

A CPU built into a single LSI chip is 4P

4 bit 4P → That makes processing of 4 bits in parallel at time

|          |     |
|----------|-----|
| PAGE NO. |     |
| DATE:    | / / |

## MICROPROCESSOR

A microprocessor is a computer processor where data processing logic & control is included on a single integrated circuit. The microprocessor contains arithmetic logic and control circuits to perform functions of a computer's central processing unit.

The first microprocessor was Intel 4004, a 4-bit PMOS microprocessor was introduced in 1971 by Intel Corporation.

8-bit 4P → 8008 4P which was introduced/invented in 1972.

This also uses PMOS technology 16 KB → Memory Address Capacity

In 1973, Intel introduced a more powerful and faster 8-bit NMOS microprocessor 8080. 64 KB.

In 1976, 8085 microprocessor was invented which is 8-bit microprocessor. It is an improved version of 8080 4P. 8085 4P uses only one +5V supply. 64 KB.

In 1978, Intel introduced 16-bit 4P that is 8086. The 16-bit 4P have more powerful instruction sets than that of 8-bit 4P. They uses VLSI technology (very large scale integration - VLSI). They can directly address memory in the range of 1MB to 16 MB. HMOS 4P / 5V dc supply for operation.

$$1\text{MB} = 10^6 \text{ bytes}$$

$$1\text{KB} = 10^3 \text{ bytes}$$

## # 8085 microprocessor

Intel 8085 is 8-bit NMOS uP. It is 40 pin IC. Using a single supplies +5V for suppression and clock speed 3 MHz. It has 80 basic instructions and 240 opcodes. Important sections of 8085 is

- ① ALU → Arithmetic Logic Unit. It performs all the arithmetic and logical operations like addition, increment, rotate-left, rotate-right, logical XOR, etc.
- ② Timing & Control Unit is a control unit of CPU. It generates timings and control signals which are necessary for execution of instructions. It controls data flow b/w CPU and peripheral devices / memory devices. provide status, control & timing s/g which are required for operation
- ③ Registers are used by uP for temporary storage of data and manipulations of data and instructions. In large computers, no. of register is more and hence the program requires less transfer of data to / from the memory. In small computer, no. of register is small due to the limited size of chip.

*This section is about how data is transferred between memory and CPU.*

So 8085 consists of one 8-bit register that is accumulator register (A), six 8-bit General purpose register (B, C, D, E, H & L), one 16-bit stack pointer, one 16-bit program counter, Instruction Register

- ACCUMULATOR

Accumulator is a 8-bit register. It is used to hold one of the operands of arithmetic or logical operations. For eg: there are some logical instructions which need only one operand so it is held in accumulator and by default all the results are stored in accumulator. This holds operands which serves as one of input of ALU.

ADD A, B



A  $\leftarrow$  A + B

→ will be stored in A

#### • GENERAL PURPOSE REGISTER

8085 consists of 6 general purpose registers (B, C, D, E, H & L).

The combinations of two 8-bit register is known as register pair. The valid register pair for 8085 are BC, DE and HL. So to hold 16 bit data, we will use these register pair. HL is used to address memories.

CPU fetches  
instruction  
from memory  
& executes  
it  
implement  
the content  
in P.C.

#### PROGRAM COUNTER

Program counter is a 16-bit register. It is used to hold the address of next instruction to be executed.

ADD A, B      Agar ADD wala field to stack in toh wo

SUB A, D      next instruction to stack a.

#### • STACK POINTER

During the execution of program, sometimes it becomes necessary to save content of some register which are needed for other operations in the subsequent steps of program.

The content of such registers are saved in stack then the register are used for some other operations. So the stack pointer controls the addressing of stack.

#### • INSTRUCTION REGISTER

Instruction registers hold the opcodes of the instructions which is being decoded & executed.

Cannot be accessed by programmes.

- FLAGS

It indicates the status of the results.

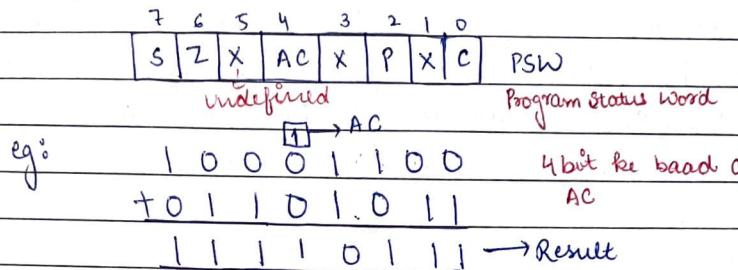
Carry

AC (Auxiliary Carry)

