Experiment 2

## **AIM:**

Addition of two 8 bit numbers and

1. Sum is also 8 bit number
2. Sum is 16 bit number

## **SOFTWARE USED :**

GNUSim 8085

## **THEORY:**

The Intel 8085 is an 8-bit microprocessor produced by Intel and introduced in March 1976. It is a software-binary compatible with the more-famous Intel 8080 with only two minor instructions added to support its added interrupt and serial input/output features.

### **INSTRUCTIONS IN 8085:**

An instruction of a computer is a command given to the computer to perform a specified operation on given data. In microprocessor, the instruction set is the collection of the instructions that the microprocessor is designed to execute.

The programmer writes a program in assembly language using these instructions. These instructions have been classified into the following groups:

1. **Data Transfer Group**
2. **Arithmetic Group**
3. **Logical Group**
4. **Branch Control Group**
5. **I/O and Machine Control Group**

These are briefly explained below

1. **Data Transfer Group**

Instructions, which are used to transfer data from one register to another register, from memory to register or register to memory, come under this group. Examples are: MOV, MVI, LXI, LDA, STA etc. When an instruction of data transfer group is executed, data is transferred from the source to the destination without altering the contents of the source

1. **Arithmetic Group**

The instructions of this group perform arithmetic operations such as addition, subtraction; increment or decrement of the content of a register or memory. Examples are: ADD, SUB, INR, DAD etc**.**

1. **Logical Group**

The Instructions under this group perform logical operation such as AND, OR, compare, rotate etc. Examples are: ANA, XRA, ORA, CMP, and RAL etc.

1. **Branch Control Group**

This group includes the instructions for conditional and unconditional jump, subroutine call and return, and restart. Examples are: JMP, JC, JZ, CALL, CZ, RST etc.

1. **I/O and Machine Control Group**

This group includes the instructions for input/output ports, stack and machine

control. Examples are: IN, OUT, PUSH, POP, and HLT etc

## **ALGORITHM**

1. Sum is 8 bit

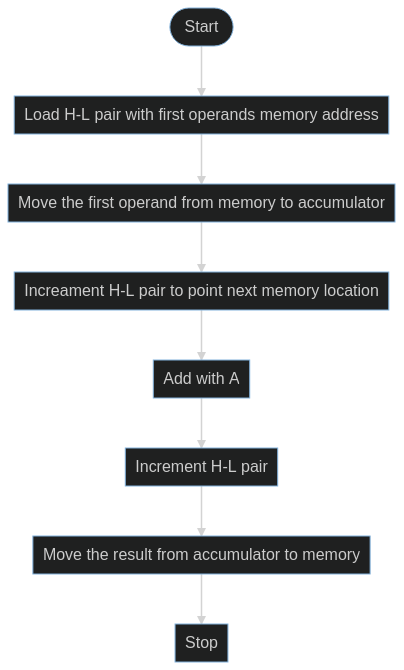
* Get address of first number in H-L pair
* First number in accumulator
* Increment content of H-L pair
* Add first and second number
* Store sum in 2503 H
* Stop

