~~Module Notes~~

8086

Basic features
- Architecture
- pin diagram
- working

☆ Basic features :-

① It is a 16 bit microprocessor implemented in n-channel depletion mode. H-MOS technology (High speed-metal oXide semi-cond).

② It is available in 40 pin IC package.

③ It has 16 bit ALU, 16 bit internal registers and instructions are designed to operate on 16 bit data.

④ It has 20 bit address bus that can access 2^{20} memory location = 1 MB

⑤ It can generate 16 bit I/o address

⑥ It has ~~16~~ 14 - 16 bit registers

(7) It supports the concept of memory segmentation.

(8) It has internal clock generator with 33% duty cycle.

8086 → 5 MHz

8086-2 → 8 MHz

8086-1 → 10 MHz

(9) It operates in two modes that is minimum and maximum mode. Minimum mode is used for small system while maximum mode is used for large system. that is multiprocessor.

(10) It supports multiprogramming that is execute in time-multiplexed mode.

(11) It fetches 6 instruction byte from memory and stores in queue to increase the speed of the execution. That it uses pipelining concept.

(12) It has powerful instruction set with multiplication and division. It also has different addressing mode like implicit, direct, indirect, register, immediate, indirect through the index & base registers, indirect through the sum of base, index and relative field.

Register Organization

① General purpose

AX	A_H	A_L
BX	B_H	B_L
CX	C_H	C_L
DX	D_H	D_L

② Segment register

CS
DS
ES
SS

16 KB.

③ Index & pointer

BP
SP
IP
SI
DI

④ Flag register

F

① Bx is used for offset storage for generating physical address in case of certain addressing mode.

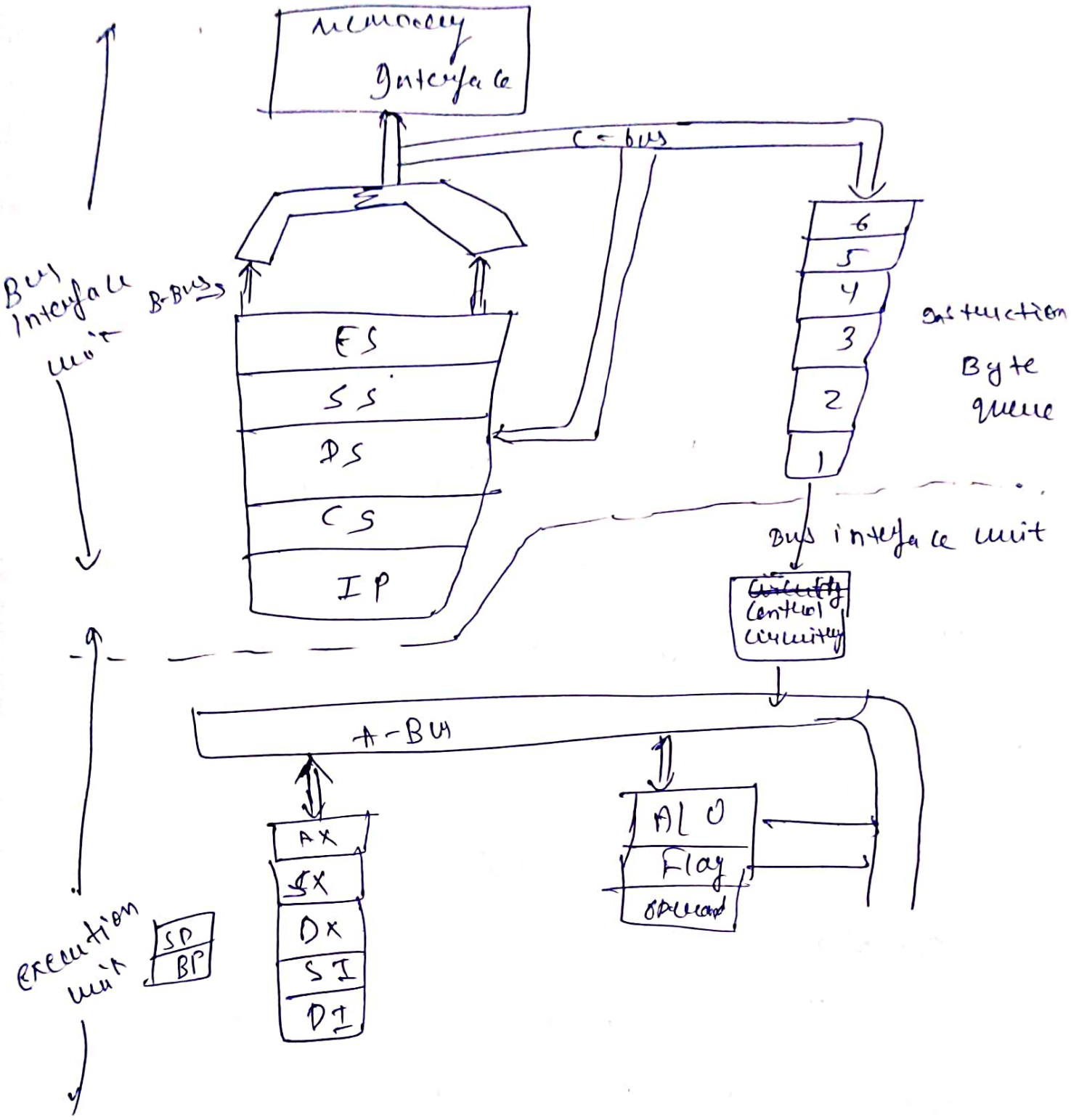
② Cx is used for default counter in case of loop and string instructions.