zero

Parity  $\xrightarrow{\text{Odd}}$  Even

sign bit  $\rightarrow$  MSB 1, then it will set to 1



$$\text{Carry flag (C)} = 0$$

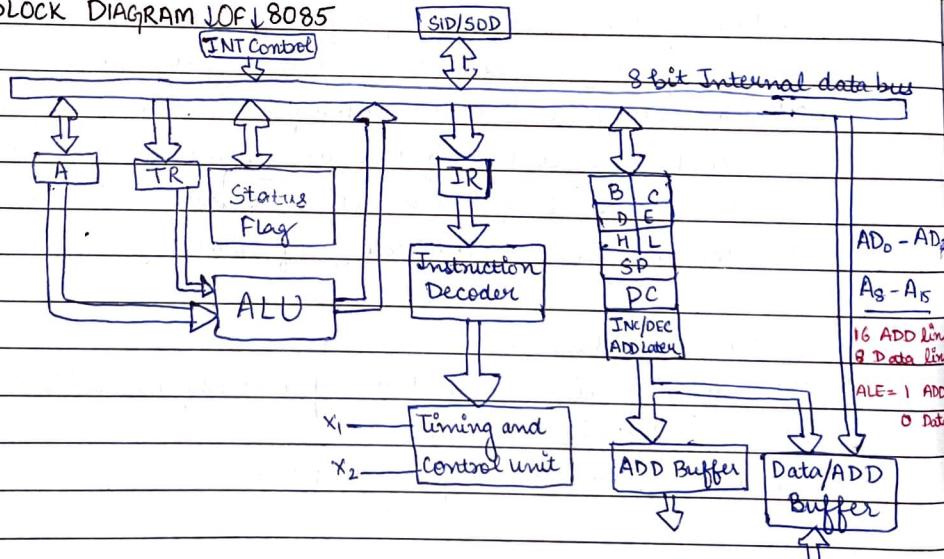
$$\text{Parity (Odd)} = 0 \quad \text{If even toh 1}$$

$$AC = 1$$

$$\text{Zero} = 0 \quad \text{Jb result 0}$$

$$\text{Sign bit} = 0 \quad \text{If -ve toh 1}$$

## # BLOCK DIAGRAM OF 8085



TR - Temporary Register

A - Acc Reg

IR - Instruction Reg

SID - Serial Input Data

SOD - Serial Output Data

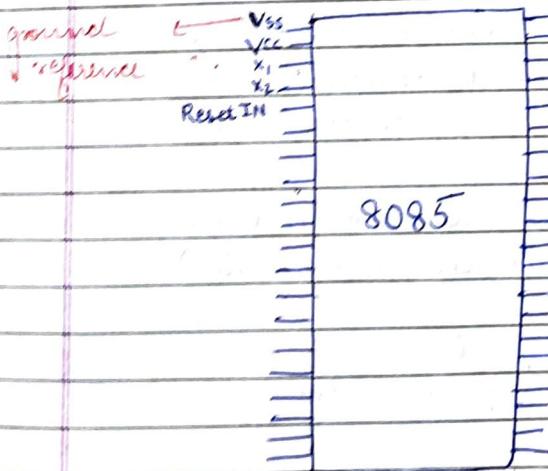
## INSTRUCTION CYCLE

An instruction is a command given to the computer to perform a specified operation on a given data. To perform a particular task, a programmer writes a sequence of steps called as a program and program and data are stored in memory. The CPU fetches one instruction from the memory at a time and execute it. So an instruction cycle consists of fetch cycle and execute cycle. (Necessary steps CPU carries out to fetch an instruction & necessary data from memory to execute it.)

$$IC = FC + EC$$

(Fetches fixed block of time required to fetch)

## PIN CONFIGURATION / DESCRIPTION OF 8085



8 bit data bus

Intel 8085 requires 16 bit wide address bus as.

memory address are of 8 bits.

Most significant 8 address are transmitted by address bus, A bus (pin 3 to 15)

Least significant 3 bit - ADO-A2

A<sub>0</sub>-A<sub>15</sub>: These are the address buses and are used for most significant bit of the memory address. This is output pin

Since pins 16 bit memory address transmit data in by u/p. Then In 8 MSB's & in 8 LSBs. Total effective width of address is 16-bit wide. The 8 bit LSB goes to ~~no~~ external latch no data in memory main. An external latch takes 16 bit address is available for further operation.

PAGE NO.

DATE: / /

$A_{D_0} - A_{D_7}$ : These are input and output pin. They are time multiplexed address and data bus. For the first clock cycle, these lines act as a address bus and they are used as data bus during second and third clock signal. It operates in time shared mode. This mode is known as multiplexing.

Address  
Latch Enable

$ALE$ : This is output pin. It goes high during first clock cycle and enables lower 8 bit address bus to be latched either to memory or external latch.

$I/O \bar{m}$ : This is output pin. It is a status signal which indicates whether the address is for memory or input-output devices. When it goes high it is for  $I/O$  & when low then for memory.

$S_0 S_1$ : This is output pin. Status sig sent by u/p to distinguish.

| $S_1$ | $S_0$ | operation | various type of operation |
|-------|-------|-----------|---------------------------|
| 0     | 0     | Halt      |                           |
| 0     | 1     | WRITE     |                           |
| 1     | 0     | READ      |                           |
| 1     | 1     | fetch     |                           |

when low,  
Selected  
memory or  
 $I/O$  to  
read

**READ**: This is active-low pin. When microprocessor read data or code from memory location or input-output devices then that is READ operation. ( $\overline{RD}$ ). Signal to control Read operation

when low,  
Selected  
data bus  
if writing  
to selected  
 $I/O$  or memory

**WRITE**: When microprocessor sends data to memory location or output devices, it is called WRITE operation. ( $\overline{WR}$ ) Control write operation

**Ready**: It is input pin. It is used by u/p to sense whether the peripheral device are ready for data transfer or not. If low then ready.

