

PROGRAMMING WITH 8085  
LECTURE 3

INTRODUCTION  
TO 8085  
INSTRUCTIONS

# Writing a Assembly Language Program

- Steps to write a program
  - **Analyze** the problem
  - Develop program **Logic**
  - Write an **Algorithm**
  - Make a **Flowchart**
  - Write program **Instructions** using Assembly language of 8085
  - Start troubleshooting i.e. debugging a program if error occurs.

Program **8085** in Assembly language to add two 8-bit numbers and store 8-bit result in register **C**.

## 1. Analyze the problem

- Addition of two 8-bit numbers to be done

## 2. Program Logic

- Add two numbers
- Store result in register **C**
- Example

00111001 (39H) **D**

10011001 (99H) **E**

11010010 (D2H) **C**

# Algorithm

1. Get two numbers
2. Add them
3. Store result
4. Stop

## Translation to 8085 operations

- Load 1<sup>st</sup> no. in register D
- Load 2<sup>nd</sup> no. in register E
- Copy register D to A
- Add register E to A
- Copy A to register C
- Stop processing

# Assembly Language Program

1. Get two numbers

- a) Load 1<sup>st</sup> no. in register D
- b) Load 2<sup>nd</sup> no. in register E

2. Add them

- a) Copy register D to A
- b) Add register E to A

3. Store result

- a) Copy A to register C

4. Stop

- a) Stop processing

```
MVI D, 2H
```

```
MVI E, 3H
```

```
MOV A, D
```

```
ADD E
```

```
MOV C, A
```

```
HLT
```

Program 8085 in Assembly language to add two 8-bit numbers. Result can be more than 8-bits.

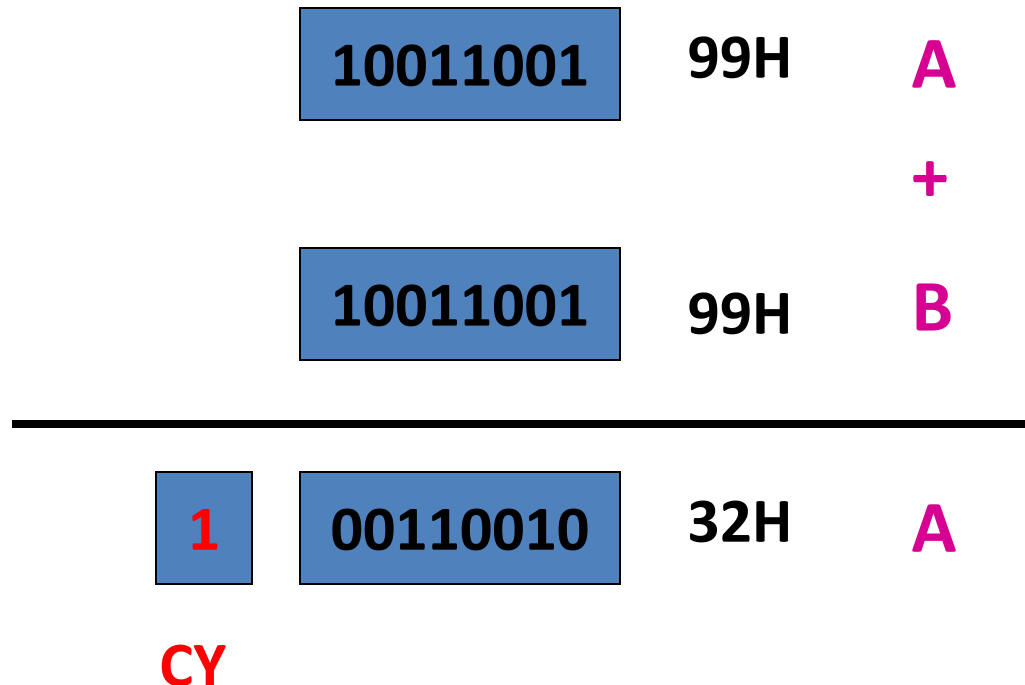
## 1. Analyze the problem

- Result of addition of two 8-bit numbers can be 9-bit
- Example

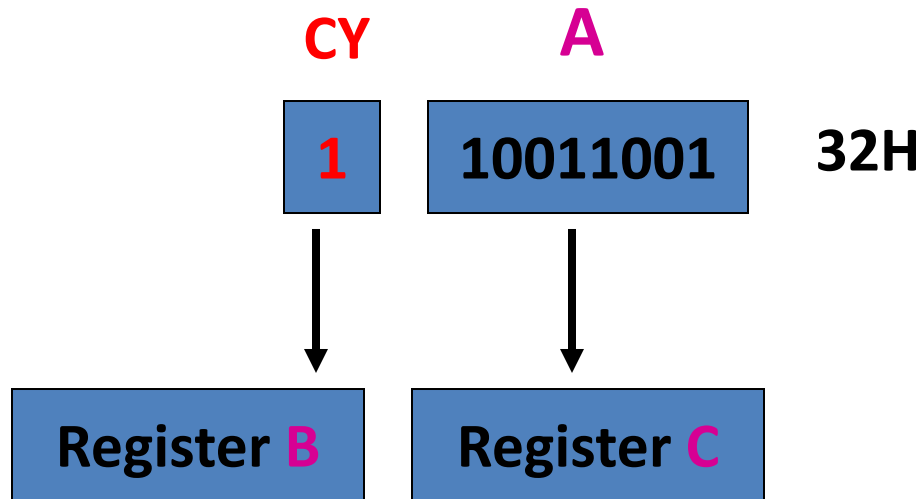
```
    10011001  (99H) A
+   10011001  (99H) B
-----
   100110010 (132H)
```

