**Chapter 4: INSTRUCTION SETS**

**Topic – 1: Addressing Modes**

**Introduction**

* **Addressing modes:** The procedure of **accessing memory** using instructions.
* Means the **ways** or **choices** a programmer gets to access a memory different way.

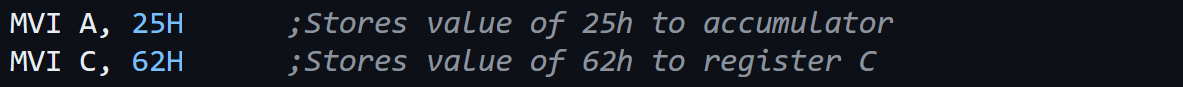
**Various Addressing Modes**

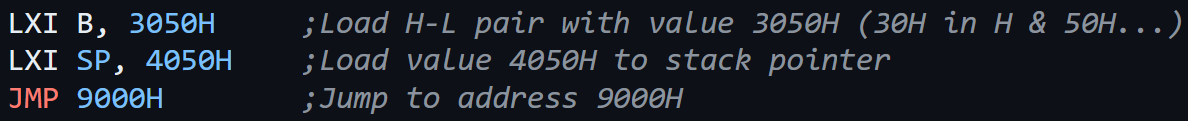
* Immediate addressing mode
* Register addressing mode
* Direct addressing mode
* Register indirect addressing mode
* Implied addressing mode

**Topic – 2: Addressing Modes (Descriptions)**

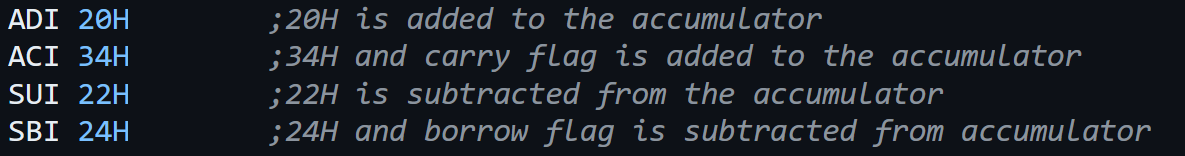
**Immediate Addressing Mode**

* Number **directly** moves to the location mentioned by the programmer.





* ***\*There is a mistake above, B must be M or HL\****
* In **HL** pair, **H** is **higher bit** & **L** is **lower bit**.
* When providing an **8-bit** number to a register pair, the **higher bit** is assumed to be **00H**.

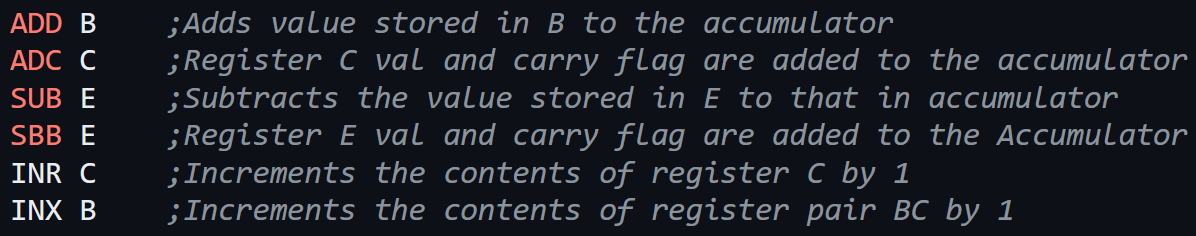


* Instructions using immediate addressing mode are:
  + **LXI**
  + **MVI**
  + **ADI**
  + **ACI**
  + **SUI**
  + **SBI**
  + **CPI**
  + **ANI**
  + **XRI**
  + **ORI**
  + **JMP**
* All immediate addressing mode keywords end with **'I'** except **JMP**.

**Register Addressing Mode**

* Data to be accessed or moved is **present in a register**.

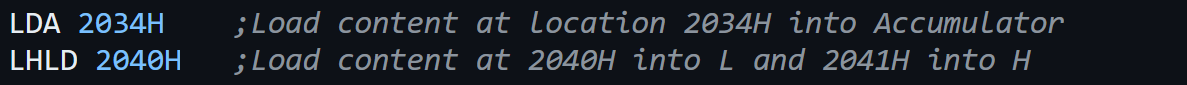


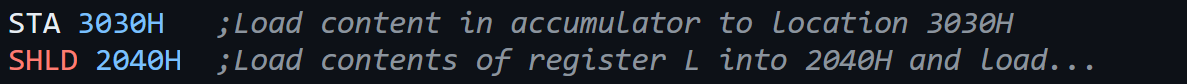


* Instructions using register addressing mode are:
  + **MOV**
  + **ADD**
  + **ADC**
  + **SUB**
  + **SBB**
  + **INR**
  + **INX**
  + **DCR**
  + **DCX**
  + **CMP**
  + **ANA**
  + **XRA**
  + **ORA**

**Direct Addressing Mode**

* We access data stored at a certain memory location using **its address**.