③ For selection of 4 segment register, the 16 bit address are provided by the BIU and used for holding the upper 16-bits of starting address also known as base address or segment address

CS → IP , SS → SP or IT

DS & ES → BP or DI

☆ Architecture of 8086 :-



Architecture is divided into two unit.

① Execution Unit :

BIU :

① It generates the physical address of the memory or input output port.

② It fetches the instruction from the memory.

③ It read / write the data from / to the memory / I/o port.

④ It supports the pipelining concept using Instruction byte queue.

⑤ It provide address relocation facility.

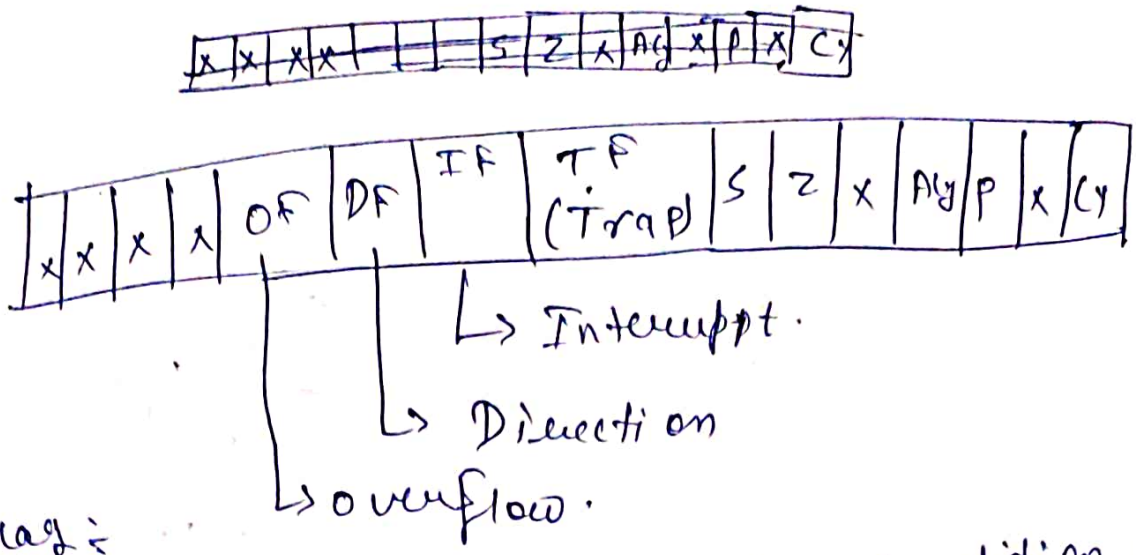
② Execution Unit :

① Control circuitry controls the internal operation of the processor

② ALU is used to perform arithmetic and logical operation on the 16 bit data.