1. Sum is 16 bit

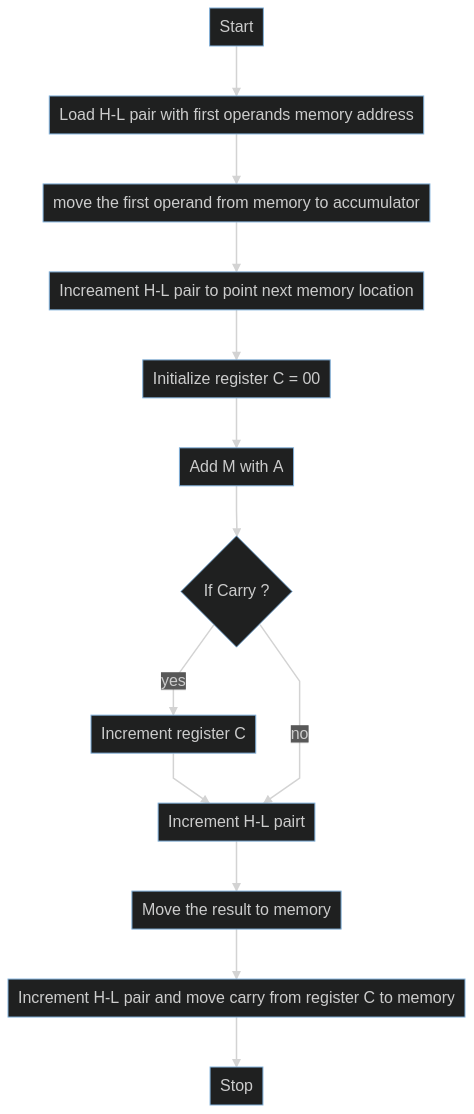
* Address of first number in H-L pair
* MSBs of sum in in register C. Initial value=00
* First number in accumulator
* Address of Second number in H-L pair
* Add the two numbers
* Is carry? If no, go to label AHEAD
* If yes, increment C
* LSB of sum in 2503 H
* MSB of sum in accumulator
* MSB of sum in 2504 H
* Stop

## **FLOW CHART:**

1. Sum is 8 bit



1. Sum is 16 bit



## **PROGRAM AND OUTPUT:**

## a) Sum is 8 bit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Add | Mnemo | Op | Comments | Addressing Mode | Machine Cycle | T-States |
| 2000 | LXI H | 2501H | Load H-L pairs with address 2501H | Immediate | Opcode fetch + 2 Memory read | 10 |
| 2003 | MOV A, M |  | Move first operand from memory to accumulator | Indirect | Opcode fetch + Memory read | 7 |
| 2004 | INX H |  | Increment H-L pair | Register | Opcode fetch | 6 |
| 2005 | ADD M |  | Add M with A | Register | Opcode fetch + Memory read | 7 |
| 2006 | INX H |  | Increment H-L pair | Register | Opcode fetch | 6 |
| 2007 | MOV M, A |  | Move contents of Accumulator to memory add sotred in H-L Pair | Indirect | Opcode fetch + memory write | 7 |
| 2008 | HLT |  | HALT | - | Opcode fetch | 4 |

## Output

### Data:

2501- 04 H  
  
 2502- 02 H

### Result:

2503- 06 H

## b) Sum is 16 bit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Add | Mn | Op | Comments | Addressing Mode | Machine Cycle | T-States |
| 2000 | LXI H | 2501 H | Load H-L pair with Address 300H | Immediate | Opcode fetch and 2 memory read | 10 |
| 2003 | MVI C | 00 | Initialize C register with zero value | Immediate | Opcode fetch + memory read | 7 |
| 2005 | MOV A, M |  | Move first operand form memory to A | Indirect | Opcode fetch + memory read | 7 |
| 2006 | INX H |  | Increment H-L Pair | Register | Opcode fetch | 6 |
| 2007 | ADD M |  | Add M with A | Indirect | Opcode fetch + memory read | 7 |
| 2008 | JNC | AHEAD | Jump to label if no carry | Immediate | Opcode fetch + 2 memory read | 10 |
| 2008 | INR | C | Increment reg C | Register | Opcode fetch | 4 |
| 2000 | STA | 2503H | Load data of accumulator to memory | Direct | Opcode fetch + 2 memory read + 2 memory write | 13 |
| 200F | MOV A, C |  | Move content of C register to accumulator | Register | Opcode fetch | 4 |
| 2010 | STA | 2504H | Load data of accumulator to memory | Direct | Opcode fetch + 2 memory read + 2 memory write | 13 |
| 2013 | HLT |  | HALT | - | Opcode | 4 |

## Output

### Data:

2501- 98 H  
  
 2502- 9A H

### Result:

2503- 32 H, LSBs of sum  
  
 2504- 01 H, MSBs of sum

## Result:

The sum of two 8 bit numbers in 8085 is obtained for both the cases when

1. Sum of the numbers is 8 bit
2. Sum of the numbers is 16 bit.