# INSTITUTE OF COMPUTER TECHNOLOGY

# B-TECH COMPUTER SCIENCE ENGINEERING 2025-26

# SUBJECT: MICROCONTROLLER & APPLICATIONS

NAME: Rahul Prajapati

ENRLL. NO: 23162171020

BRANCH: CYBER SECURITY

BATCH: 52

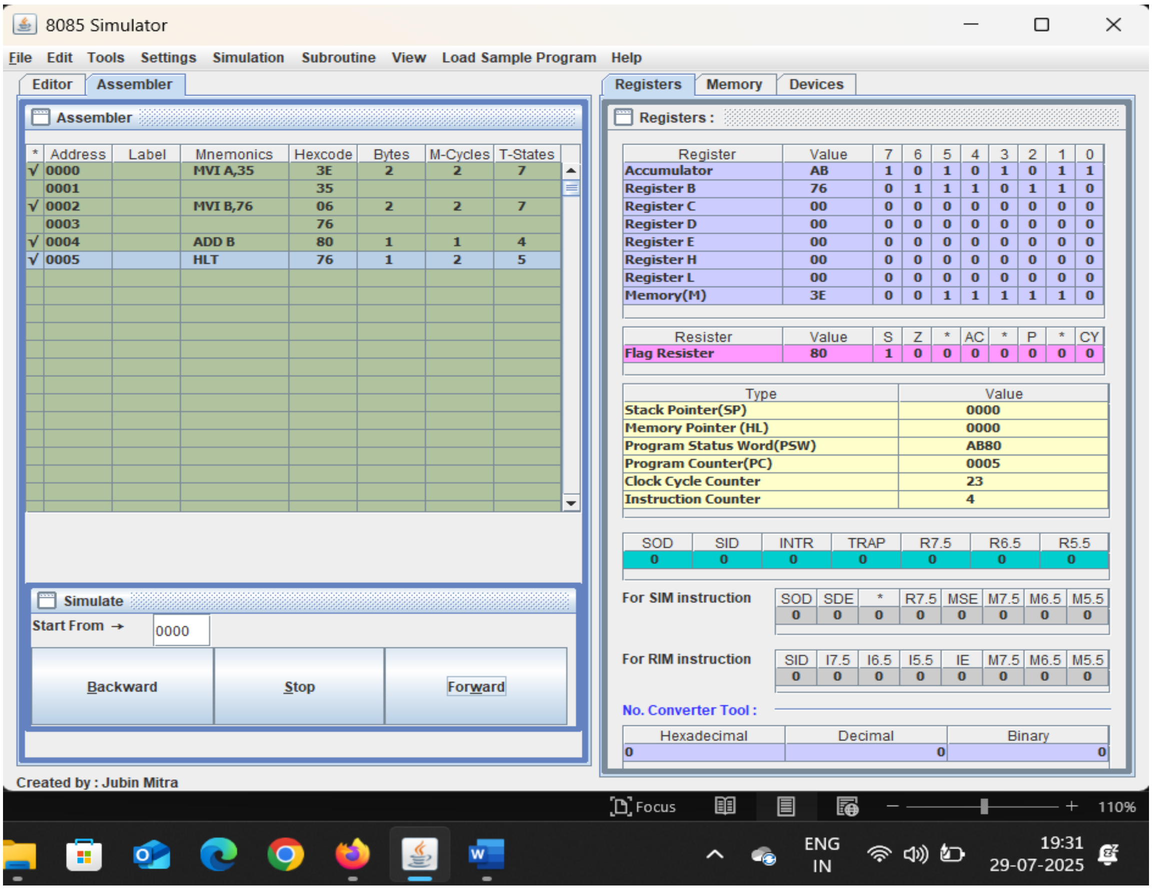
## PRACTICAL_1

## Aim: Learning Programs using Data Transfer Instructions and Arithmetic Instructions

1. Write programs for addition of two data for the following conditions and show the status of flags in eachcase.

[A] Load an immediate data 35H in register A and 76H in register B. Add them. Comment on result.

