Priority

- Interrupt type 00H
  ↳ Divided by 00H
- Type 01H → Single Step Execution } ①
- Type 02H → NMI (Non-Maskable Interrupt) } ②
- Type 03H → INT 00 - FFH } ③
- Type 04 → Overflow [Overflow flag is set, if after processing two +ve data we get -ve output, or after processing two -ve data, we get +ve output] ~~[illegible]~~

Microcontroller 8051 (Intel)

8 bit microcontroller
↓
8 bit ALU in CPU

Max Rom that can be added = 64K

PIC 16F877
↓
another μcontroller

CPU	128 byte RAM	4K ROM
TIMER COUNTER 2	Serial Port 1	Parallel Ports 4

--→ 4 parallel I/O Ports (Port 1, Port 2, Port 3, Port 4)

- Can't perform several tasks at a time.
- Everything present in a single chip

Registers:

- $\underbrace{\text{Accumulator (A), B, R}_0\text{, R}_1\text{, R}_2\text{, R}_3\text{, R}_4\text{, R}_5\text{, R}_6\text{, R}_7\text{, PC, SP}}_{8\ \text{bit}}$

8 bit (B); ~~8 bit~~ 8 bit (SP); PC → 16 bit

- DPTR (16 bit)
  ↳ Data Pointer Register)

$$\underset{(16)}{DPTR} = \underset{(8)}{DPTR_H} + \underset{(8)}{DPTR_L}$$

- PSW (8 bit) PSW → Program Status Word
  ↳ Flag Register

PSW (8)

D7	D6	D5	D4	D3	D2	D1	D0
CY	AC	F0	RS1	RS0	OV	--	P
carry	Set for BCD Addition	Not in use	Helps in Selecting Register Bank		Overflow	Not in use	Parity

Overflow (While adding two number, and result in zero, and vice versa, overflow flag is set)

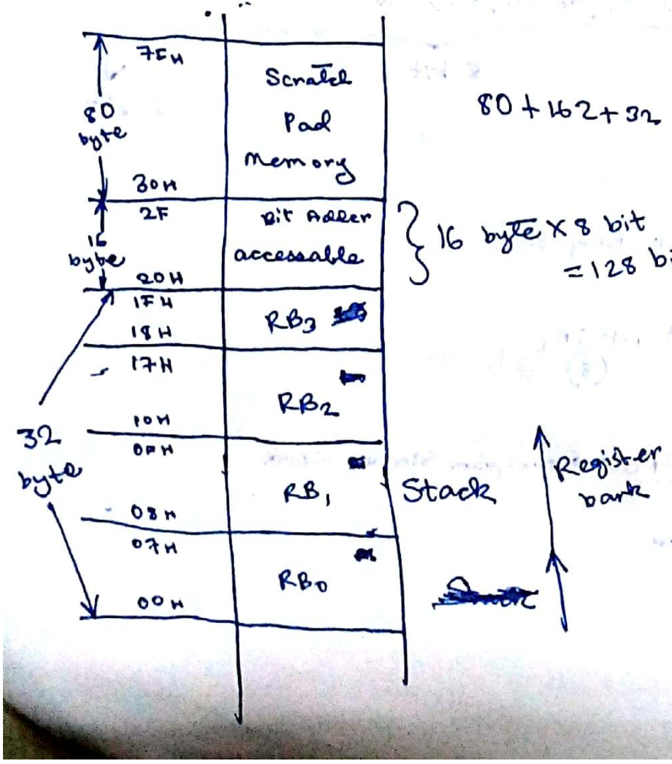
Helps in Selecting Register Bank (There are 4 reg. banks)