- The 9<sup>th</sup> bit in the result is called CARRY bit.

- How 8085 does it?
  - Adds register **A** and **B**
  - Stores 8-bit result in **A**
  - SETS carry flag (CY) to indicate carry bit



- Storing result in Register memory



Step-1 Copy A to C

Step-2

- Clear register B
- Increment B by 1



## 2. Program Logic

1. Add two numbers
2. Copy 8-bit result in A to C
3. If CARRY is generated
  - Handle it
4. Result is in register pair BC

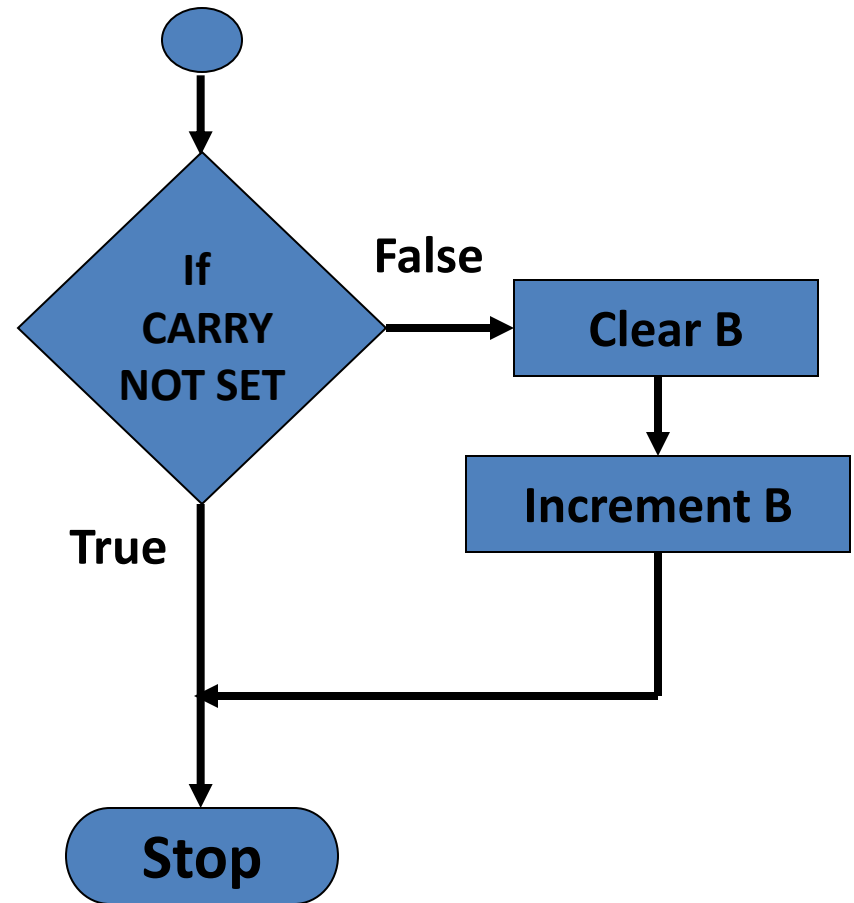
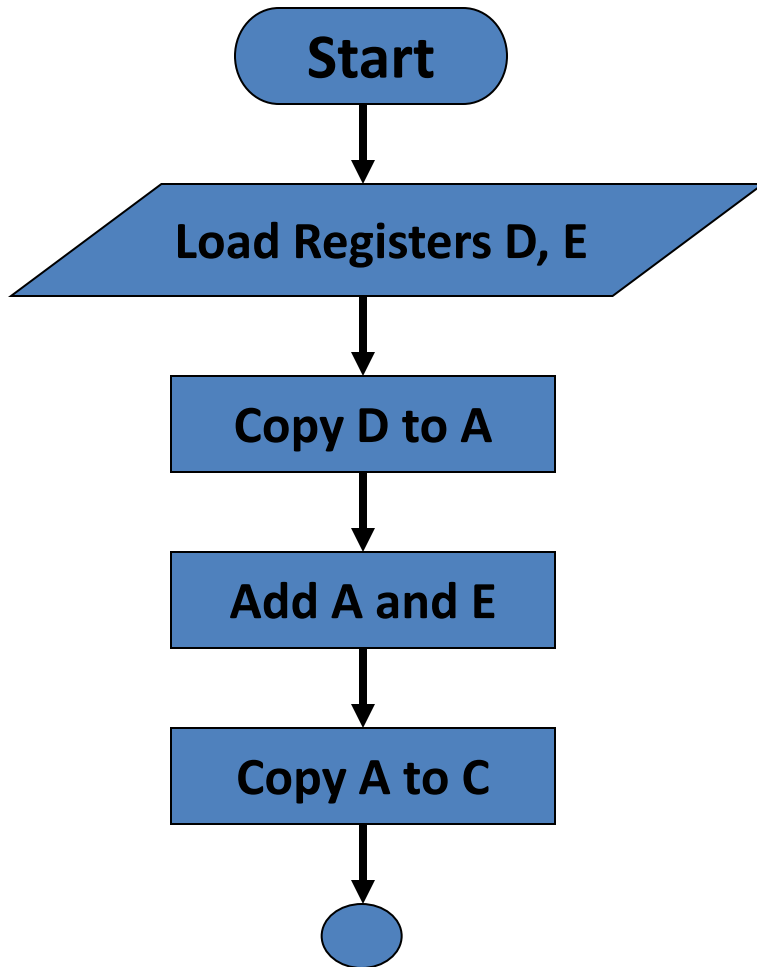
### 3. Algorithm

1. Load two numbers in registers D, E
2. Add them
3. Store 8 bit result in C
4. Check CARRY flag
5. If CARRY flag is SET
  - Store CARRY in register B
6. Stop

### Translation to 8085 operations

- Load registers D, E
- Copy register D to A
- Add register E to A
- Copy A to register C
- Use Conditional Jump instructions
- Clear register B
- Increment B
- Stop processing

## 4. Make a Flowchart



# 5. Assembly Language Program

- Load registers D, E
- Copy register D to A
- Add register E to A
- Copy A to register C
- Use Conditional Jump instructions
- Clear register B
- Increment B
- Stop processing

```
MVI D, 2H
MVI E, 3H
MOV A, D
ADD E
MOV C, A
JNC END
MVI B, 0H
INR B
END: HLT
```

# Addressing Modes of 8085

- Format of a typical Assembly language instruction is given below-

[Label:] Mnemonic [Operands] [;comments]

HLT

MVI A, 20H

MOV M, A ;Copy A to memory location whose address is stored in register pair HL

LOAD: LDA 2050H ;Load A with contents of memory location with address 2050H

READ: IN 07H ;Read data from Input port with address 07H

- The various formats of specifying operands are called addressing modes
- Addressing modes of 8085
  1. Register Addressing
  2. Immediate Addressing
  3. Memory Addressing
  4. Input/Output Addressing

# 1. Register Addressing

- Operands are one of the internal registers of 8085
- Examples-

**MOV A, B**

**ADD C**

## 2. Immediate Addressing

- Value of the operand is given in the instruction itself
- Example-

**MVI A, 20H**

**LXI H, 2050H**

**ADI 30H**

**SUI 10H**



### 3. **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(a) Direct Addressing

- **16-bit** Address of the memory location is specified in the instruction directly
- Examples-

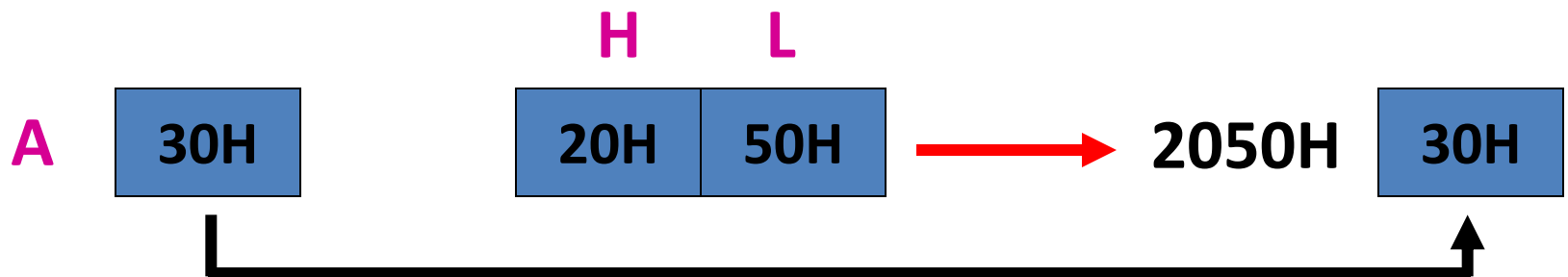
**LDA 2050H** ;load A with contents of memory location with address 2050H

**STA 3050H** ;store A with contents of memory location with address 3050H

## 3(b) Indirect Addressing

- A **memory pointer** register is used to store the address of the memory location
- Example-

**MOV M, A** ; copy register A to memory location whose address is stored in register pair HL



## 4. **Input/Output** Addressing

- **8-bit** address of the port is directly specified in the instruction
- Examples-

**IN 07H**

**OUT 21H**