CODE:-



| * | Address | Label | Mnemonics | Hexcode | Bytes | M-Cycles | T-States |
|---|---|---|---|---|---|---|---|
| √ | 0000 | | MVI A,35 | 3E | 2 | 2 | 7 |
| | 0001 | | | 35 | | | |
| √ | 0002 | | MVI B,76 | 06 | 2 | 2 | 7 |
| | 0003 | | | 76 | | | |
| √ | 0004 | | ADD B | 80 | 1 | 1 | 4 |
| √ | 0005 | | HLT | 76 | 1 | 2 | 5 |

| Register | Value | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Accumulator | AB | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Register B | 76 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| Register C | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register D | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register E | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 3E | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 80 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | AB80 |
| Program Counter(PC) | 0005 |
| Clock Cycle Counter | 23 |
| Instruction Counter | 4 |

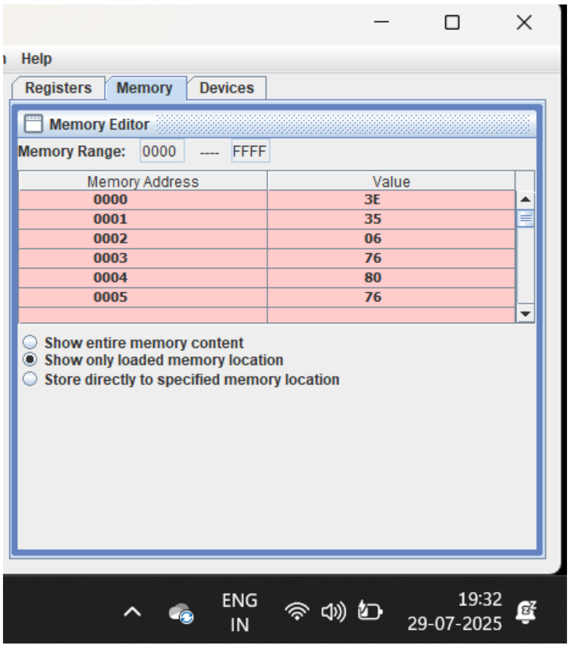
| SOD | SID | INTR | TRAP | R7.5 | R6.5 | R5.5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For SIM instruction

| SOD | SDE | * | R7.5 | MSE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