RS1 RS0	
00 →	Reg. Bank 0
01 →	Reg. Bank 1
10 →	Reg. Bank 2
11 →	Reg. Bank 3

RB0

RB0
R7
R6
R5
R4
R3
R2
R1
R0

Similarly for Reg. Bank 1, 2, 3



80+16+32 = 128 byte

16 byte × 8 bit = 128 bit

On ~~reset~~ RESET,

SP → 07H

MOV A, R1

MOV R1, P4

~~Set B-Set (Set B), B) Set PSW. 4~~

~~CLR PSW, 3~~

~~MOV, R1 R7~~

SET B PSW4

CLR PSW 3

MOV R1, R7

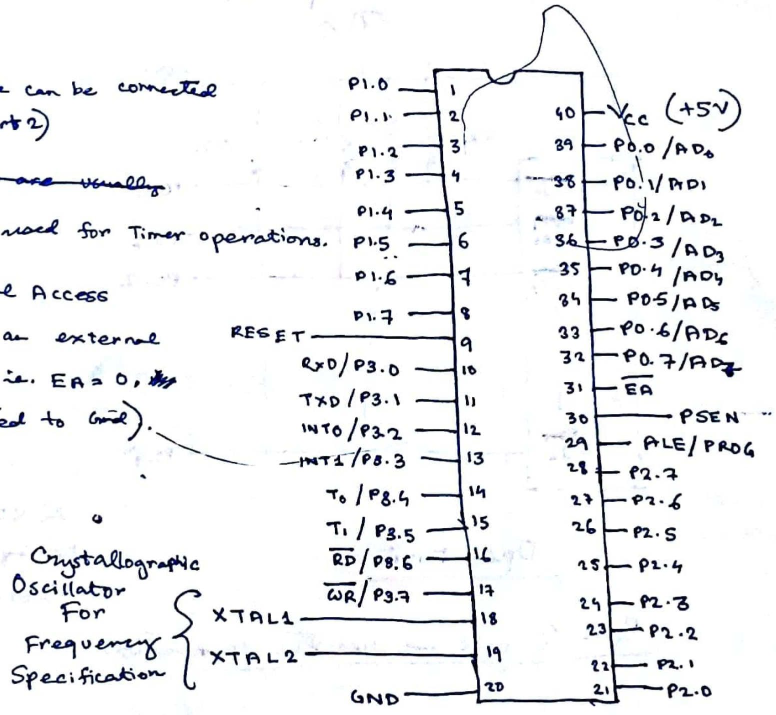
- Any I/O Device can be connected to P2 (Port 2)

~~• P3 pins ports are usually~~

- T_0, T_1 are used for Timer operations.

- EA → External Access

When we use an external ROM, $\overline{EA} = 1$, i.e. EA = 0, (it is connected to Gnd).



Pin Diagram of 8051

Power on Reset (Required for Sem)

Circuit is automatically restarted ~~if~~ when power is turned on.

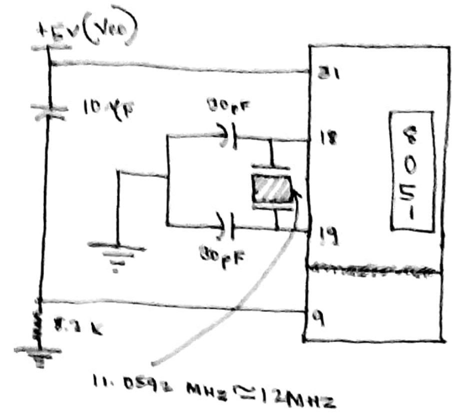
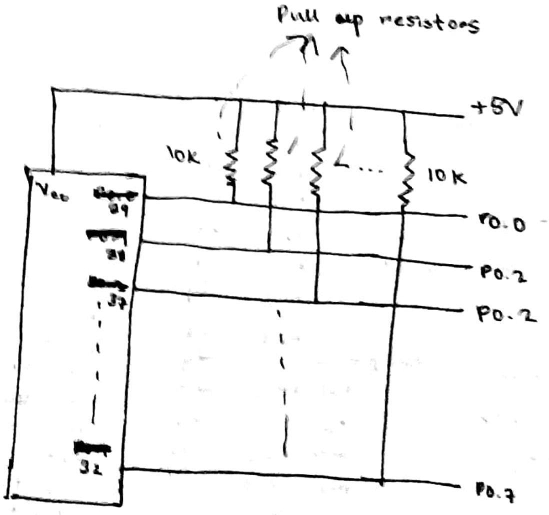


Fig: Power on RESET Circuit



Open Drain

Port 0 pull up resistor connection

- In 8051 which ports are having open drain structure. How is it solved?

