Addressing modes

The way of accessing data is called addressing mode. The CPU can access the data in different ways by using addressing modes. The 8051 microcontroller consists of five addressing modes such as:

- Immediate Addressing Mode
- Register Addressing Mode
- Direct Addressing Mode
- Indirect Addressing Mode
- Base Index Addressing Mode

Immediate Addressing Mode:

In this addressing mode, the source must be a value that can be followed by the '#' and destination must be SFR registers, general purpose registers and address. It is used for immediately storing the value in the memory registers.

Syntax:

MOV A, #20h   //A is an accumulator register, 20 is stored in the A//

MOV R0,#15   // R0 is a general purpose register; 15 is stored in the R0 register//

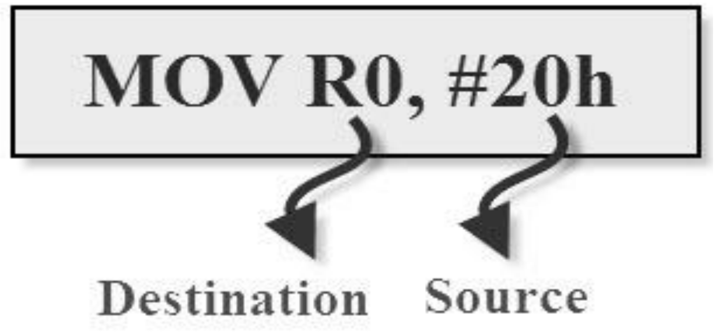
MOV P0, #07h   //P0 is a SFR register;07 is stored in the P0//

MOV 20h,#05h   //20h is the address of the register; 05 stored in the 20h//

Ex:

MOV R0, #1

MOV R0, #20   //R0 <—R0[15]+20, the final value is stored in R0//



Register Addressing Mode:

In this addressing mode, the source and destination must be a register, but not general purpose registers. So the data is not moved within the general purpose bank registers.

Syntax:

MOV A, B; // A is a SFR register, B is a general purpose register//

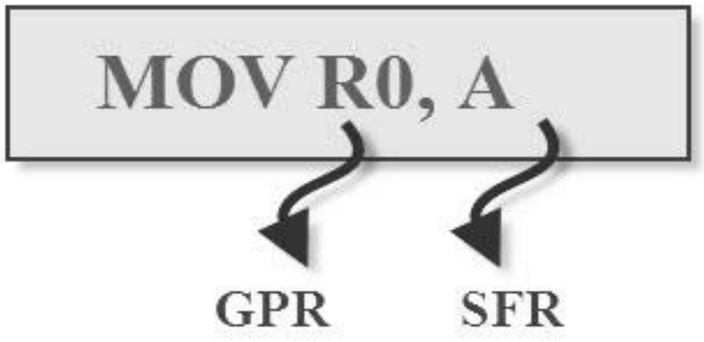
MOV R0, R1 //Invalid instruction, GPR to GPR not possible//

EX:

MOV R0, #02h

MOV A, #30h

ADD R0, A //R0<—R0+A, the final value is stored in the R0 register//



Direct Addressing Mode

In this addressing mode, the source or destination (or both source and destination) must be an address, but not value.

Syntax:

MOV A,20h // 20h is an address; A is a register//

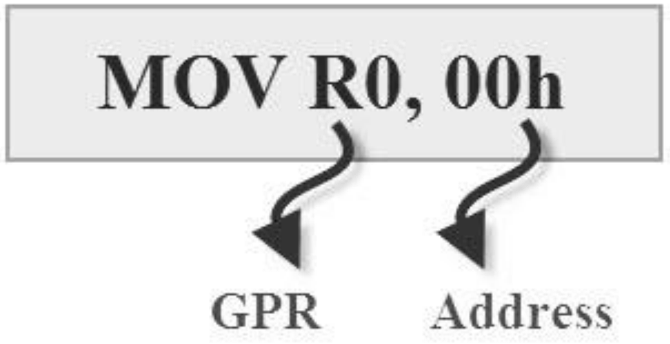
MOV 00h, 07h // both are addressed of the GPS registers//

Ex:

MOV 07h,#01h

MOV A, #08h

ADD A,07h //A<—A+07h the final value is stored in A//

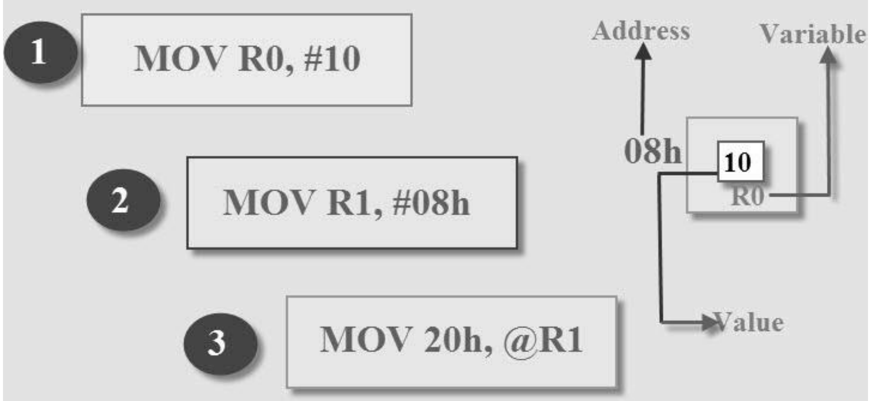


Indirect Addressing Mode:

In this addressing mode, the source or destination (or destination or source) must be a indirect address, but not a value. This addressing mode supports the pointer concept. The pointer is a variable that is used to store the address of the other variable. This pointer concept is only used for R0 and R1 registers.

Syntax:

```
MOVR0, #01h    //01 value is stored in the R0 register, R0 address is 08h//
MOV R1, #08h//R1 is the pointer variable that stores address (08h) of R0 //
MOV 20h,@R1    //01 value is stored in the 20h address of the GP register//
```



Base Index Addressing Mode:

This addressing mode is used to read the data from the external memory or ROM memory. All addressing modes cannot read the data from the code memory. The code must read through the DPTR register. The DPTR is used to point the data in the code or external memory.

Syntax:

```
MOVC A, @A+DPTR   //C indicates code memory//
MOCX A, @A+DPTR    // X indicate external memory//
EX:    MOV A, #00H        //00H is stored in the A register//
MOV DPTR, #0500H        //DPTR points 0500h address in the memory//
MOVC A, @A+DPTR    //send the value to the A register//
MOV P0, A           //date of A send to the PO registrar//
```