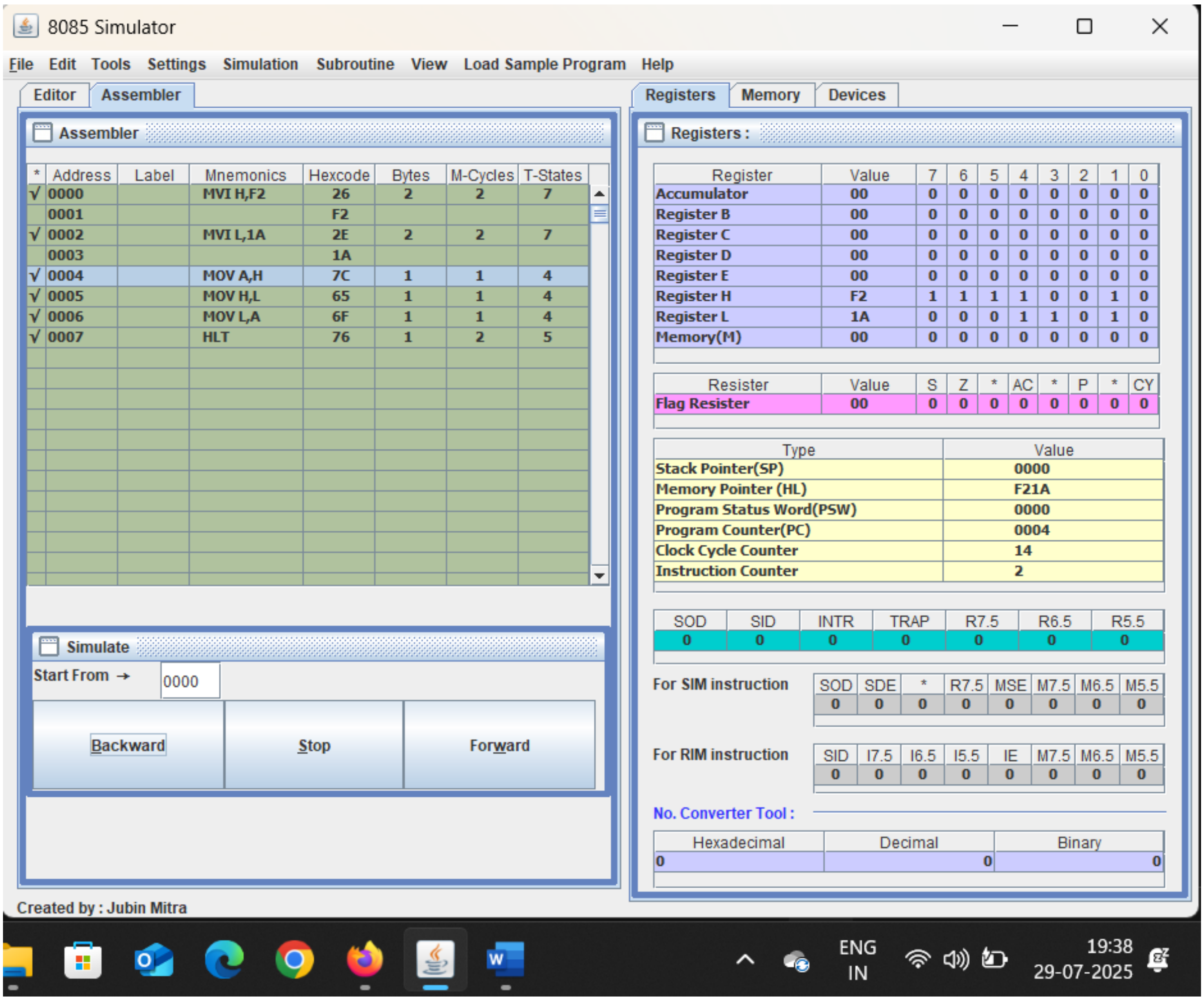
Memory Range: 0000 ---- FFFF

| Memory Address | Value |
|---|---|
| 0000 | 3E |
| 0001 | 35 |
| 0002 | 06 |
| 0003 | 76 |
| 0004 | 80 |
| 0005 | 76 |

○ Show entire memory content
● Show only loaded memory location
○ Store directly to specified memory location

[B] Write a program that swap the contents of register H and L registers.

CODE:-


8085 Simulator
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor
Assembler
Registers
Memory
Devices
Assembler

| * | Address | Label | Mnemonics | Hexcode | Bytes | M-Cycles | T-States |
|---|---|---|---|---|---|---|---|
| √ | 0000 | | MVI H,F2 | 26 | 2 | 2 | 7 |
| | 0001 | | | F2 | | | |
| √ | 0002 | | MVI L,1A | 2E | 2 | 2 | 7 |
| | 0003 | | | 1A | | | |
| √ | 0004 | | MOV A,H | 7C | 1 | 1 | 4 |
| √ | 0005 | | MOV H,L | 65 | 1 | 1 | 4 |
| √ | 0006 | | MOV L,A | 6F | 1 | 1 | 4 |
| √ | 0007 | | HLT | 76 | 1 | 2 | 5 |

Start From → 0000

| Register | Value | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Accumulator | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register B | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register C | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register D | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register E | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | F2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Register L | 1A | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | F21A |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 0004 |
| Clock Cycle Counter | 14 |
| Instruction Counter | 2 |

| SOD | SID | INTR | TRAP | R7.5 | R6.5 | R5.5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For SIM instruction

| SOD | SDE | * | R7.5 | MSE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