8051 Processing Modes —

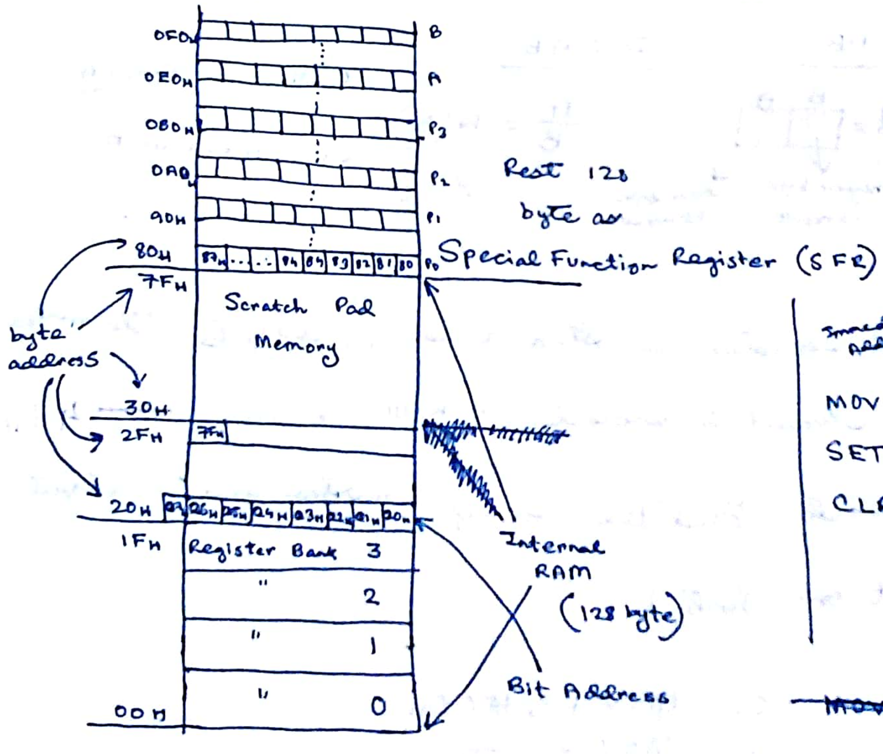
1. Immediate → MOV A,#40H → sign specifying that H is an immediate data
2. Register → MOV R1, B
3. Direct → MOV A, 40H → 40H is memory address; data in memory transfered ~~[illegible]~~ to ~~[illegible]~~ reg A
4. Indirect → MOV B,@R0 , MOV @R1,A → this sign means that R1 is holding some address
5. Indexed. → mov DPL, #~~[illegible]~~10H , mov DPH, #20H. } together
   move A,@A + DPTR

~~MOV ADD.~~

- After @R1 only A ~~[illegible]~~ or ~~[illegible]~~ B can be used.

MOV R1,#39H → R1= 39H

MOV B, @R1 → from ~~[illegible]~~ memory stored at R1 To B



Immediate Addressing mode ← P1

(90H) Memory Address (Direct Add. mode)

MOV P1,#0FFH

SETB P1.0

CLR P2.0 → Pin 21 Status → 0

~~MOV 90~~,

256 Byte ALLOCATION

Example of Direct Addressing Mode:

MOV A, 90H

SP → ~~0FH~~ 07H

PUSH 01H

POP ~~0FH~~ 03H

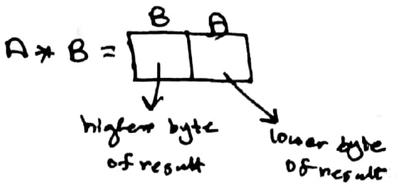
$\underbrace{\text{MOV SP, \#3FH}}$

Stack top location #3FH

```
MOV SP, #3FH
PUSH 01H
PUSH 31H
POP 18H
POP 19H
```

BCD Addition → DAA

MUL AB



DIV AB

$\frac{A}{B} = A(Q)$ → Quotient in Acc. (A)

$B(R)$ → Remainder in B

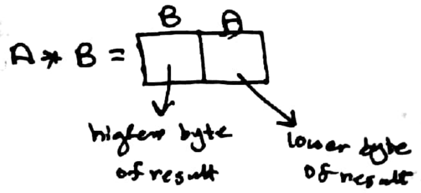
1. In a semester a student has 6 subjects. The marks of student is stored in RAM location ~~40H~~ 47_H onwards. Find the average marks and output it on Port 1.

```
0.      MOV P1,#00H
1.      MOV R2,#06H
2.      MOV B,#06H
3.      MOV R0,#47H
4.      MOV A,#00H
5. L2:  ADD A,@R0
6.      INR R0
7.      DEC R    ~~DJNZ~~
```

```
MOV SP, #3FH
PUSH 01H
PUSH 31H
 POP 18H
POP 19H
```

BCD Addition → DAA

MUL AB



DIV AB

$\frac{A}{B}$ = A(Q) → Quotient in Acc. (A)