③ Instruction decoder is used to decide that which operation should be performed.

Flag register:



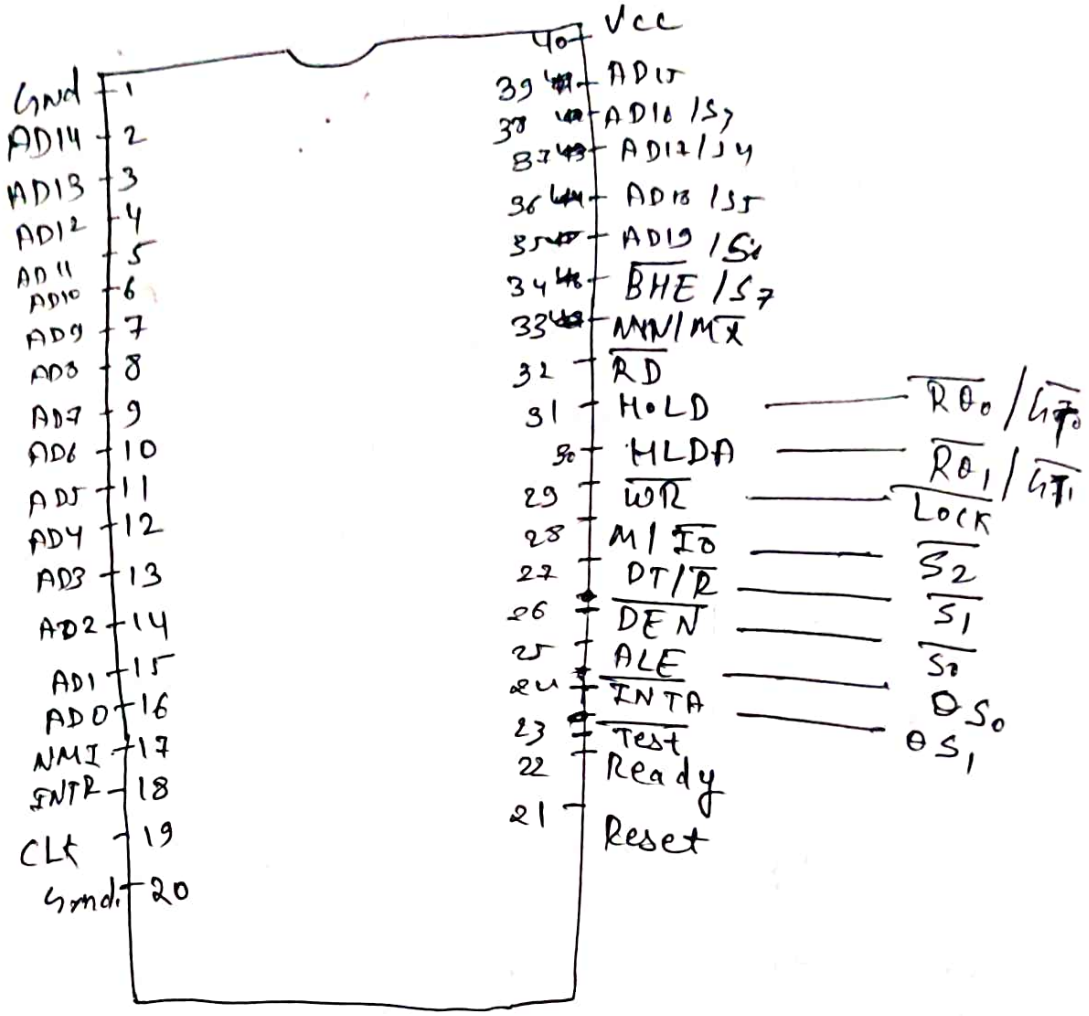
Overflow flag:
This flag is used to detect the condition of overflow. It is set 1 (set). When there is a carry into MSB and no carry out of MSB.

Direction flag: It is used in the processing of string instruction. If the string is process from lower address to higher address then it is reset (0) & vice-versa.

Interrupt flag: If the maskable interrupt is recognized by the processor then it is set (1) otherwise reset (0).

(4) Trap flag:
This flag to detecting and correcting the error through single stepping method. If its value is 1 then processor will display the content of register or memory variables after the execution of each instruction. otherwise zero.

Pin diagram :-



① Vcc / Gnd (+5v)

② AD_0 – AD15 (multiplex Address data line.)

③ AD_{16}/S_3 — AD_{19}/S_6

S_3, S_4 for choosing register

S_4	S_3	Segment register
0	0	ES
0	1	SS
1	0	CS
1	1	DS

S_5 → Interrupt flag

S_6 → always zero.

(14) $\overline{BHE}$ / S_7

↳ BUS HIGH ENABLE

$\overline{BHE}$ is used to enable the data onto High data line ($D_8 - D_{15}$)

$\overline{BHE}$	A_0	Action
0	0	Access 16-bit word
0	1	odd bitte
1	0	even bitte
1	1	No Action

(13) MN/$\overline{MX}$ ÷ 0 → maximum

1 → minimum

(12) $\overline{RD}$ ÷ It is an active low signle. and used for read operation.

(11) DT/$\overline{R}$ ÷ This signal is used to control the direction of data flow.

1 → Data is transmitting.

0 → receiving.

(10) $\overline{DEN}$ (Data enable) ÷ It is used to enable external data buses

(9) $\overline{Test}$ ÷ ~~This~~ It is an active low signal used with wait instruction. If 0, the execution continued otherwise the processor remains in Idle state.

(7) Reset → when high, the processor terminates the current activity and flags bit becomes zero. the content of CS → FFFF and the content of DS, SS, IP, ES → Zero.

(10) Ready → when high, then the processor is ready to do the data transfer with I/o device or memory. It is an acknowledgement signal from the I/o device or memory.

(9) LOCK : It is an active low signal. It indicates that the others system can not take the control of the system Buses.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Action
0	0	0	Interrupt acknowledge
0	0	1	I/o read
0	1	0	I/o write
0	1	1	Halt
1	0	0	opcode fetch
1	0	1	m/m read
1	1	0	m/m write
1	1	1	No active.

(13) $\overline{RQ_0}/\overline{GT_0}$ > $\overline{RQ_1}/\overline{GT_1}$ (priority)

Request, Grant : Signal used to take control of buses from the processor

(14) Queue status :

QS_1	QS_0	Queue status.
0	0	No operation.
0	1	first byte
1	0	Empty queue
1	1	subsequent byte.

(1) NMI ⇒ Non maskable interrupt. You cannot disable it (Level triggred)

INTR → level triggered (Can be disable)

Generation of 20bit physical Address :

16 bit address into segment ~~key~~.reg.

↳ shift it to left side by 4-bit.

→ Add the content of index or base register with it to get physical address.

CS - 1234 : IP = 5678 shifted CS = 12340

5678

P.A = 17978

A large size of memory that is 1MB in 8086. Can be divided into small areas or segments each of 64kB. It is known as memory segmentation. Each segment is specified by a segment register that gives the base or starting address.

Advantages

(1) It provides memory relocation.

(2) Memory can be access by using 16 bit register even though the size is of 1MB.

(3) other segment can also be used in the case by program if is of more than 64 KB.

÷ Addressing modes ÷

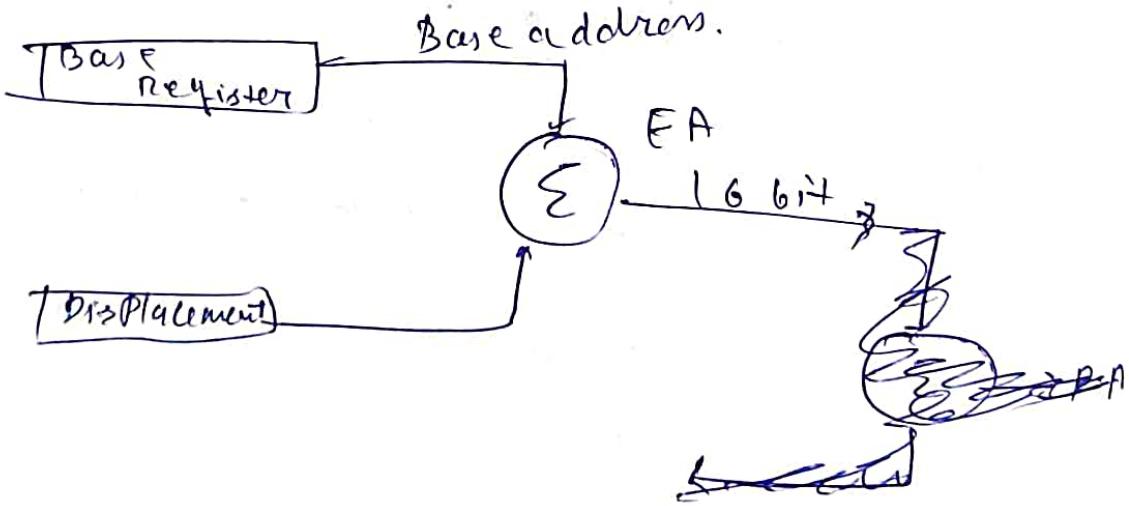
(1) (1) Immediate (2) Register (3) Register Indirect

(4) Direct (5) Relative Based (6) Base Indexed

(7) Relative Based Indexed.

(2)

Relative Based ÷

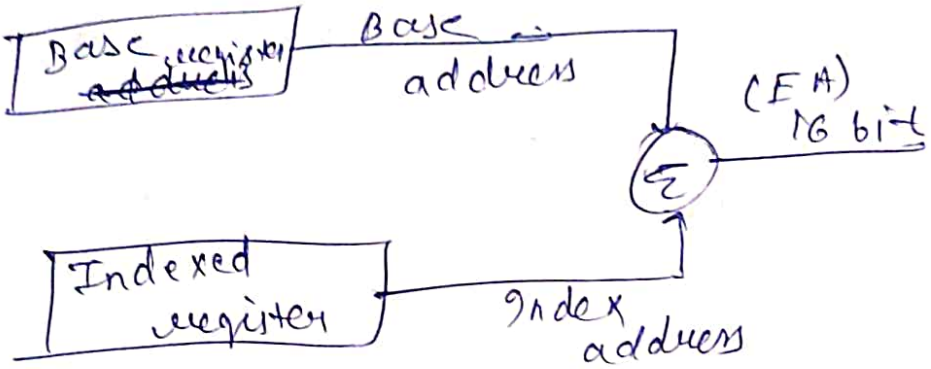


The effective address is the summation of base address given by base register and displacement.

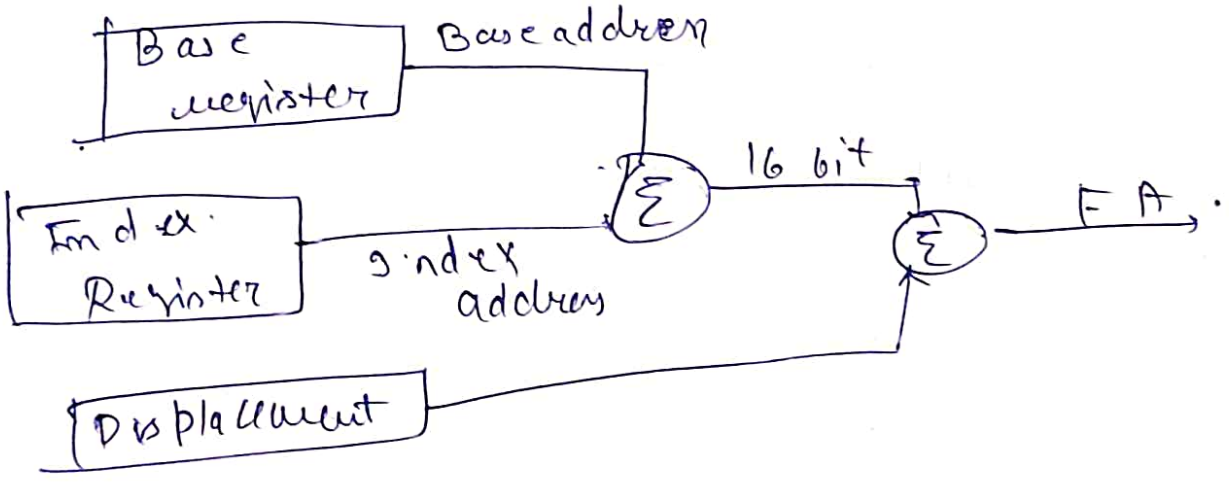
e.g MOV Ax, [SI] + Displacement.