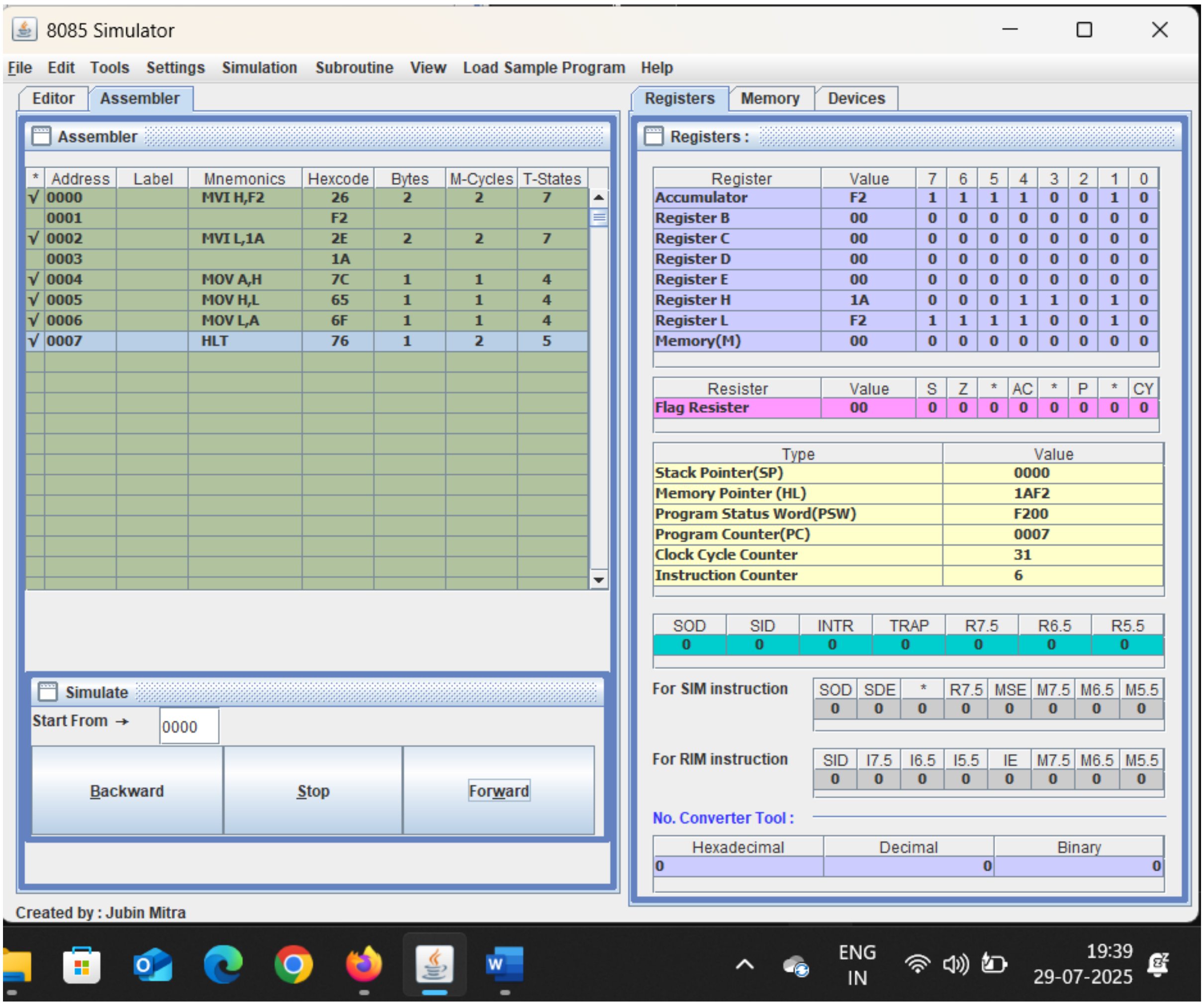
For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

