

ADDRESSING MODES OF 8051

Addressing Modes is the manner in which operands are given in the instruction.
8051 supports the following 5 addressing modes:

1) IMMEDIATE ADDRESSING MODE

In this addressing mode, the **Data** is given in the **Instruction** itself.
We put a "#" symbol, before the data, to identify it as a data value and not as an address.

Eg: MOV A, #35H ; A ← 35H

 MOV DPTR, #3000H ; DPTR ← 3000H

2) REGISTER ADDRESSING MODE

In this addressing mode, **Data** is given by a **Register** in the instruction.
The permitted registers are **A, R7 ... R0** of each memory bank.

Note: Data transfer between two RAM registers is not allowed.

Eg: MOV A, R0 ; A ← R0 ... If R0 = 25H, then A gets the Value 25H.

 MOV R5, A ; R5 ← A

 MOV Rx, Ry ; NOT ALLOWED. That's because this would allow 64 combinations of registers
 ; As registers invite opcodes, this would need 64 opcodes!

3) DIRECT ADDRESSING MODE

Here, the **address** of the operand is given in the **instruction**.

Only Internal RAM addresses (**00H...7FH**) and **SFR** addresses (from **80H to FFH**) allowed.

Eg: MOV A, 35H ; A ← Contents of RAM location 35H

 MOV A, 80H ; A ← Contents of Port 0 (SFR at address 80H)

 MOV 20H, 30H ; [20H] ← [30H]
 ; i.e. Location 20H gets the contents of Location 30H

4) INDIRECT ADDRESSING MODE

Here, the **address** of the operand is given **in a register**.

Internal RAM and External RAM can be accessed using this mode.

The advantage of giving an address using a register is that we can increment the address in a loop, by simply incrementing the register, and hence access a series of locations.

INTERNAL RAM: (8-BIT ADDRESS GIVEN BY R0 OR R1)

ONLY R1 or R0, called as *Data Pointers*, can be used to specify **address** (00H ... 7FH).

An "@" sign is present before the register to indicate that the register is giving an address.

Eg: `MOV A, @R0` ; $A \leftarrow [R0]$
 ; i.e. $A \leftarrow$ Contents of Internal RAM Location whose address is given by R0.
 ; if $R0 = 25H$, then A gets the contents of Location 25H from Internal RAM

`MOV @R1, A` ; $[R1] \leftarrow A$ i.e. Internal RAM Location pointed by R1 gets value of A.

EXTERNAL RAM: (16 BIT ADDRESS GIVEN BY DPTR)

For the **External RAM**, **address** is provided by **R1 or R0**, or by **DPTR**.

If DPTR is used to give an address, then the full 64KB range of External RAM from 0000H... FFFFH is available. This is because DPTR is 16-bit and $2^{16} = 65536$.

An "X" is present in the instruction, to indicate External RAM.

Eg: `MOVX A, @DPTR` ; $A \leftarrow [DPTR]^{\wedge}$
 ; A gets the contents of External RAM Location whose address is given by DPTR
 ; If $DPTR = 2000H$, then A gets contents of Location 2000H from External RAM

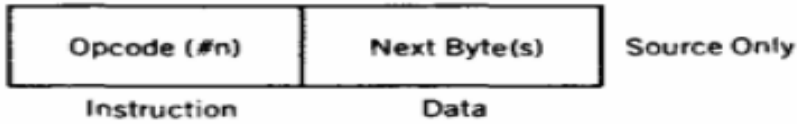
`MOVX @DPTR, A` ; $[DPTR]^{\wedge} \leftarrow A$
 ; i.e. A is stored at the External RAM Location whose address is given by DPTR

EXTERNAL RAM: (8 BIT ADDRESS GIVEN BY R0 OR R1)

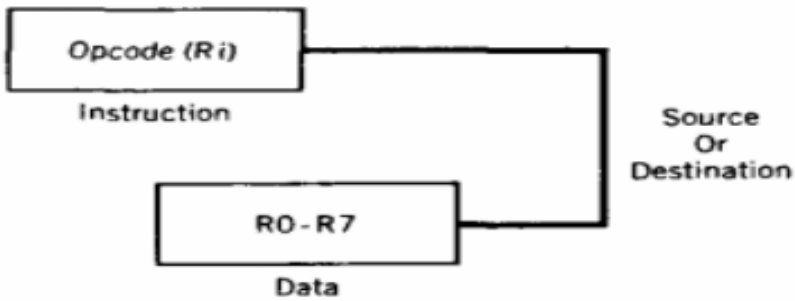
If R0 or R1 is used to give an address, then only the first 256 locations of External RAM is available from 0000 H to 00FF H. This is because R0 or R1 are 8-bit and $2^8 =$ only 256.

Eg: `MOVX A, @R0` ; $A \leftarrow [R0]^{\wedge}$
 ; i.e. A gets the contents of External RAM Location whose address is given by R0
 ; If $R0 = 25H$, then A gets contents of Location 0025H from the External RAM

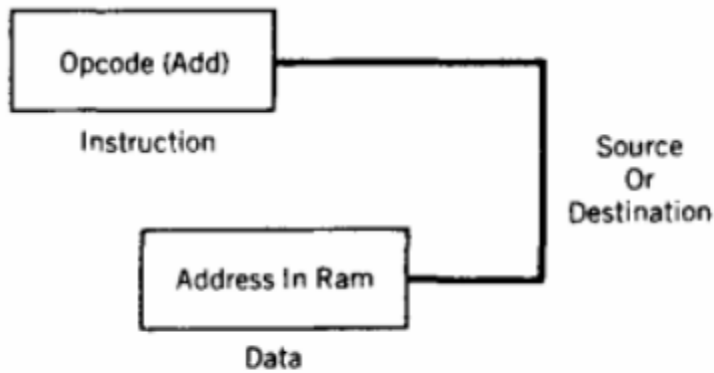
`MOVX @R1, A` ; $[R1]^{\wedge} \leftarrow A$
 ; i.e. A is stored at the External RAM Location whose address is given by R1



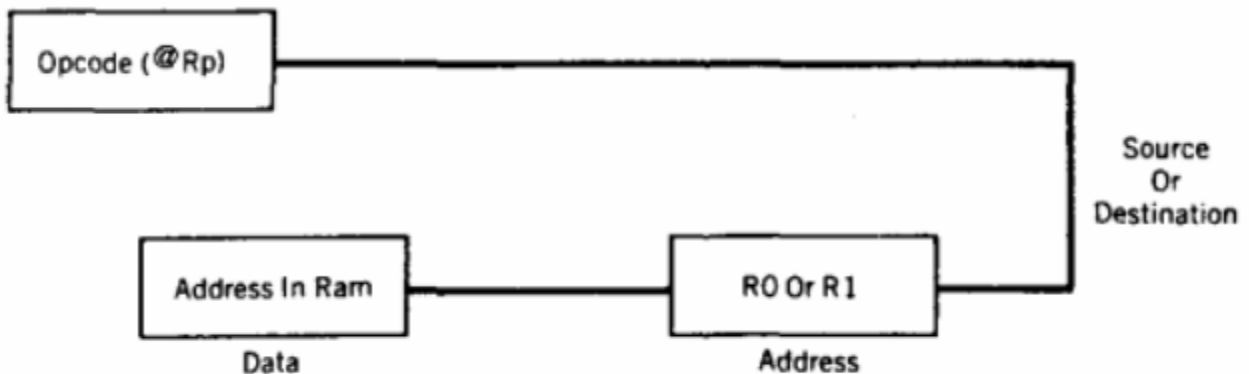
Immediate Addressing Mode



Register Addressing Mode



Direct Addressing Mode



Indirect Addressing Mode