

Unit + 8

1.) 8051 Addressing Modes ...

The 8051 microcontroller has several addressing modes, which define how the operand (data) is accessed in an instruction.

Types of Addressing Mode :-

- i) Register Addressing mode.
- ii) Immediate Addressing mode.
- iii) Direct Addressing mode.
- iv) Indirect Addressing mode.
- v) Index Addressing mode.

i) Register Addressing mode :-

ADD . A, B → Add the content of B register to accumulator.

MOV A, B → Move the content of B register to accumulator.

→ B register to accumulator.

→ The operand is in a register.

ii) Immediate Addressing mode :-

Move A, #12H → Move immediately 12H to accumulator.

→ The operand is given directly in the instruction.

iii) Direct Addressing Mode :-

MOV A, 35H → Move B, 80H

MOV A, 4000H (copy data from memory address 400H into A).

→ The address of the operand is specified directly.

iv) Indirect Addressing Mode :- MOV

- MOV A, @R0 (R0 holds the address of the operand).
- Move the content of the RAM memory location whose address is specified by R1 Register to accumulator.
 - The address of the operand is held in a register.

v) Index Addressing Mode :-

- MOV C A, @A + DPTR → Move the content of the external memory, whose address is generated by adding content of accumulator and DPTR to accumulator.
- MOV A, 3000H + x (x is index register).
- The effective address is obtained by adding a constant (index to register).

2) Instruction Set ...

Data transfer Instruction Set ...

- ① MOV R, m :- Move the content memory to register.
- ② MOV m, R :- Move the content register to memory.
- ③ MVI R, data :- Move immediate data to register.
- ④ MVI m, data :- Move immediate data to memory.
- ⑤ LXI rp, data :- Load register pair immediate.
- ⑥ LDA, address :- Load accumulator direct.
- ⑦ STA, address :- Store accumulator direct.
- ⑧ LHLD, address :- Load HL (register) pair direct.
- ⑨ SHLD, address :- Store HL (register) pair direct.
- ⑩ LDAX, rp :- Store accumulator indirect.
- ⑪ STAX, rp :- Store accumulator indirect.

⑫ XCHG :- Exchange the content of HL (register) with DE pair.

3. Arithmetic Instruction Set

➤ Addition :-

- ① ADD, r :- Add register to accumulator.
- ② ADD, m :- Add memory to accumulator.
- ③ ADC, r :- Add register carry to accumulator.
- ④ ADC, m :- Add memory carry to accumulator.
- ⑤ ADI, add :- Add immediate data to accumulator.
- ⑥ ACI, data :- Add with carry immediate data to accumulator.
- ⑦ DAD, rp :- Add register pair to HL pair.

➤ Subtraction :-

- ① SUB, r :- Subtract register from accumulator.
- ② SUB, m :- Subtract memory from accumulator.
- ③ SBB, r :- Subtract register from accumulator with borrow.
- ④ SBB, m :- Subtract memory from accumulator with borrow.
- ⑤ SUI, data :- Subtract immediate data from accumulator.
- ⑥ SBI, data :- Subtract immediate data from accumulator with borrow.
- ⑦ INR, r :- Increment register content (+1).
- ⑧ INR, m :- Increment memory content.
- ⑨ DCR, r :- Decrement register content (-1).
- ⑩ DCR, m :- Decrement memory content.
- ⑪ INX, rp :- Increment register pair.

⑫ DCX, r_p :- Decrement register pair.

⑬ DAD :- Decimal Adjust Accumulator.

Logical Ground Instruction Set...

① ANA, r :- AND register with accumulator.

② ANA, m :- AND memory with accumulator.

③ ANI, $data$:- AND immediate data with accumulator.

④ ORA, r :- OR register with accumulator.

⑤ ORA, m :- OR memory with accumulator.

⑥ ORI, $data$:- OR immediate data with accumulator.

⑦ XRA, r :- Exclusive OR register with accumulator.

⑧ XRA, m :- Exclusive OR memory with accumulator.

⑨ XRI, r :- Exclusive OR immediate register with accumulator.

⑩ CMA :- Compliment the accumulator status.

⑪ CMC :- Compliment the carry status.

⑫ SCS :- Set carry status.

⑬ CMP, m :- Compare memory with accumulator.

⑭ CMP, r :- Compare register with accumulator.

⑮ CPI, $data$:- Compare immediate data with accumulator.

⑯ RLC :- Rotate accumulator left.

⑰ RRC :- Rotate accumulator right.

⑱ RAL :- Rotate accumulator left through carry.

⑲ RAR :- Rotate accumulator right through carry.