DS – Data Segment:

It Consist of Data used by the Program and is accessed in the data Segment by an offset address or the Content of other register that holds the offset address.

3. SS – stack Segment:

It handles memory to store data and address during execution.

4. ES - Extra Segment:

It is a additional data Segment, which is used by the String to hold the extra destination data.

5. Instruction Pointer:-

16 bit register used to hold the address of the next instruction to be executed.

* Addressing modes of 8086

- the way in which a Source operand is denoted in an instruction is known as addressing mode.
- there are 8 different addressing mode in 8086 prog".

1. Immediate addressing mode :-

the addressing mode in which the data operand is a part of instruction itself.

Eg MOV CX, 4929h. CX → dst, 4929h - Src.

ADD AX, 2387h. ; AX + 2387h ; dst + Src.

MOV AL, FFh. AL → dst. FFh - Src.

2. Register addressing mode :-

the addressing mode in which register is the Source of an operand for an instruction.

Eg MOV Cx, Ax ; Copy the Content of Ax to Cx reg.

ADD Bx, Ax ; Add the Content of Ax to Content of Bx.

Register indirect

3. ~~Direct~~ addressing mode :-

It allows data to be addressed at any memory location through an offset address held in any of the following register. i.e Ax, Bx, Cx, Dx.

Eg Mov Ax, [Bx] ; Suppose Bx Contains 4895h.
then Contents present in 4895h location will be moved to Ax. register.

ADD Cx, {Bx}

4. Based addressing mode :-

the offset address ~~of~~ of the operand is given by the Sum of Contents of the Bx / BP register and 8bit / 16bit displacement.

Eg Mov Dx, [Bx+04] ; Suppose Bx = 4895h + 04h.
= 4899h → offset address
the Content of 4899h will be move to Dx register.

5. Indexed addressing mode :-

the operands offset address is found by adding the content of SI or DI register and 8bit / 16bit displacement.

Eg MOV Bx, [SI+16] ; SI = 4989h.

[SI+16] = 4989+16

= 5005h.

SI = 5005h content of SI will be moved to Bx register.

6. Based-index addressing mode :-

the offset address of the operand is computed by summing the base register to the contents of an Index register.

Eg Add Cx, [Ax+SI] ; Ax → Base register. SI - Index reg.

Ax = 1200h SI = 1400h.

[AX+SI] = 1200h + 1400h

= 2600h → offset address

Content of 2600h will be added to content of Cx

7. Based indexed with displacement mode :-

the operands offset is computed by adding the base register content, Index register content and 8 bit / 16 bit displacement.

Eg MOV Ax, [Bx+DI+08] ; Bx = 25h. , DI = 25h., 08

[Bx+DI+08] = 25+25+08

= 58h.

Content of 58h will be Copied to Ax Base register.

8. ~~Register~~ direct addressing mode:-

the addressing mode in which the effective address of memory location is written directly in the inst".

eg MOV Ax, [1592h]

MOV AL, [0300h]