No. Converter Tool :

| Hexadecimal | Decimal | Binary |
|---|---|---|
| 0 | 0 | 0 |

Created by : Jubin Mitra



Assembler

| * | Address | Label | Mnemonics | Hexcode | Bytes | M-Cycles | T-States |
|---|---|---|---|---|---|---|---|
| √ | 0000 | | MVI H,F2 | 26 | 2 | 2 | 7 |
| | 0001 | | | F2 | | | |
| √ | 0002 | | MVI L,1A | 2E | 2 | 2 | 7 |
| | 0003 | | | 1A | | | |
| √ | 0004 | | MOV A,H | 7C | 1 | 1 | 4 |
| √ | 0005 | | MOV H,L | 65 | 1 | 1 | 4 |
| √ | 0006 | | MOV L,A | 6F | 1 | 1 | 4 |
| √ | 0007 | | HLT | 76 | 1 | 2 | 5 |

Simulate

Start From → 0000

Registers :

| Register | Value | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Accumulator | F2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Register B | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register C | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register D | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register E | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 1A | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| Register L | F2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 1AF2 |
| Program Status Word(PSW) | F200 |
| Program Counter(PC) | 0007 |
| Clock Cycle Counter | 31 |
| Instruction Counter | 6 |

| SOD | SID | INTR | TRAP | R7.5 | R6.5 | R5.5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For SIM instruction

| SOD | SDE | * | R7.5 | MSE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

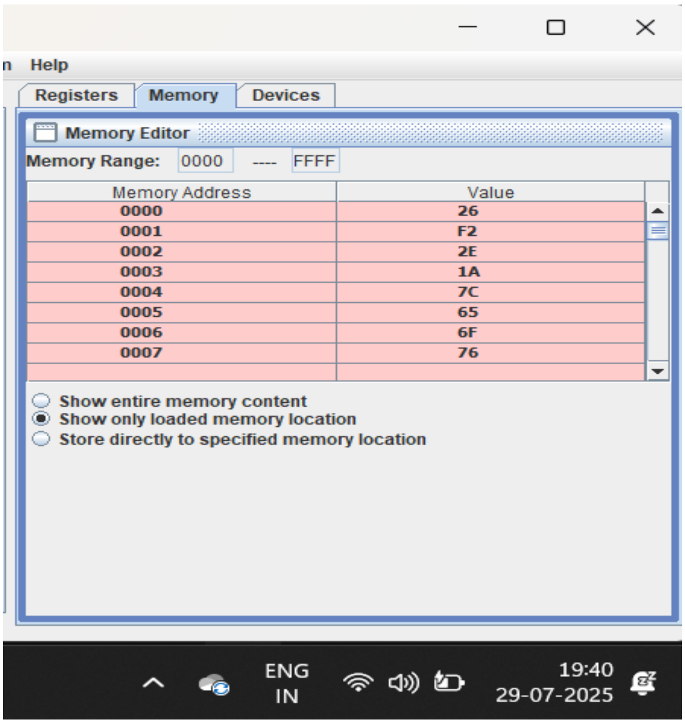
For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

No. Converter Tool :