157146

Base Indexed



Relative Base Indexed :



** Time Delay & Counter in 8085 :-

Time delay = no. of T-states * clock period.

Using a one Register :-

7T ← MVI C, FF // Counter FF

4T ← loop DCR C

10T ← JNZ loop

fmay = 1 MHz.
Clock Period = 1μs

Total no. of T-states = 7T + 255 (10T+4T)

Time delay = 7T + 255(10T+4T) * 1μs

Using a register pair :-

Time delay = No. of T-states × clock period.

LXI H, Count (1234) — 10T

loop DCX H — 6T.

used to set zero flag { MOV A, L — 4T
ORA H — 4T

JNZ loop — 10T

Time delay = 1×10^{-6} (10+24 * count in decimal

Q# to count FF to 00, with 0.5 μsec, use register C to set up a 1 millisecond delay & display each count at one of the output

Port. (reverse order = down counter)

```
      MVI B, 00
next  DCR B            → (FF) → 4T
      MVI C, count     → 7T
loop  DCR C            → 4T  ]⇒ loop
      JNZ loop         → 10T ]
      MOV A, B         → 4T
      OUT port         → 10T
      JMP next         → 10T
```

Time delay in loop $T_L = [14 \times 0.5\mu \times \text{count}]$

Time delay in outter loop $= T_{OL} = 35 \times 0.5\mu$.

Total delay $T_d = T_L + T_{OL}$

$1\,ms = 7\,\mu sec\,(\text{count}) + 17.5\,\mu sec$

$1000\,\mu s = 7\,\mu sec\,(\text{count}) + 17.5\,\mu sec$

$\text{count} = \left(\frac{1000 - 17.5}{7}\right)$

count = 140.~~357~~

Q. WAP to count from 0 to 9 with 1 sec delay. Use HL register pair & display each count with processor frequency 1MHz.

```
Start   MVI B, 00   — 7
        MOV A, B    — 4
display out port    — 10
        LXI H, count — 10
loop    DCX H       — 6
        MOV A, H    — 4
        ORA L       — 4
        JNZ loop    — 10
        INR B       — 4
        MOV A, B    — 4
        CPI 0A      → 7
        JNZ display — 10
        JZ Start.
```

loop: (24 × count + 10) × 45

$1 sec = 45 \times 1 \mu sec + (24 \times count + 10) \times 1 \mu sec.$

$10^6 \mu sec = 45 \mu sec + 24\ count\ \mu sec.$

$$count = \frac{1000000 - 45}{24} = 41665$$

~~8255~~

☆ - Programmable Peripheral Interface (PPI) or 8255.

(1) 8255 is a general purpose programmable device used for parallel data transfer.

(2) It has 24 I/o pins divided into three 8 bit ports. (Port A, B, C)

Port C can be used as individual 4 bits named as PC upper & PC lower.

(3) There are 2 mode of operation.

(a) BSR mode (8 bit set reset) :- It is used to set or reset the bits in port C.

(b) Io mode :- This mode can be programmed by writing a control word into control register.

(b1) ⇒ mode 0 :- port A and B can be configured as simple 8-bit Input-output port ~~and~~ without hand-shaking :-

Port C can be programmed seperately as 4 bit Input-output port.

(b2) ⇒ mode 1 ⇒ There are two groups each of 12 pins. Group A consists port A and port C upper. Group B consists port B and Port C lower.

port A and B are used as 8-bit into the output port with 3 lines of port C for handshaking.

