**Experiment 1.2**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 807/B

**Semester:** 4th **Date of Performance:** 08/03/2022

**Subject Name:** MPI Lab **Subject Code:** 22E-20CSP-253

**1. Aim/Overview of the practical:**

Addition of two 16-bits numbers.

**2. Task to be done/ Which logistics used:**

To add a two 16-bit numbers in 8085 microprocessor.

**3. Example:**

Input data of 1st no.→

Memory Address→

Memory Address→

|  |  |
| --- | --- |
| **66** | **45** |
| **2051** | **2050** |

Input data of 2nd no.→

Memory Address→

Memory Address→

|  |  |
| --- | --- |
| **33** | **22** |
| **2053** | **2052** |

**Carry**

**↓**

Output data→

Memory Address→

Memory Address→

|  |  |  |
| --- | --- | --- |
| **00** | **99** | **67** |
| **3052** | **3051** | **3050** |

1. **Addition of 16 bit numbers using 8 bit operation –** It is a lengthy method and requires more memory as compared to 16 bit operation.

**4. Algorithms:**

* Load the lower part of first number in B register
* Load the lower part of second number in A (accumulator)
* Add both the numbers and store
* Load the higher part of first number in B register
* Load the higher part of second number in A (accumulator)
* Add both the numbers with carry from the lower bytes (if any) and store at the next location

**5. Programs:**

|  |  |  |
| --- | --- | --- |
| **Memory Address** | **Mnemonics** | **Comment** |
| 2000 | LDA 2050 | A<-2050 |
| 2003 | MOV B, A | B<-A |
| 2004 | LDA 2052 | A<-2052 |
| 2007 | ADD B | A<-A+B |
| 2008 | STA 3050 | A->3050 |
| 200B | LDA 2051 | A<-2051 |
| 200E | MOV B, A | B<-A |
| 200F | LDA 2053 | A<-2053 |
| 2012 | ADC B | A<-A+B+CY |
| 2013 | STA 3051 | A->3051 |
| 2016 | HLT | Stops execution |

**6. Explanation:**

* **LDA 2050** moves the contents of 2050 memory in A accumulator.
* **MOV B, A**stores the value of A into B register
* **LDA 2052** stores the value at 2052 in A
* **ADD B**add the contents of B and A and store in A
* **STA 3050** stores the result in memory location 3050
* **LDA 2051** stores the value at 2051 in A
* **MOV B, A**stores the value of A into B register
* **LDA 2053**  stores the value at 2053 in A
* **ADC B**add the contents of B, A and carry from the lower bit addition and store in A
* **STA 3051** stores the result in memory location 3051
* **HLT**stops execution.

1. **Addition of 16 bit numbers using 16 bit operation –** It is a very short method and less memory is also required as compared to 8 bit operation.

**7. Algorithm:**

* Load both the lower and the higher bits of first number at once
* Copy the first number to another register pair
* Load both the lower and the higher bits of second number at once
* Add both the register pairs and store the result in a memory location.