| Hexadecimal | Decimal | Binary |
|---|---|---|
| 0 | 0 | 0 |

Created by : Jubin Mitra



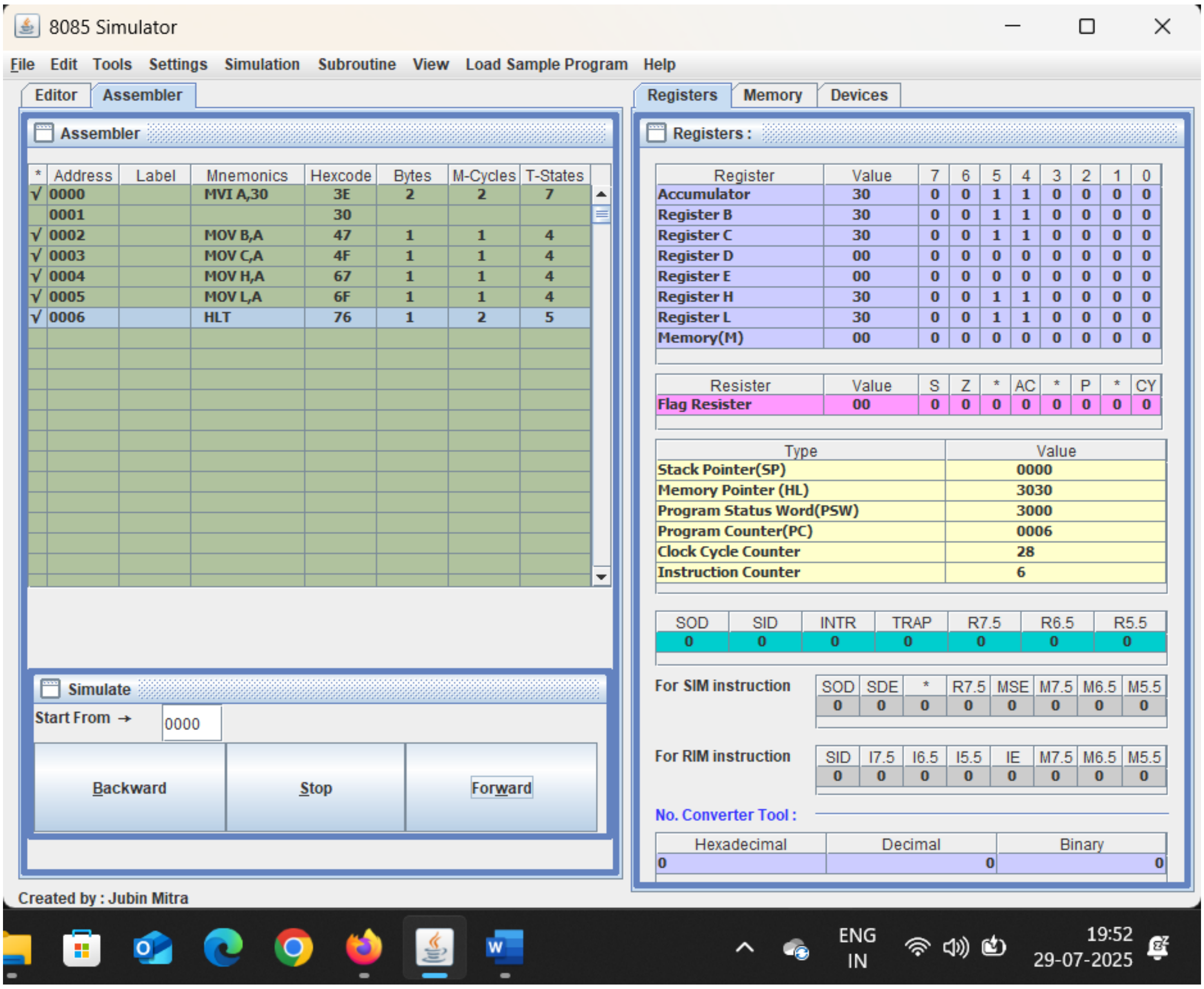
Memory Editor

Memory Range: 0000 ---- FFFF

| Memory Address | Value |
|---|---|
| 0000 | 26 |
| 0001 | F2 |
| 0002 | 2E |
| 0003 | 1A |
| 0004 | 7C |
| 0005 | 65 |
| 0006 | 6F |
| 0007 | 76 |

○ Show entire memory content
● Show only loaded memory location
○ Store directly to specified memory location

[C] Write a program to load an immediate data 30H in register A and copy it into registers B, C, H, L.



