Microcontroller

→ 8051 – Architecture, Signals, I/O Ports, Memory, Counters & Timers – Serial data I/O Interrupts. Interfacing – Keyboard, LCD, Stepper Motor Control.

The building block of 8051 :- CPU, Interrupt controller, on chip RAM, on chip ROM, 4 I/O Ports, serial port, oscillator ckt, bus, control and timer (timer 0 & timer 1) ↓ used to produce delay.

* Register section of 8051 :-

widely used A, B

8 others are $R_0, R_1 \ldots R_{87}$ } 8 bit register.

two popular 16-bit register :

i) PC (Program Counter)

ii) DPTR (Data Pointer).

Instruction MOV

1) MOV A, #50H → load the value 50H in register A

2) MOV A, 50H → to move into A, the value held in memory location 50H

3) MOV R1, A → similar to processor

4) MOV A, #F9H → error

MOV A, #0F9H → ✓

Instruction ADD

3) Add two numbers 25H & 34H.

→ MOV A, #25H

MOV R2, #34H

ADD A, R2.

certain things are not allowed.

→ An instruction such as ADD R2, #812H is invalid because any arithmetic operation must involve the register A, Also, inst. like ADD R3, A is invalid because A should be destination address.

* PSW / flag register.
(Program status word)

→ this is 8 bit register of which 2 bits are used as user-definable flags.

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	-	P

PSW.7 → Carry

PSW.6 → Auxilliary Carry

PSW.5 → F0 (zero) → available to the user for general purpose

PSW.4 → RS1 → Register bank selecta bit 1.

PSW.3 → RS0 → " " " bit 0.

PSW.2 → OV → overflow overflow flag used for sign bit operation

PSW.1 → User definable

PSW.0 → Parity

Carry Flag :- It can be set to 1 or directly by an instruction such as.
"SETB C" → set bit carry "CLR C" → clear carry.

Parity flag :- If the "A" register contains an odd no. of 1's then P = 1,
for even no. of 1's, P = 0.

RAM Memory space allocation in 8051

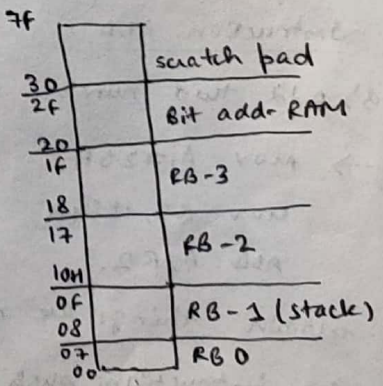
There are 128 bytes of RAM in 8085 which are assigned addresses from 00 to 07H & can be accessed directly as memory locations.

RAM is divided into 3 diff. grp as follows :-

1) A total of 32 bytes of gram from locations from 00 to 1FH and set aside for register banks and stack.

2) A total of 16 bytes from locations from 20 to 2FH are set aside for bit-addressable R/W memory. Here we can read or write any single bit by using unique address for that bit

3) A total of 80 bytes from location 80 to 7FH are used for read and write storage normally known as scratch pad, they might be use for storing data and parameter by the program.



gister bank section

32 bytes divided into 4 register bank. Each consisting of 8 register.

→ Register Bank 1 uses the same RAM space as the stack.

→ Register Bank 0 rating from 00 to 07 is the default register bank, that is the when 8051 is powered up RAM locations 0, 1, ... 7 are accessed with the names $R_0, R_1 \ldots R_7$.

→ Switching of Register Banks.

One can switch b/w to any other register bank by using bit D_4 & D_3 of the PSW register.

	RS1 (PSW4)	R10 (PSW3)
Bank 0	0	0
" 1	0	1
" 2	1	0
" 3	1	1

PSW 4 & PSW 3 can be accesed by the bit addressable instructions "SETB" & "CLR".

Q) Write instructions to use the register bank 3 and load the value 05H in the register R_0 & R1.

Soln :-
```
SETB PSW4
SETB PSW3
MOV R0, #05H
MOV R1, #05H
```

Addressing mode of 8051 :-

1) Immediate 2) Register 3) Direct 4) Register Indirect 5) Index.

1) Immediate :

```
MOV A, #25H
MOV DPTR, #1008H
     DPH
     DPL
MOV DPH, #10H
MOV DDL, #08H
```

2) Also Register Addressing Mode :-

→ It involves the use of registers to hold the data to be manipulated.

Ex:- MOV A, R0
MOV R5, A

MOV R5, R0 X
(invalid)
involve A → right way.

3) Direct.

→ this is used to access data stored in RAM & registers of 8051.

```
MOV  R0, 40H
MOV  55H, A      (content of 40H loaded into R0)
```

[SFR (special function registers)
→ each of them have specific location in memory
→ ranges from 80H – FFH.

Direct add mode can also be used in stack.

POP 04
It stores the top of stack into R4 of chosen bank.

4) Indirect addressing mode :-
In this mode, a register is used as a pointer to the data. only registers R_0 & R_1 are used for this purpose. R_2 to R_7 can not be used to hold address of operand located in RAM. When R_0 & R_1 hold the addresses of RAM locations, they must be preceded by '@'.

Eg:- i) MOV A, @R_0 [move contents of RAM location whose address is held by R_0 to 'A']

ii) MOV @R1, B [contents of B is to be transfered to the address indicated by R1].

5) Write a program to copy the value 55H into RAM location 40H to 45H, used in
(i) direct addressing mode (ii) with a loop.

Ans) (i)
```
MOV A, #55H
MOV 40H, A
MOV 41H, A
:
MOV 45H, A
```

(ii)
```
MOV A, #55H
MOV R0, #40H   (RAM pointer)  or
MOV R2, #05H   (data counter)
```

(ii)
```
MOV @R0, A
INC R0
DJNZ R2, Again
↓
(decrement & jump
if register ≠ 0)
```

5) Index addressing mode :

- MOVC A, @A + DPTR

the 16 bit register DPTR and register A are used to form the address of the data element stored in on-chip ROM.

```
DPTR – 1000H
A   –    68H
-------------
      1068H → final address.
```

Move the content of 1068 to the 'A'

MOVC → Move code