**8. Program:**

LHLD 1000

XCHG

LHLD 1002

MVI C,00

DAD D

JNC LABEL

INR C

MOV A,C

STA 1006

LABEL: SHLD 1004

HLT

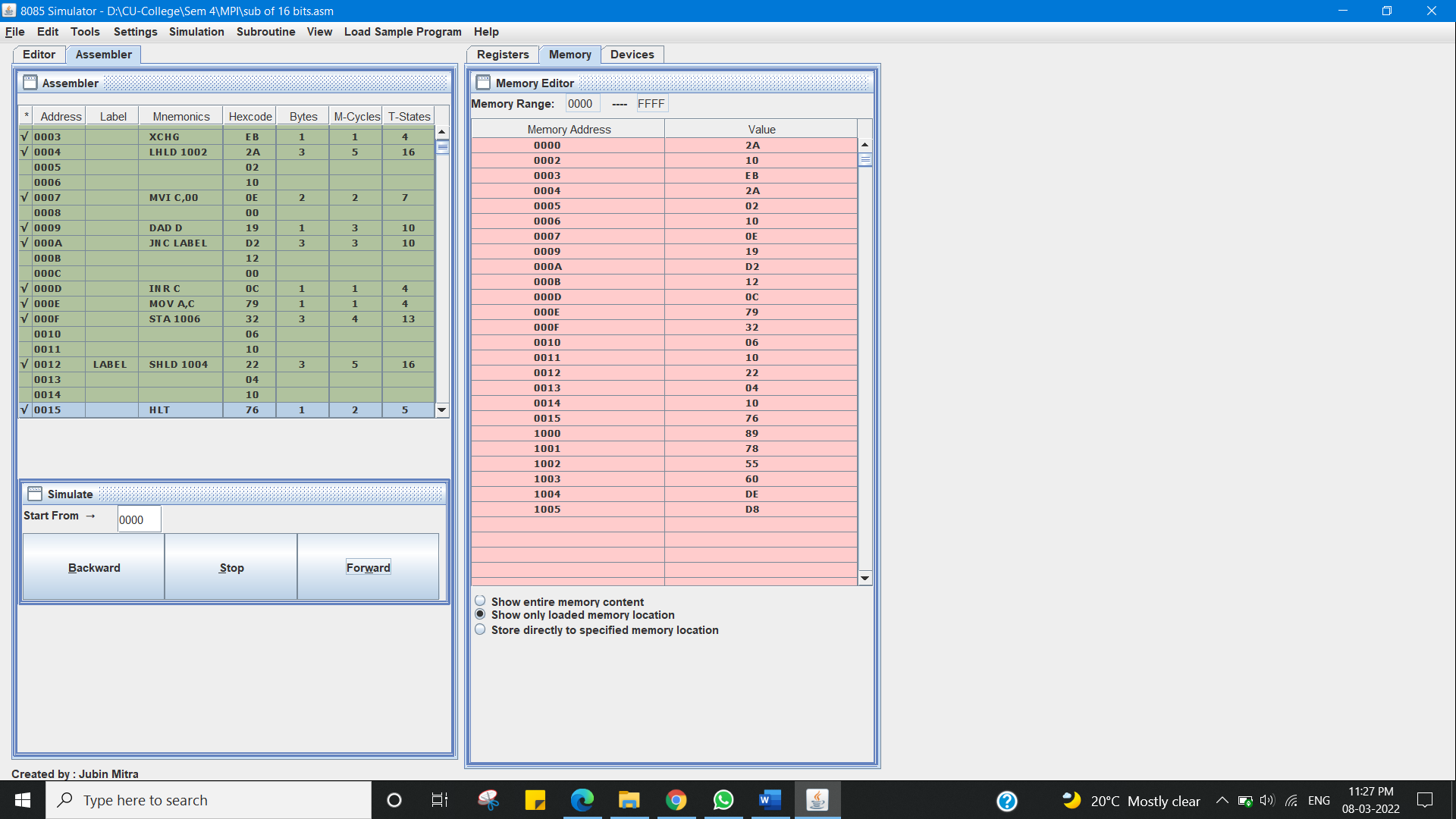
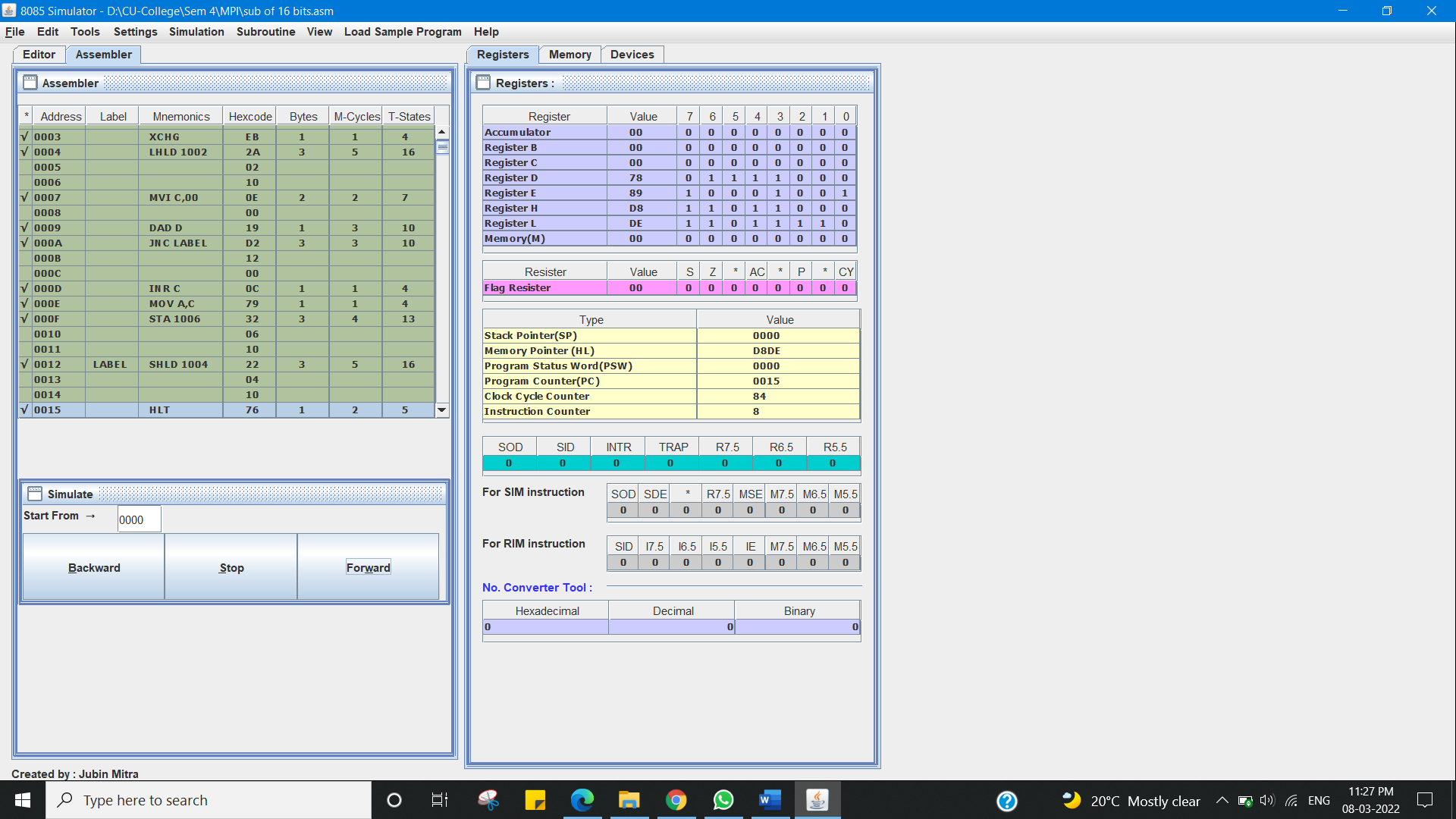
# ORG 1000

# DB 89,78,55,60

|  |  |  |
| --- | --- | --- |
| **Memory Address** | **Mnemonics** | **Comments** |
| 2000 | LHLD 2050 | H-L<-2050 |
| 2003 | XCHG | D<->H & E<->L |
| 2004 | LHLD 2052 | H-L<-2052 |
| 2007 | DAD D | H<-H+D & L<-L+E |
| 2008 | SHLD 3050 | A->3050 |
| 200B | HLT | Stops execution |

**9. Explanation:**

* **LHLD 2050** loads the value at 2050 in L register and that in 2051 in H register (first number)
* **XCHG** copies the content of H to D register and L to S register
* **LHLD 2052** loads the value at 2052 in L register and that in 2053 in H register (second number)
* **DAD D** adds the value of H with D and L with E and stores the result in H and L
* **SHLD 3050** stores the result at memory location 3050
* **HLT** stops execution

**10. Result/Output/Writing Summary:**

I have successfully completed this experiment.

**Learning outcomes (What I have learnt):**

1. Addition of two 16-bits numbers.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |