

## **DATA TRANSFER INSTRUCTIONS**

Here you will see operations like:

- MOV
- MOVS
- MOVC
- PUSH
- POP
- Exchange
- Exchange Digit

## 51) MOV A, #n | “Move”

Example:

**MOV A, #25H; A ← 25H**

*Operation:*

*A Register gets the value of the Immediate data.*

No of cycles required: **1**

---

## 52) MOV A, Rr | “Move”

Example:

**MOV A, R0; A ← R0**

*Operation:*

*A Register gets the value of a RAM register.*

*The value remains in the RAM register and is also copied into A register.*

No of cycles required: **1**

---

## 53) MOV A, addr | “Move”

Example:

**MOV A, 25H; A ← [25H]**

*Operation:*

*A Register gets the contents of the address.*

No of cycles required: **1**

---

## 54) MOV A, @Rp | “Move”

Example:

**MOV A, @R0; A ← [R0]**

*Operation:*

*A Register gets the contents of the location pointed by the register.*

*If R0 = 20H and Location 20H contains value 35H, then 35H will be copied into to A register.*

No of cycles required: **1**

---

**55) MOV Rr, A**

| “Move”

Example:

**MOV R0, A; R0 ← A***Operation:**RAM Register gets the value of A Register.*No of cycles required: **1**

---

**56) MOV Rr, #n**

| “Move”

Example:

**MOV R0, #25H; R0 ← 25H***Operation:**RAM Register gets the value of the Immediate Data.*No of cycles required: **1**

---

**57) MOV Rr, addr**

| “Move”

Example:

**MOV R0, 25H; R0 ← [25H]***Operation:**RAM Register gets the contents of the address.*No of cycles required: **2**

## 58) MOV addr, A | “Move”

Example:

**MOV 25H, A; [25H] ← A**

*Operation:*

*The RAM location gets the value of A Register.*

No of cycles required: **1**

---

## 59) MOV addr, #n | “Move”

Example:

**MOV 25H, #25H; [25H] ← 25H**

*Operation:*

*The RAM location gets the value of the Immediate Data.*

No of cycles required: **2**

---

## 60) MOV addr, Rr | “Move”

Example:

**MOV 25H, R0; [25H] ← R0**

*Operation:*

*The RAM location gets the contents of the RAM register.*

No of cycles required: **2**

---

## 61) MOV addr1, addr2 | “Move”

Example:

**MOV 30H, 25H; [30H] ← [25H]**

*Operation:*

*The RAM location 1 gets the contents of RAM location 2.*

No of cycles required: **2**

---

## 62) MOV addr, @Rp | “Move”

Example:

**MOV 25H, @R0; [25H] ← [R0]**

*Operation:*

*The RAM location gets the contents of the address pointed by the register.*

No of cycles required: **2**

---

**63) MOV @Rp, A** | “Move”

Example:

**MOV @R0, A; [R0] ← A***Operation:**The RAM location pointed by the register gets the value of A Register.*No of cycles required: **1**

---

**64) MOV @Rp, #n** | “Move”

Example:

**MOV @R0, #25H; [R0] ← 25H***Operation:**The RAM location pointed by the register gets the value of the Immediate Data.*No of cycles required: **1**

---

**65) MOV @Rp, addr** | “Move”

Example:

**MOV @R0, 25H; [R0] ← [25H]***Operation:**The RAM location pointed by the register gets the contents of the address.*No of cycles required: **2**

---

**66) MOV DPTR, #nn** | “Move”

Example:

**MOV DPTR, #2000H; DPTR ← 2000H***Operation:**DPTR register gets the 16 bit Immediate Data given in the instruction.*No of cycles required: **2**

## 67) MOVX A, @Rp | “Move Ex”

Example:

**MOVX A, @R0; A ← [R0]^**

*Operation:*

*A Register gets the data from the location pointed by R0 in the External RAM.*

*If R0 = 20H then A Register gets data from External RAM location 0020H.*

No of cycles required: **2**

---

## 68) MOVX A, @DPTR | “Move Ex”

Example:

**MOVX A, @DPTR; A ← [DPTR]^**

*Operation:*

*A Register gets the data from the location pointed by DPTR in the External RAM.*

*If DPTR = 4000H then A Register gets data from External RAM location 4000H.*

No of cycles required: **2**

---

## 69) MOVX @Rp, A | “Move Ex”

Example:

**MOVX @R0, A; [R0]^ ← A**

*Operation:*

*Location pointed by R0 in the External RAM gets the data from A Register.*

*If R0 = 20H then value of A Register gets stored into External RAM location 0020H.*

No of cycles required: **2**

---

## 70) MOVX @DPTR, A | “Move Ex”

Example:

**MOVX @DPTR, A; [DPTR]^ ← A**

*Operation:*

*Location pointed by R0 in the External RAM gets the data from A Register.*

*If DPTR = 4000H then value of A Register gets stored into External RAM location 4000H.*

No of cycles required: **2**

### IMPORTANT TIP FROM BHARAT ACHARYA

When working with External RAM, remember the following rules:

**1) We cannot use direct addressing mode.**

We can ONLY use Indirect addressing using R0/ R1 or DPTR.

**2) All data can only be transferred to or from “A” register.**

MOVX operates only on A Register.

**3) We can only perform data transfers with these memories.**

We can NOT directly perform ADD or SUB etc.

Such operations are only allowed with Internal RAM.