| * | Address | Label | Mnemonics | Hexcode | Bytes | M-Cycles | T-States |
|---|---|---|---|---|---|---|---|
| √ | 0000 | | MVI A,30 | 3E | 2 | 2 | 7 |
| | 0001 | | | 30 | | | |
| √ | 0002 | | MOV B,A | 47 | 1 | 1 | 4 |
| √ | 0003 | | MOV C,A | 4F | 1 | 1 | 4 |
| √ | 0004 | | MOV H,A | 67 | 1 | 1 | 4 |
| √ | 0005 | | MOV L,A | 6F | 1 | 1 | 4 |
| √ | 0006 | | HLT | 76 | 1 | 2 | 5 |

Start From → 0000

Created by : Jubin Mitra

| Register | Value | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Accumulator | 30 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Register B | 30 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Register C | 30 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Register D | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register E | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 30 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Register L | 30 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 3030 |
| Program Status Word(PSW) | 3000 |
| Program Counter(PC) | 0006 |
| Clock Cycle Counter | 28 |
| Instruction Counter | 6 |

| SOD | SID | INTR | TRAP | R7.5 | R6.5 | R5.5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For SIM instruction

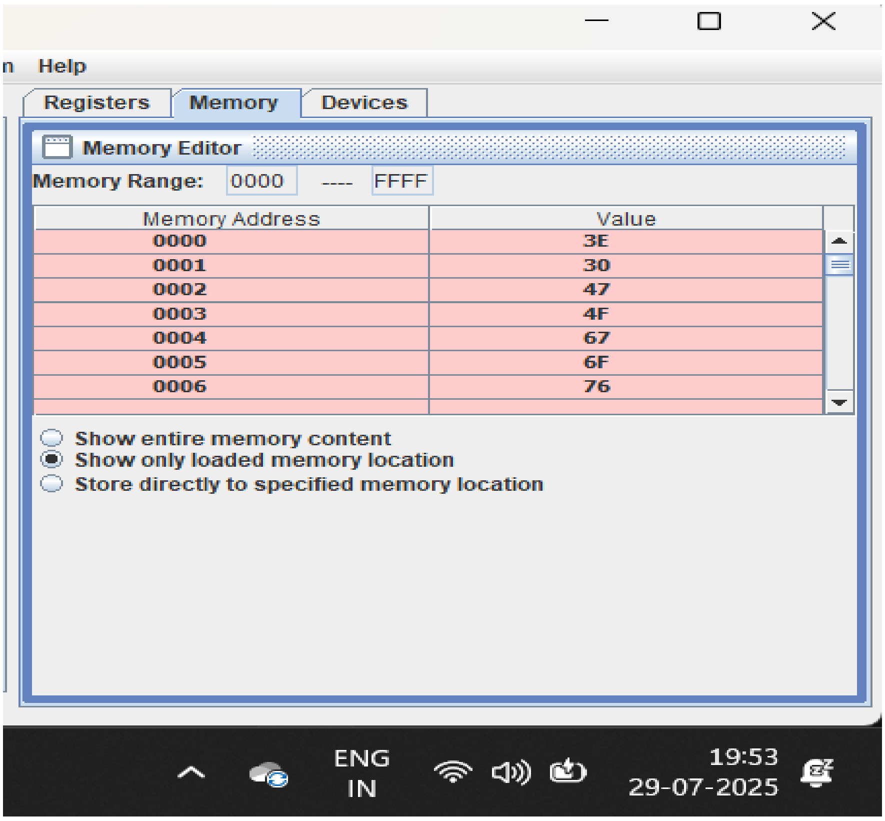
| SOD | SDE | * | R7.5 | MSE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

No. Converter Tool :