**Clk**: It is output pin.

It is clock output for user, which can be used for other digital IC's.

If low then it waits till it gets ready.

## \* 4 interrupt in 8085

↳ 8-bit data bus.

- HOLD - Indicates that another device is requesting the use of address & data bus.
- HLDA - Hold Acknowledgement. Indicates that hold request is received. When HOLD remains, HLDA goes low.

PAGE NO.

DATE: / /

TRAP - 6<sup>th</sup> pin

RST 5.5 - 7<sup>th</sup> pin

RST 6.5 - 8<sup>th</sup> pin

RST 7.5 - 9<sup>th</sup> pin.

INTR → Interrupt Request.  
when it goes high, PC  
does not increment its content.  
Input pin.

JNTA (output) - Acknowledgment  
sent by ~~high~~ up when INTR is  
received.

This is basically the interrupt. Interrupt is a signal to the processor generated by hardware and software integrating and immediate action mediate by some event.

Non Maskable — TRAP ↑ → highest priority → It is unaffected by any mask or interrupt.

Restart { RST 7.5

Interrupt { RST 6.5

↓ RST 5.5 ↓ → least priority

↳ Cause internal restart to be automatically initiated.

decreases

Vector Address :

0024H - TRAP

003CH - RST 7.5

0034H - RST 6.5

002CH - RST 5.5

Jb interrupt recognise kya jaata h, toh next instruction is executed from fixed location in memory.

NMI: Non Maskable Interrupt. We cannot disable this interrupt.

TRAP - NMI

RST 7.5 - MI

RST 6.5 - MI

RST 5.5 - MI

DI instruction for disable & EI for enable.

SID/SOD : SID is 5<sup>th</sup> pin and SOD is fourth pin.

HOLD - is requesting the use of address and data bus. 39<sup>th</sup> pin.  
HLDA This is input pin.

HLDA - 38<sup>th</sup> pin. This is output pin.

Reset IN (Input) - Reset the PC to 0.

" OUT (O/P) - CPU is being reset.

OpCodes - specifies the task to be performed by computer.

PAGE NO. \_\_\_\_\_  
DATE: / /

INTR - Active high. Input pin

INTA - Active-low. Output pin

## # 8086 Architecture.

~~20-bit address bus~~  
It is a 16 bit UP. It has databus of 16 bit. It has 2 unit BIU (Bus Interface Unit) and EU (Execution Unit). It has four general purpose register : AX, BX, CX, DX. (16 bit)

Directly  
address  
 $2^{20} = 1\text{MB}$   
for memory  
address

AX → serves as accumulator  
AL → serves as general purpose register  
8 most significant bit → 8 least significant bit  
Data temporary when store long a job operations perform longer.

Source Index - SI } store - Used in  
Destination Index - DI } offset - memory  
Stack Pointer - SP } Address - address  
Base Pointer - BP } - computation

## 20 Address Bus

which operate in multiplexed mode.  
16 low order address bus line  
are multiplexed with data &  
4 high order address bus.  
lines are multiplexed with  
status signal.

We have four segments:

CS → Core Segment → initial address of core segment.

SS → Stack Segment

DS → Data Segment

ES → Extra Segment.

Instruction Pointer → Store address of next address to be executed.

In this block-diagram, CS, SS, DS & ES will store the initial segment of memory.

Base segment 16 address } Except AH, BH, CH, DH  
 ya offset address }

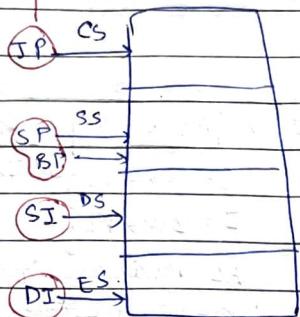
Address Bus: send address

Data bus: To data area in memory through.

PAGE NO. / /  
 DATE: / /

885 AD0 AD7  
 PS AL5

→ offset for corresponding set of memory



ADD. Bus<sup>20</sup>  
 Data bus<sup>16</sup>  
 multiplex ADD. AD<sub>5</sub>  
 AD<sub>7</sub> AD<sub>9</sub>

Let us suppose CS has address (initial) 1000H.

offset address IP → 2486H

Physical / Real / Effective Address - ?

multiply initial address by 10 & add offset

$$\begin{aligned} \text{Real address} &= 1000 \times 10 + 2486 \\ &= 12486 \end{aligned}$$

Physical address will go to memory interface & fetch data.

In this when a block will be empty then it will fetch.

Here fetch & execution simultaneously take place so the time gets saved. This is called pipelining architecture.

1

2 — Jump

3

4

5

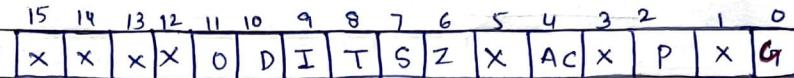
6

7

directly goes to 7 because both satisfy in 1st slot.  
 operation of pipelining will fail.