**71) MOVC A, @A + DPTR** | “Move C”

Example:

**MOVC A, @A + DPTR; A ← [A + DPTR]<sub>ROM</sub>***Operation:***A Register gets the data from the location pointed by A + DPTR in ROM.***If A = 05H and DPTR = 0400H, then A Register gets data from ROM location 0405H.**This operation can happen either on Internal ROM or External ROM.**If the address formed after adding A + DPTR is 1000H or more, the operation will happen on External ROM.**If the address is less than 1000H, then it depends upon  $\overline{EA}$  pin.**If  $\overline{EA}$  pin = 0, operation will happen on External ROM.**If  $\overline{EA}$  pin = 1, operation will happen on Internal ROM.*No of cycles required: **2****IMPORTANT TIP FROM BHARAT ACHARYA**

This operation can happen either on Internal ROM or External ROM.

If the address formed after adding A + DPTR is 1000H or more, the operation will happen on External ROM.

If the address is less than 1000H, then it depends upon  $\overline{EA}$  pin.If  $\overline{EA}$  pin = 0, operation will happen on External ROM.If  $\overline{EA}$  pin = 1, operation will happen on Internal ROM.

This instruction is extremely useful in accessing Look Up Tables. DPTR is initialized with the starting address of the table. E.g.: 0400H.

The required index in the table is initialized in A register. E.g.: 05H.

By doing MOVC A, @ A+DPTR, A register gets the value from element 5, of table starting at 0400H.

Please recollect the 7-Segment code translation example from the classroom lecture.

**71) MOVC A, @A + PC** | “Move C”

Example:

**MOVC A, @A + PC; A ← [A + PC]<sub>ROM</sub>***Operation:***A Register gets the data from the location pointed by A + PC in ROM.***If A = 25H and PC = 0400H, then A Register gets data from ROM location 0425H.*No of cycles required: **2**

## 73) PUSH addr

| “Push”

Example:

**PUSH 25H; First  $SP \leftarrow SP + 1$ , Then  $[SP] \leftarrow [25H]$**

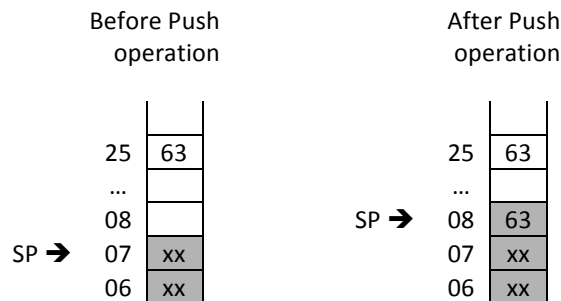
Operation:

*This instruction is used to push new data into the top of stack.*

*First SP will become  $SP + 1$*

*Then at the new location pointed by SP, data from the specified address will be pushed.*

*This newly pushed data will now become the top of stack.*



No of cycles required: **2**

## 74) POP addr

| “Pop”

Example:

**POP 25H; First  $[25H] \leftarrow [SP]$ , Then  $SP \leftarrow SP - 1$**

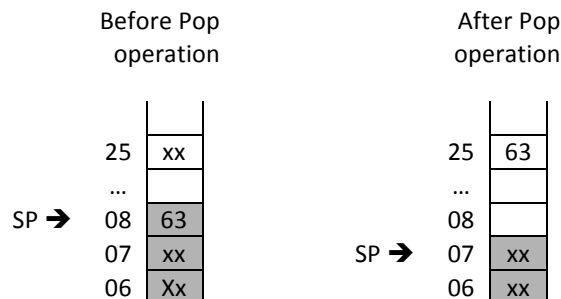
Operation:

*This instruction is used to pop data from the top of Stack and store it into the desired RAM location.*

*First data from the top of Stack will be popped and stored into the desired RAM location.*

*Then SP will become  $SP - 1$*

*The data now pointed by SP becomes the new top of stack.*



No of cycles required: **2**



**75) XCH A, Rr**

| “Exchange”

Example:

**XCH A, R0; A  $\leftrightarrow$  R0***Operation:****It will interchange the values of A Register and the specified RAM Register.****Suppose A Register contains 34H and R0 contains 67H, then after the operation, A will get 67H and R0 will get 34H.*No of cycles required: **1****76) XCH A, addr**

| “Exchange”

Example:

**XCH A, 25H; A  $\leftrightarrow$  [25H]***Operation:****It will interchange the values of A Register and the contents of the specified RAM location.****Suppose A Register contains 34H and location 25H contains 67H, then after the operation, A will get 67H and location 25H will get 34H.*No of cycles required: **1****77) XCH A, @Rp**

| “Exchange”

Example:

**XCH A, @R0; A  $\leftrightarrow$  [R0]***Operation:****It will interchange the values of A Register and the contents of the location pointed by the Register.****Suppose A Register contains 34H and location pointed by R0 contains 67H, then after the operation, A will get 67H and location pointed by R0 will get 34H.*No of cycles required: **1****78) XCHD A, @Rp**

| “Exchange Digit”

Example:

**XCHD A, @R0; A Lower Nibble  $\leftrightarrow$  [R0] Lower Nibble***Operation:****It will interchange only the lower nibbles of A Register and the contents of the location pointed by the Register.****Since only one digit is exchanged it is called Exchange Digit.**Suppose A Register contains 34H and location pointed by R0 contains 67H, then after the operation, A will get 37H and location pointed by R0 will get 64H.*No of cycles required: **1**