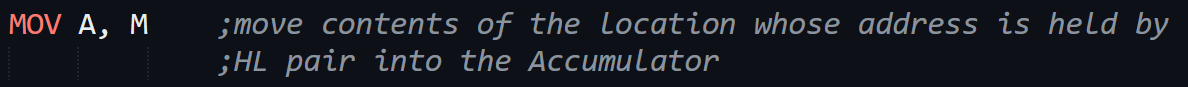




* Instructions using register addressing mode are:
  + **LDA**
  + **LHLD**
  + **STA**
  + **SHLD**

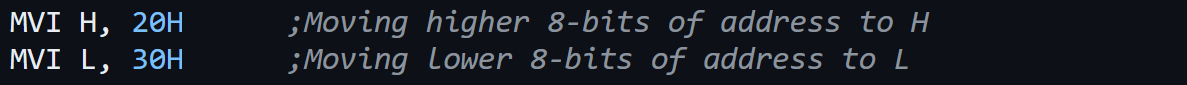
**Register Indirect Addressing Mode**

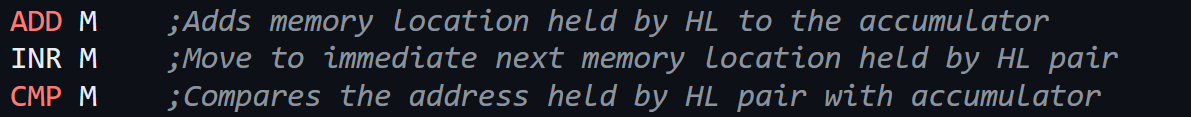
* **Registers contain address** from where data has to be accessed.



* **H** holds the **higher** **8-bits** of the address & **L** does so with **lower** **8-bits**.
* **M** represents/denotes the **HL** pair.

Means:-

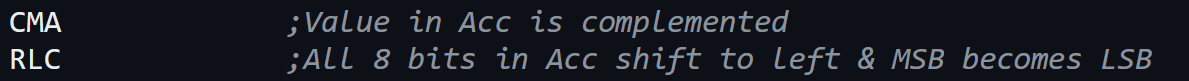


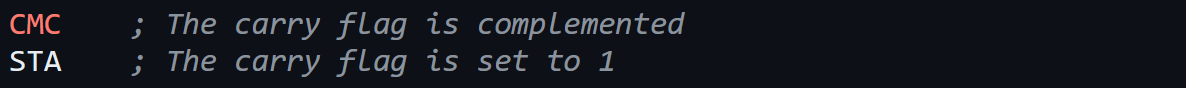


* For CMP,
  + If **A < M** then **carry flag** is set
  + If **A = M** then **zero flag** is set
  + If **A > M** then **carry** & **zero flags** are reset
* Instructions using register addressing mode are:
  + **MOV**
  + **MVI**
  + **ADD**
  + **ADC**
  + **SUB**
  + **SBB**
  + **INR**
  + **DCR**
  + **CMP**
  + **ANA**
  + **XRA**
  + **ORA**

**Implied Addressing Mode**

* Register **already knows** the address of data.





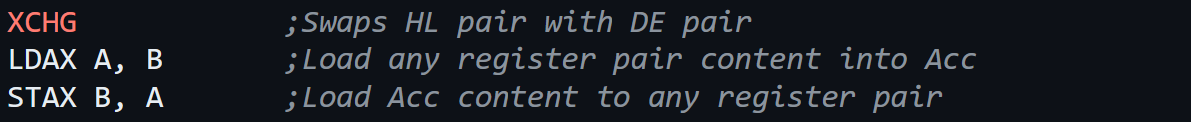
* Instructions using **register addressing mode** are:
  + **RLC**
  + **RRC**
  + **RAL**
  + **RAR**
  + **CMA**
  + **CMC**
  + **STC**

**Topic – 3: Instruction Set**

**Introduction**

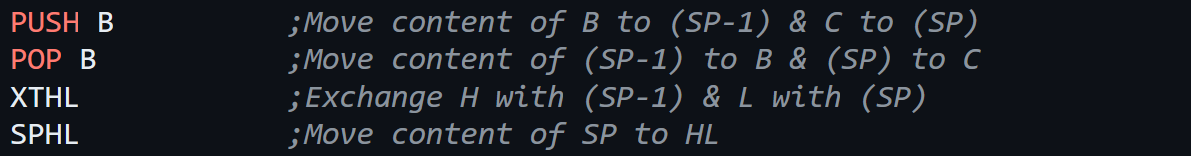
* **Instruction set:** All the **instructions** supported by a microprocessor.
* Intel’s 8085 supports **246** instructions.
* Each of these instructions are denoted by an **8-bit** **binary** value.
* These **8-bit** binary values are known as **op-codes**.
* **Opcode** is first part of the **instruction** & **operand** is the **second**.
* **Instruction word size:** Size of machine code.
* Minimum instruction word size of **8085** is **1 byte**.