Decode logic to logic control system.

## # Flag Register of 8086



O = Overflow flag

S = Sign flag  $\rightarrow$  Jb -ve no's set

D = Direction flag

Z = Zero flag

I = Interrupt flag

AC = Auxiliary Carry

T = Trap flag

P = Parity  $\rightarrow$  for even no. of one's.

X = Unused flag

C = Carry parity will set

**Overflow flag:** Overflow flag will set if overflow occurs if the result of side operation is large enough to accomodate in a destination register.

$$\begin{array}{r} \text{Signed No.'s:} \quad 11\ 00 \\ \quad + 10\ 00 \\ \hline \quad 101\ 00 \end{array}$$

If signed no. two are operation perform karna ke baad uska sign ball jaye toh overflow flag will set.

**Direction flag:** It is used by string manipulation instructions.

If this flag bit is zero, then the string is processed from the lowest address to highest address i.e. auto-increment mode.

If this flag bit is one, then the string is processed from the higher address to lower address i.e. auto-decrement mode.

**Interrupt flag:** If this flag is set, then maskable interrupt are recognised by CPU.

**Trap flag:** If this flag is set then the processor enters the single step execution mode.

1 MB memory can be accessed  
by microprocessor

AD<sub>0</sub>-AD<sub>15</sub> - Bi-directional • Low-order address bus  
multiplexed with data.

PAGE NO.  
DATE: / /

## # Pin Configuration of 8086

40 pin IC

- minimum mode
- maximum mode

AD<sub>16</sub>-AD<sub>19</sub> - High order address bus  
multiplexed with status signal

The 8086 operates in single processor or multi-processor configuration to achieve high performance. So if it works as single processor then minimum mode & if we are operating as multi-processor then maximum mode.

### 1. GND (Ground Pin)

2. From 2<sup>nd</sup> to 16<sup>th</sup> we have AD<sub>14</sub> - AD<sub>0</sub> {2-AD<sub>14</sub> & 16-AD<sub>0</sub>}

These are time multiplexed ~~memory~~ address and data bus because same bus can be used as address bus and at other instant it can be used as data bus.

3. Pin no. - 39 AD<sub>15</sub>. It is also time multiplexed address and data bus.

4. 38 A<sub>16</sub> / S<sub>3</sub>                                    S for status

37 A<sub>17</sub> / S<sub>4</sub>                                    S<sub>4</sub> is always low

36 A<sub>18</sub> / S<sub>5</sub>                                    S<sub>5</sub> → The status of Interrupt flag is

35 A<sub>19</sub> / S<sub>6</sub>                                    displayed on S<sub>5</sub>

D<sub>0</sub> - D<sub>15</sub> = 16 data bus

A<sub>0</sub> - A<sub>19</sub> = 20 address bus

| S <sub>4</sub> | S <sub>3</sub> | Indication     |
|----------------|----------------|----------------|
| 0              | 0              | Alternate Data |
| 0              | 1              | Stack          |
| 1              | 0              | Code           |
| 1              | 1              | Data           |

During memory for input or output operation, status information is available.

### 5. BHE / S7: Not in use.

Enable

data into

MSB half

of data bus

D8-D15

8 bit device

connected to upper

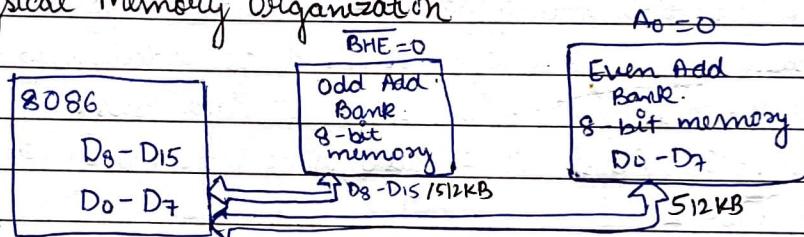
half of data bus

use BHE signal.

→ Bus High Enable (24<sup>th</sup> Pin) multiplexed with status signal S7.  
we ~~can't~~ use it with conjunction of A0

| BHE | A0 | Indication                        |
|-----|----|-----------------------------------|
| 0   | 0  | whole word                        |
| 0   | 1  | Upper byte from 0 to odd address  |
| 1   | 0  | lower byte from 0 to even Address |
| 1   | 1  | None                              |

### \* Physical Memory Organization



In 8086 based systems, 1MB memory is physically organised as an odd bank and even bank each of 512KB addressed in parallel by the processor. Byte data with even address is transferred on D0-D7 and byte data with odd ~~state~~ address is transferred on D8-D15.

6. RD

7. Ready -

8. Reset

9. INTR / INTA

10. TEST

11. NMI

|            |                         |                                |
|------------|-------------------------|--------------------------------|
| 12. CLOCK. | $\overline{WR}$ - Write | DEN - Data Enable              |
| 13. Vcc    | Hold / HLDA             | DT/R - Data Transmit / Receive |
| 14. GND    | ALE                     |                                |

# Ready - This pin is acknowledgement from slow devices or memory that they have completed data transfer.

# Reset - This input causes the processor to terminate the current activity.

### Interrupt Request (INTR)

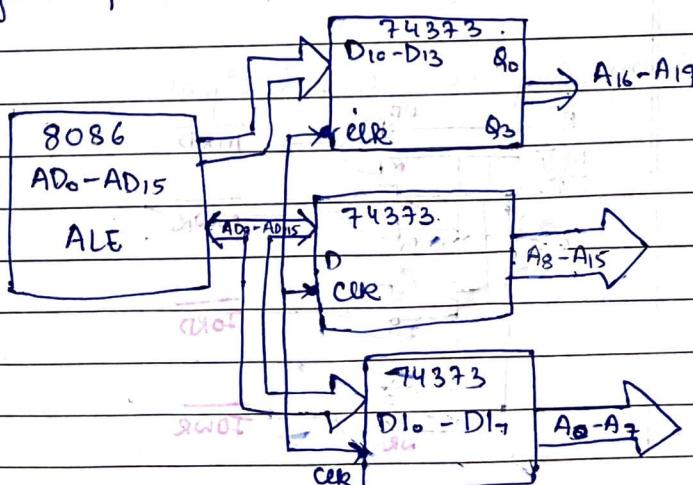
# This input signal is examined by TEST instruction. If test input goes low execution will continue otherwise remains in Idle state - TEST

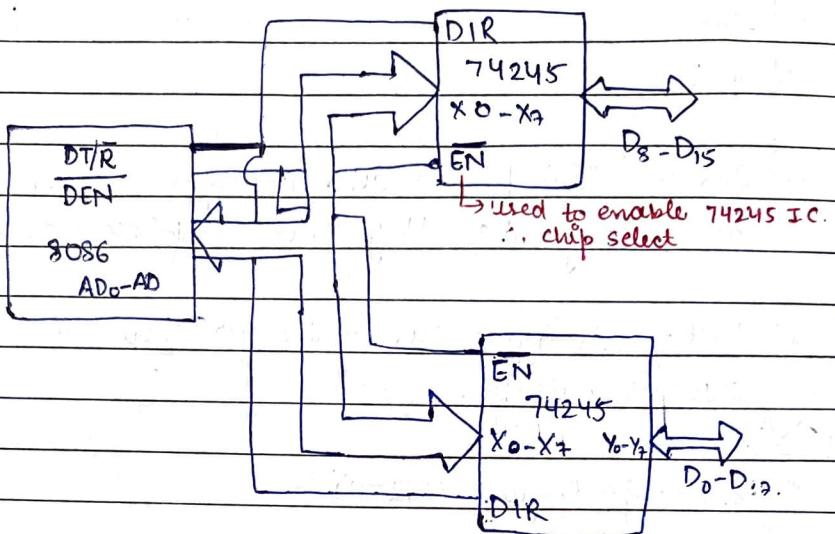
### Deriving System Bus

When ALE high, then address bus; if low then data bus. 8086 has a multiplexed 16 bit AD<sub>0</sub>-AD<sub>15</sub> and a multiplexed 4 bit address and status bus (A<sub>16</sub>/S<sub>13</sub>-A<sub>9</sub>/S<sub>6</sub>). The address can be latched using ALE pin, and a circuit for this pin is

IC-74373

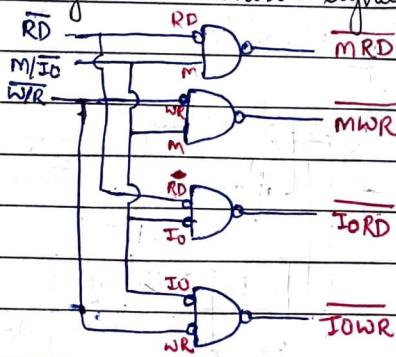
When demultiplexed data bus, we will only get output as address bus.





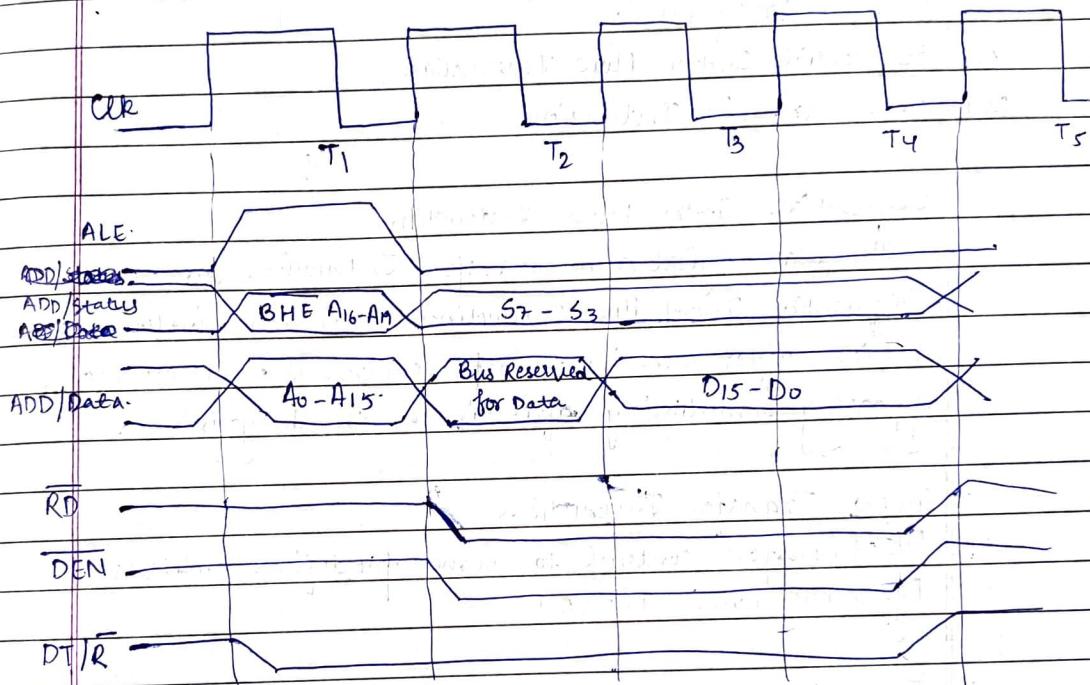
$\overline{DEN}$  and  $DT/R$  indicate the presence of data on the data bus and the direction of data, that is to / from the microprocessor and they are used to derive the chip select and direction pins of the buffers as indicated in fig. If  $\overline{DEN}$  is low, it indicates that the data is available on the multiplexed bus and both the buffers i.e. 7425 IC are enabled to transfer the data and when  $DIR$  pin goes high, the data available on  $X$  pin of 74245 are transferred to  $Y$  pin.

### # Deriving 8086 control signal



## \* Timing Diagram for Minimum Mode

### Read Cycle Timing Diagram for Minimum Mode



### WRITE Cycle Timing Diagram

- (1) CLR sig → Reference
- (2) Send Address → So it should be 1
- (3) ADD/status
- (4) ADD/data
- (5) WR
- (6) DEN
- (7) DT/R → opp of Read Cycle

## II ADDRESSING MODES OF 8086

Addressing Modes Indicate / describes the ~~area~~ types of operands and the way they are accessed for executing an instruction. According to the flow of instruction execution, the instructions may be categorized as

- (i) Sequential Control Flow Instructions
- (ii) Control Transfer Instructions

### Sequential Control Flow Instructions

Arithmetical Logical Data Transfer Instructions are eg of. Sequential Control Flow Instructions. These are instructions in which after execution transfers control to next instruction appearing immediately after it in the program.

### Control Transfer Instructions

It transfers control to some predefined address.

Eg- JUMP, CALL, RETURN

#### (i) Immediate Addressing Mode

In this addressing mode, immediate data is present in the instruction. for eg:- MOV AX, 5000H

#### (ii) Direct Addressing Mode

In this mode, a 16-bit memory address is directly given, for eg: MOV AX, [5000H]

#### (iii) Register Addressing Mode

In this mode, data is stored in register. All the register except IP may be used in this mode

Eg: AX, BX

(iv) Register Indirect Addressing Mode

In this mode, address of the memory locations which contains the data is find out in an indirect way using offset registers.

Eg: AX, [BX]

[BX], SI or DI for offset address.

(v) Index Addressing Mode

In this mode, offset of data/ operand is stored in one of the index register and DS is the default segment for Index register i.e. SI and DI

Eg - MOV AX, [SI]

Here the data is available at an offset address stored in SI in DS.

$$\text{Effective address} = DS \times 10 + SI$$

(vi) Register Relative Addressing Mode

In this mode, data is available at an effective address formed by adding 8-bit or 16-bit displacement with the content of any one of the register that is BX, BP, SI and DI in four segment. Eg: MOV AX, 50H[BX]

$$EA = 10 * DS + 50H + [BX]$$

(vii) Based Indexed Addressing Mode

The EA of the data is formed in this addressing mode by adding the content of base register to the content of index register

Ex: MOV AX, [BX][SI]

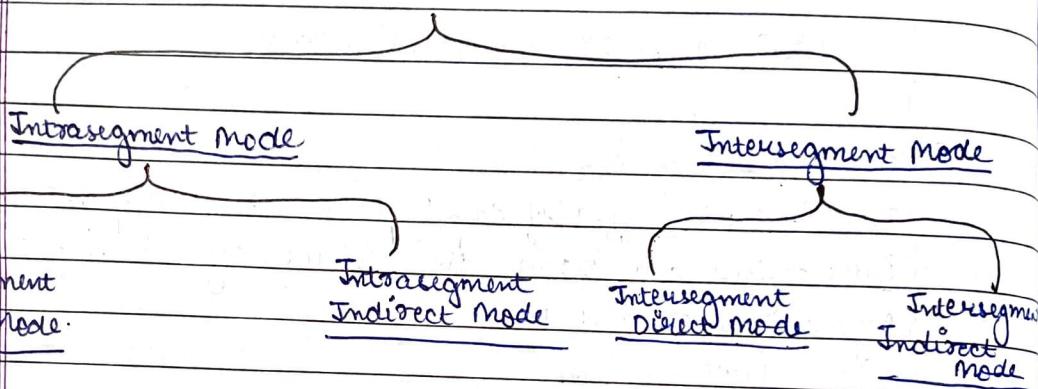
$$EA = DS \times 10 + [BX] + [SI]$$

### (viii) Relative Based Indexed Addressing Mode

Eg:  $MOV AX, 50[BX][SI]$

$$EA = DS * 10 + 50 + [BX] + [SI]$$

There are two addressing modes for Control Transfer Instructions:

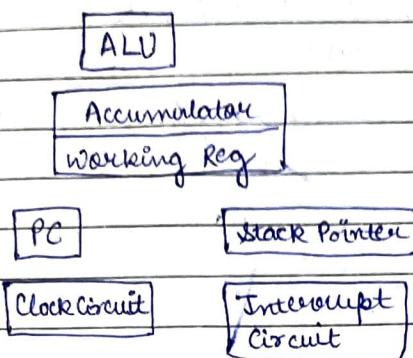


If the location to which the control is transferred lies in different segment other than the current one, the mode is called intersegment mode. If the destination location lies in same segment then it is called Intrasegment Mode.

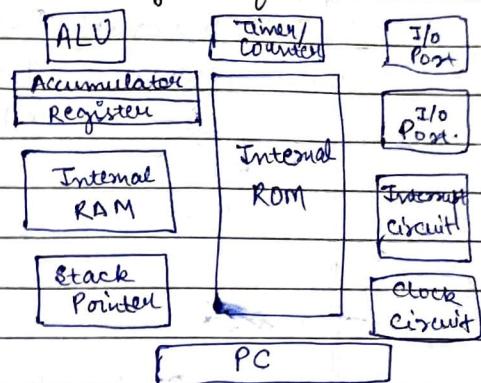
#### Intrasegment Direct Mode

In this mode, the address to which the control is to be transferred lies in same segments and appear directly in the direction as an immediate displacement value.

Block Diagram of Microprocessor



Block Diagram of Microcontroller



Microcontroller is true computer on a chip. The design incorporates all the features found in a CPU i.e. ALU, PC, SP, Registers and also some additional feature to make a complete computer i.e. ROM, RAM, Parallel I/O, Serial I/O, Counters and stack and clock circuit.

### # 8051 Microcontroller

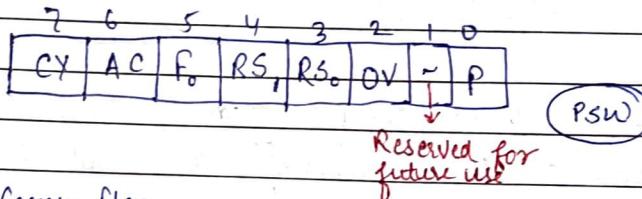
8051 is 40 pin IC. It has Internal ROM and RAM, Timer and Counters, Serial Data Communication, ALU, PC, Registers and Clock Circuits.

8051 Architecture consist of these specific features:

- Eight bit CPU with Register A, B
- 16 bit Program Counter (PC) & DPTR (Data Pointer)
- 8 bit PSW.
- 8 bit SP
- Internal ROM/EPRAM : 4K byte
- Internal RAM of 128 byte
- Few Reg Bank, each contains 8 Registers
- 16 byte, which may be addressable at bit level.
- 8 byte of general purpose data memory.
- 32 I/O Pin (4 <sup>Port</sup> & 80/P)

- Full Duplex Serial Data Receiver / Transmitter
- Control Register TCON, TMOD, SCON, PCON / SBUF, IP & IC
- Two external + 3 Internal Interrupt
- Oscillator and Clock circuit
  - ↓
  - 12MHz

# PSW :



CY : Carry flag.

- It is used in arithmetic, jump,rotate and boolean instructions.

AC : Auxiliary Carry

- Used in BCD Addition / Subtraction.

F<sub>0</sub> is user-defined flag.

RS<sub>1</sub> / RS<sub>0</sub> → to select any bank

RS<sub>1</sub>      RS<sub>0</sub>      Function.

0      0      Select bank B<sub>0</sub>

0      1      Select bank B<sub>1</sub>

1      0      B<sub>2</sub>

1      1      B<sub>3</sub>

OV → Overflow Flag

$P \rightarrow$  Parity flag

→ will set when it will have odd parity (odd no. of 1)

## # A and B Register

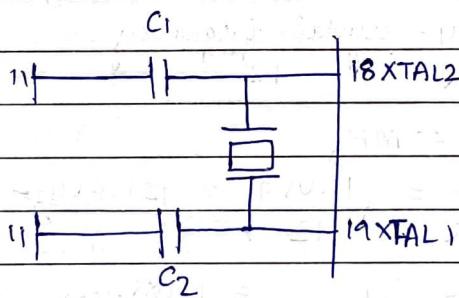
These are associated with processing unit. i.e. CPU Register 8051 contains 34 general purpose registers. Two of these registers A and B which hold result of many instructions particularly logical and math operations of 8051 CPU. The other 32 registers are arranged as a part of internal RAM in 4 banks i.e.  $B_0 - B_3$ .

A stores result of all arithmetic operations.

Eg: MUL B

$$A \leftarrow A * B$$

## # 8051 Oscillator and Clock Circuit



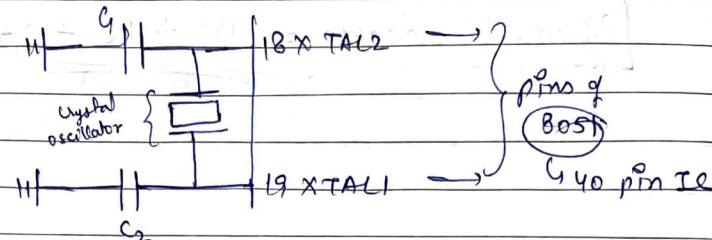
The heart of 8051 is the circuit that generates the clock pulses by which all internal operations are synchronized.

The manufacturers make available 8051 design that can run at speed max. and min. frequencies 1MHz to 16MHz. The crystal frequency is basically internal block frequency of the microcontroller. Ceramic resonators or quartz crystal can be used in this circuit.

25/4/22

MM

8051 oscillator &amp; clock circuit



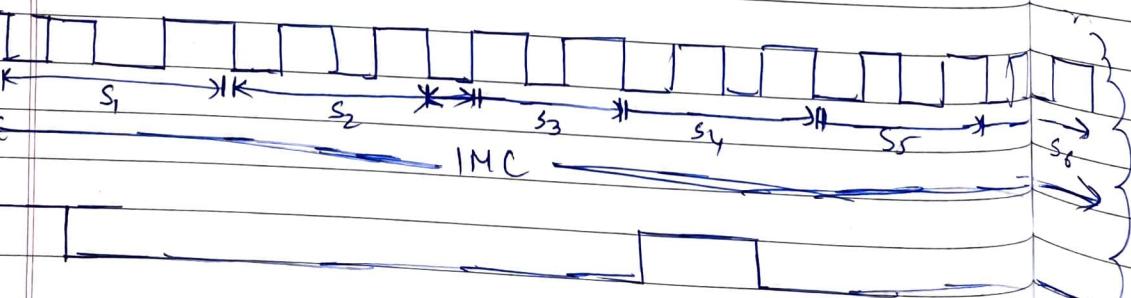
The heart of 8051 is the circuit that generates the clock pulses by which all internal operations are synchronized. The manufacturer make available 8051 design that can ~~not~~ run at speed maximum & minimum frequencies  $1\text{MHz}$  to  $16\text{MHz}$ .

\* Variation of frequency on speed : The crystal frequency is basically internal clock frequency of the microcontroller. Ceramic resonators/may be used in this circuit.



$1 \text{ machine cycle} = 12 \text{ clock pulse}$

Clk



ALE

\* To find machine cycle when

a) XTAL = 11.059 MHz

b) XTAL = 16 MHz

$$\text{clk frequency} = \frac{\text{crystal frequency}}{12}$$

a) CF =  $\frac{11.059}{12} = 921.6 \text{ KHz}$

b)  $\frac{16}{12} = 1333.3 \text{ KHz} = 1.33 \text{ MHz}$

C clock period =  $\frac{1}{1.33} = 0.75 \text{ usec.}$

clock period =  $\frac{1}{921.6} = 1.085 \text{ usec.}$

\* Program counter and Dptr :

(PC)

D data pointer → indicate data address.

These are two 16 bit register in 8051, The PC is basically used to hold the address of a byte in a memory.

\* Memory organization of 8051

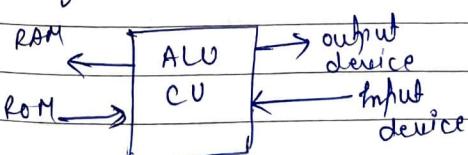
→ Data memory

RAM → 128 byte

ROM → 4 KB

(only read)  
program memory

8051 microcontroller implement Harvard architecture.  
Program & data are stored in different memory



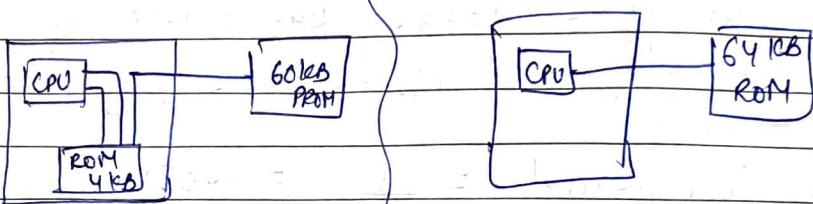
## ROM

Extended upto 64 kbyte.

Ways to extend

Internal  
ROM

External  
ROM



4 kB + 60 kB

EA (External access pin) = +5V

64 kB → external pin

EA = grounded (GND)

## RAM → Data memory

General  
purpose  
Registers

special  
function  
Registers

Take 168 Byte  
RAM

Take 168 byte  
RAM

Total 32 CPR  
in 8051