B(R) → Remainder in B

1. In a semester a student has 6 subjects. The marks of student is stored in RAM location ~~40H~~ 47_H onwards. Find the average marks and output it on Port 1.

```
0.  MOV P1,#00H
1.  MOV R2,#06H
2.  MOV B,#06H
3.  MOV R0,#47H   ~~mov R0,47H~~ ??
4.  MOV A,#00H
5. L2: ADD A,@R0
6.  INR R0? → INC R0
7.  DEC
```

~~JNZ L2~~

8. JNZ L2

• 7,8 can be replaced as DJNZ R_2,L2

9. DIV AB

10. MOV P1, A

Timer/Counter of 8051

How to Load:

MOV TH1,#0FF$_H$

Timer 0 → $\underbrace{\text{TH0}}_{}$ + $\underbrace{\text{TL0}}_{}$

higher 8 bit   Lower 8 bit

Timer 1 → TH1 + TL1

TMOD$_{(8)}$

TCON$_{(8)}$

• TMOD → Timer MODE Set

• TCON → Timer Control Set

TMOD$_{(8)}$

D7	D6	D5	D4	D3	D2	D1	D0
Gate	T/C	M_1	M_0	Gate	T/C	M_1	M_0
Timer 1				Timer 0			

Gate → 1 → Means we require some hardware triggering for controlling the timer operation

0 → Means we require some software operation to trigger

T/C → Used for controlling Timer/Counter operations.

1 → Counter → for event counting

0 → Timer

M_1 M_0 → Used for setting the mode of timer/counter

0 0 → Mode 0 (13 bit)

0 1 → mode 1 (16 bit) → (Normally Used)

1 0 → mode 2 (8 bit auto reload)

1 1 → Mode 3 (Split Mode)

TCON$_{(8)}$ → Present in the SFR Zone

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF$_x$ → Timer Overflow Flag

TR$_x$ → Timer Run/Start-Stop Flag

IE$_x$ → Ext. Interrupt Occurrence Flag

IT$_x$ → Ext Interrupt Type

? Check

IE$_0$ → 1 → Implies Ext. INT 0 is high

→ 0 → When the Mµcontroller is serving some interrupt on INT0

TF$_x$ → Check Overflow. On the next go, it resets its value.

~~Setting~~

Clock pulse → 11. 0892 MHz

≃ 12 MHz

~~⩸~~ T = 1.085µ

mov TMOD, # 01$_H$ → Timer 0 in Mode 1

mov TH0, # 0~~0~~$\overset{FF}{\wedge}$$_H$

mov TL0, # 0$\overset{F}{\wedge}$0$_H$

Cont
FFFF FFF0 ↑

~~mov TMOD, # 01$_H$ → Timer 0 in Mode~~

~~mov TH0, # 00$_H$~~

~~mov TL0, # 00$_H$~~

SET TR0

L1 : JNB TF0, L1

CLR TR0

CLR TF0

SET TR0 → Set TR0 ~~to~~ to 1

Keep counting till Overflow (TF0 → 1)

↓

Means Overflow

L1: JNB TF0, $\overset{L1}{\wedge}$ → Branch when TF0 is not high

- In the following program, a waveform is generated which has equal high and low period, we are creating a ~~[illegible]~~ square wave of 50% duty cycle ~~of~~ on P1.5 bit. Timer 0 is used for generation of time delay. Calculate the frequency of square wave generated on P1.5.

```
      MOV  TMOD,#01H           2T
  L1: MOV  TH0, #0FFH          2T
      MOV  TL0 , #0F0H         2T
      CPL  P1.5 → Complement   1T
      CALL DELAY               2T
?LJMP ? SJMP L1 → Short Jump   2T

DELAY: SETB TR0    1T
  L2: JNB TF0, L2  16T
      CLR TR0      1T
      CLR TF0      1T
      RET          2T
                 ______
                  30 + 2 → First Line is out of Loop
```

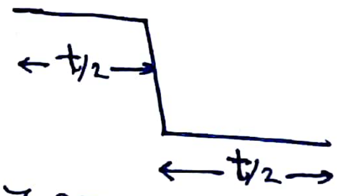
Time Delay

FFFF

↑ 16 T

FFF0

Calculating frequency:



$$t = (2 \times 30 + 2) \times 1.085\ \mu s = 67.27\ \mu s$$

$$\therefore f = \frac{1}{t} = 0.014\ MHz \text{ (Ans)}$$