Addressing Modes of 8085

- The microprocessor has **different ways of specifying the data or operand** for the instruction.

- The **various formats of specifying operands** are called **addressing modes**

- The 8085 has **Five addressing modes**:

- 1) **Register Addressing mode**: This type of addressing mode specifies register or register pair that contains data

Addressing Modes of 8085

- 2) **Immediate Addressing Mode:**

- In this type of addressing mode, **immediate data byte is provided with the instruction.**

- Example: MVI A,47H
- LXI H, 4100H etc.

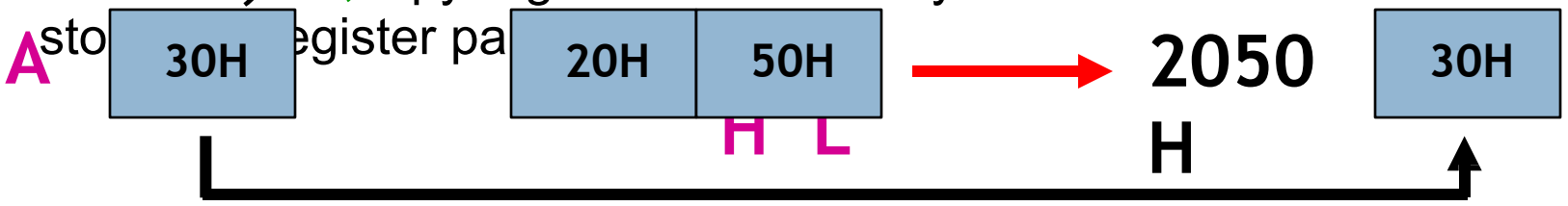
Memory Addressing

- One of the operands is a memory location
- Depending on **how address of memory location is specified**, **memory** addressing is of two types
  - **Direct** addressing
  - **Indirect** addressing

- 3) **Direct Addressing Mode: In this type of addressing mode, the 16bit memory address is** directly provided with the instruction.
- Example: LDA C500H
- **STA 3050H** etc

Addressing Modes of 8085

- 4) **Indirect Addressing Mode: In this type of addressing mode, the 16bit memory address** is indirectly provided with the instruction using a register pair
- Example: LDAX B
- **(Load the accumulator with the contents of the memory location whose address is stored in the register pair BC)**
- **MOV M, A** ;copy register A to memory location whose address is sto egister pa



Addressing Modes of 8085

1

- 5) **Implied or implicit Addressing mode:** In this type of addressing mode, **No operand (register or data) is specified in the instruction.**

- The **operand is inborn to the instruction**.

- Example: CMA (Complement Accumulator) , RAL (rotated accumulator left), etc

Instruction Format

Instruction Format

- An **instruction** is a command to the microprocessor to perform a given task on a specified data.
- Each instruction has two parts: one is task to be performed, called the **operation code** (opcode), and the second is the data to be operated on, called the **operand.**
- The operand (or data) can be specified in various ways.
- It may include 8-bit (or 16-bit) data, an internal register, a memory location, or 8-bit (or 16-bit) address.
- In some instructions, the operand is implicit.

Instruction word size

- The 8085 instruction set is classified into the following three groups according to word size:

1. One-word or 1-byte instructions
2. Two-word or 2-byte instructions
3. Three-word or 3-byte instructions

- In the 8085, "byte" and "word" are synonymous because it is an 8-bit microprocessor.
- However, instructions are commonly referred to in terms of bytes rather than words.

One-Byte Instructions

- A 1-byte instruction includes the opcode and operand in the same byte.
- Operand(s) are internal register and are coded into the instruction.

Task	Op code	Operand	Binary Code	Hex Code
Copy the contents of the accumulator in the register C.	MOV	C,A	0100 1111	4FH
Add the contents of register B to the contents of the accumulator.	ADD	B	1000 0000	80H
Invert (compliment) each bit in the accumulator.	CMA		0010 1111	2FH

One-Byte Instructions

- In the first instruction, both operand registers are specified.
- In the second instruction, the operand B is specified and the accumulator is assumed.
- the third instruction, the accumulator is assumed to be the implicit operand
- These instructions are stored in 8-bit binary format in memory; each requires one memory location.

One-Byte Instructions Example

- MOV rd, rs

rd <-- rs copies contents of rs into rd.

- Example: MOV A,B

Coded as 01111000 = 78H

ADD r

A <-- A + r

Two-Byte Instructions

- In a two-byte instruction, the first byte specifies the operation code and the second byte specifies the operand.
- Source operand is a data byte immediately following the opcode

Task	Opcode	Operand	Binary Code	Hex Code	
Load an 8-bit data byte in the accumulator.	MVI	A, Data	0011 1110 DATA	3E Data	First Byte Second Byte

Two-Byte Instructions

- Assume that the data byte is 32H. The assembly language instruction is written as

Mnemonics	Hex code
MVI A, 32H	3E 32H

- The instruction would require two memory locations to store in memory.

MVI r,data
r <-- data
Example: MVI A,30H coded as 3EH 30H as two contiguous bytes. This is an example of immediate addressing.

ADI data
A <-- A + data

Three-Byte Instructions

- In a three-byte instruction, the first byte specifies the opcode, and the following two bytes specify the 16-bit address.
- The second byte is the low-order address and the third byte is the high-order address.
- opcode + data byte + data byte

Task	Opcode	Operand	Binary code	Hex Code	
Transfer the program sequence to the memory location 2085H.	JMP	2085H	1100 0011 1000 0101 0010 0000	C3 85 20	First byte Second Byte Third Byte

Three-Byte Instructions

- This instruction would require three memory locations to store in memory.
- Three byte instructions - opcode + data byte + data byte
- LXI rp, data16
- rp is one of the pairs of registers BC, DE, HL used as 16-bit registers.
- The two data bytes are 16-bit data in L H order of significance.
- rp <-- data16

Three-Byte Instructions

Example:
LXI H,0520H coded as 21H 20H 50H in three bytes. This is also immediate
addressing.

LDA addr
A <-- (addr) Addr is a 16-bit address in L H order. Example:
LDA 2134H coded as
3AH 34H 21H. This is also an example of direct addressing.

References

1. Gaonkar, R. S. (1990). *Microprocessor Architecture, Programming and Applications with the 8085*. Fifth Edition Prentice Hall PTR.