**More Missed Instructions**



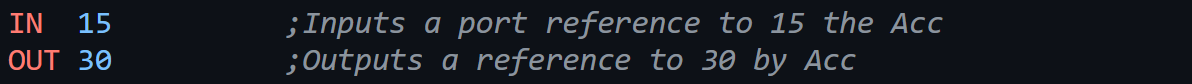
* A **register pair** in instruction where a **pair** is expected to be written, is represented by pair’s **first register**.

**Stack Operations**



* **Address** of a memory location is of **16-bits**.

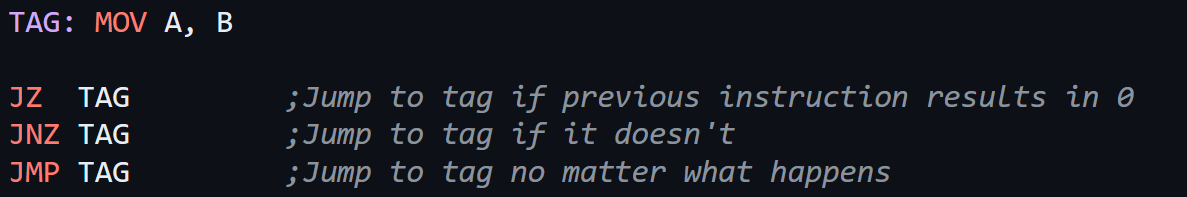
**I/O Data Transfer**



**Size of Various Instructions**

* **1 byte –** **MOV**, **STAX**, **XCHG**, **PUSH**, **POP**, **XTHL**, **SPHL**
* **2 bytes –** **MVI**, **IN**, **OUT**
* **3 bytes –** **LXI**, **LDA**, **LDAX**, **LHLD**, **STA**, **SHLD**

**Jump Instructions**

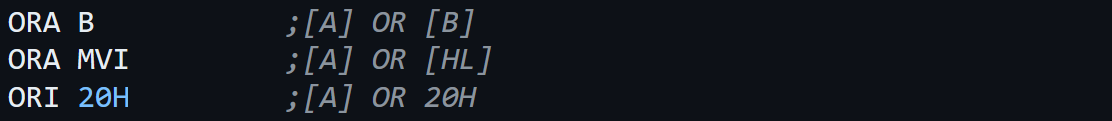


**Topic – 4: Operational Instructions**

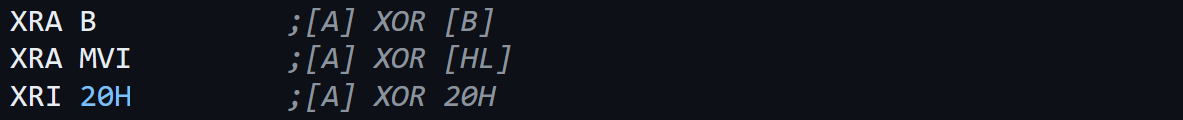
**AND**



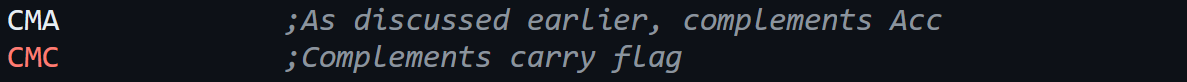
**OR**



**XOR**



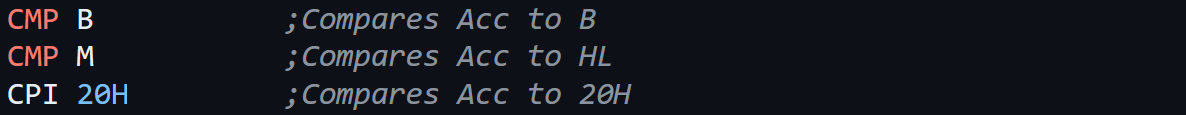
**Complement**



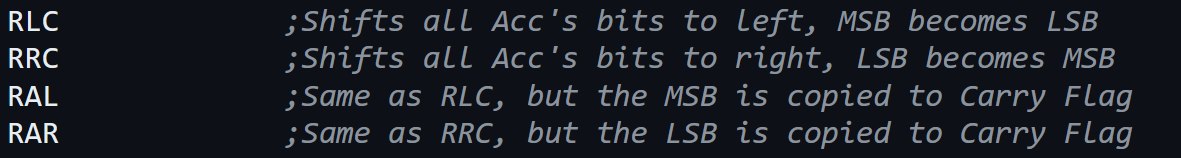
**Set**



**Comparison**



**Rotations**



**Additional Jump Instructions**