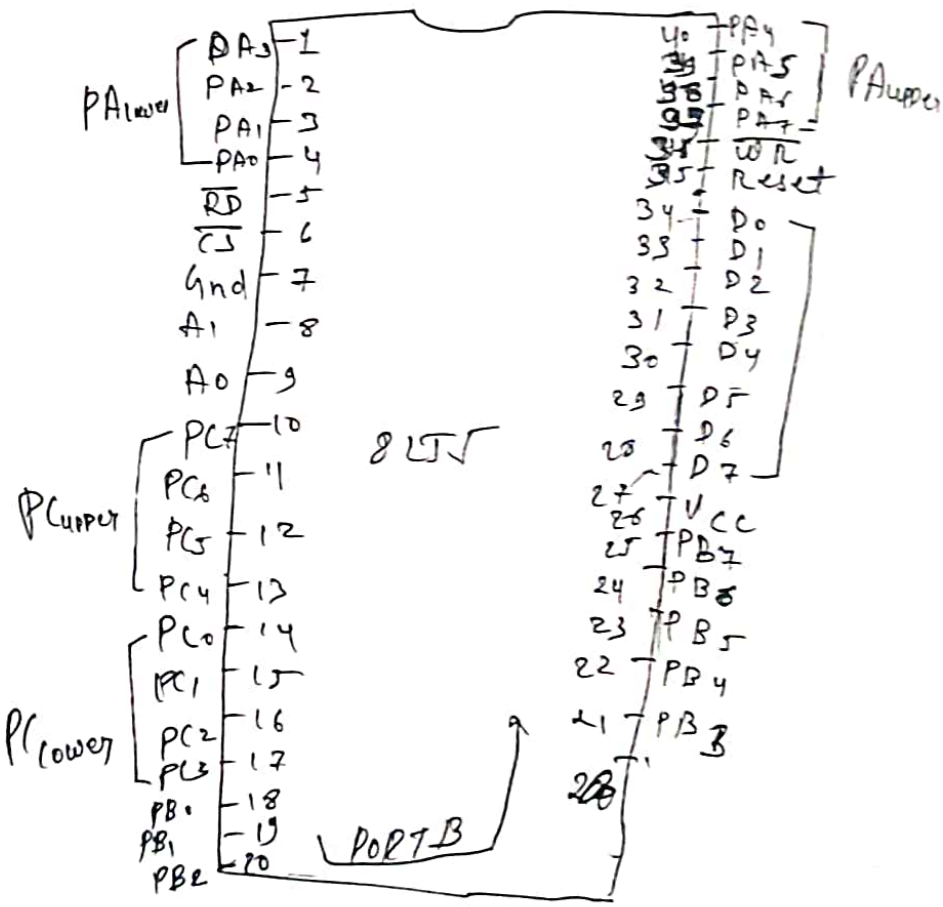
(4) 6) → Mode (2) :- Only port A can be used as bi-directional port. 5 lines of port C (PC3-PC7) are used for handshaking. port B is not used in this mode

(3) (4) It can be programmed under various condition i.e. from simple I/O to Interrupt I/O.

(5) It is compatible with all Intel microprocessor

(6) All pins of 8255 have 2.5 mAmp DC driving current.

40 Pin diagram :-



- Port A ⇒ $PA_0 - PA_7$ (8 pins)
- Port B ⇒ $PB_0 - PB_7$ (8 pins)
- Port C ⇒ $PC_0 - PC_7$ (8 pins)

Data bus ⇒ $D_0 - D_7$ (8 pins)

Power supply ⇒ +5 V (V_{cc} & Gnd)

Reset

Control signals ⇒ $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A_1, A_0

$\overline{CS}$: chip select signal

⇒ Reset :- This pin is used to reset the 8255 by connecting it to the reset out pin of the microprocessor. When active the content of control word register are cleared and all ports behave as input ports.

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	operation.
0	0	0	1	0	Port A to data bus
0	1	0	1	0	Port B to data bus
1	0	0	1	0	Port C to " "
1	1	0	1	0	Invalid.
0	0	1	0	0	Data bus to port A
0	1	1	0	0	" " " B
1	0	1	0	0	" " " C
1	1	1	0	0	" " " control register.