| Hexadecimal | Decimal | Binary |
|---|---|---|
| 0 | 0 | 0 |



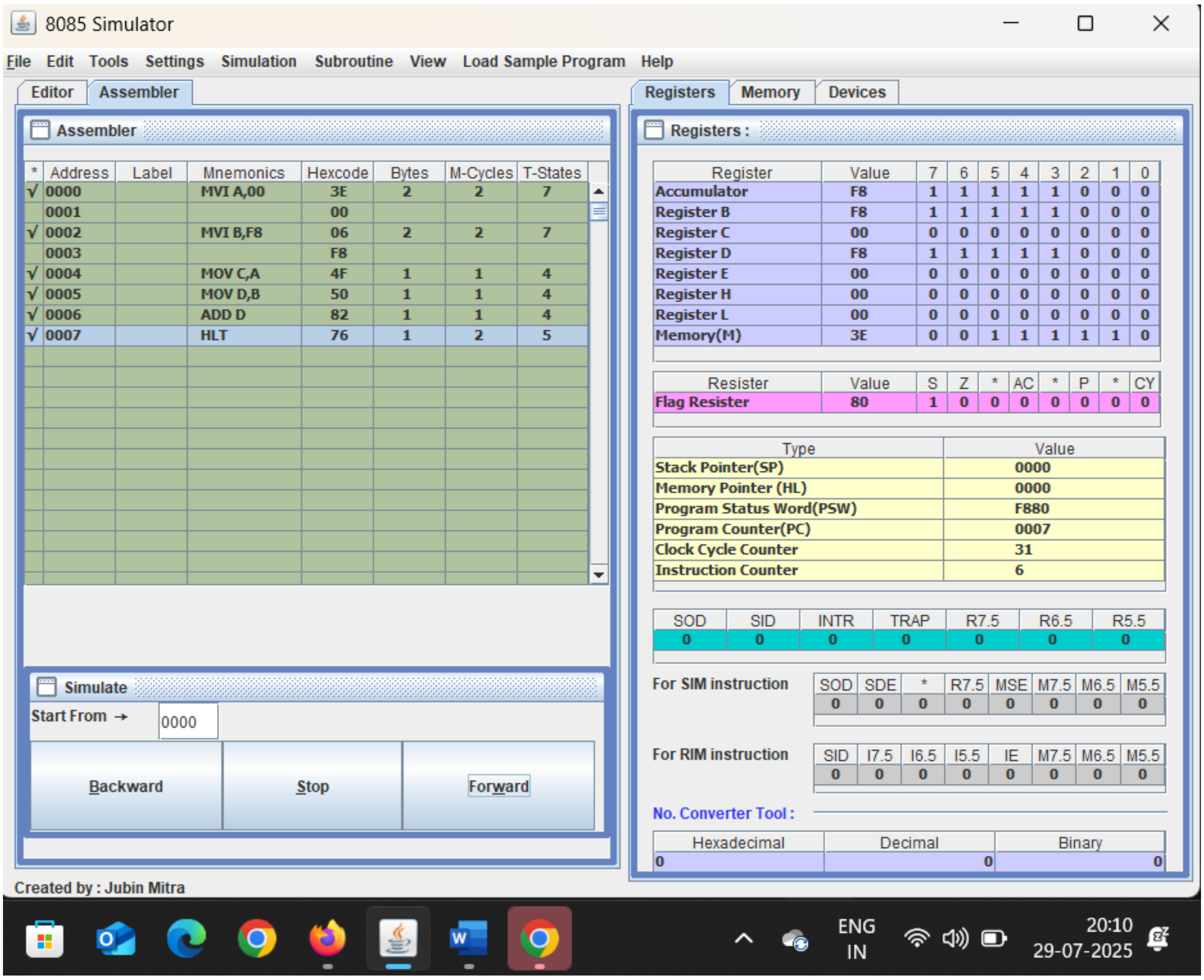
Memory Range: 0000 ---- FFFF

| Memory Address | Value |
|---|---|
| 0000 | 3E |
| 0001 | 30 |
| 0002 | 47 |
| 0003 | 4F |
| 0004 | 67 |
| 0005 | 6F |
| 0006 | 76 |

○ Show entire memory content
● Show only loaded memory location
○ Store directly to specified memory location

2. Specify the register contents and the flag status after each instruction as they are executed.

| | A | B | C | D | S | Z | CY |
|---|---|---|---|---|---|---|---|
| | XX | XX | X | X | X | X | X |
| MVI A,00H | 00 | XX | X | X | X | X | X |
| MVI B,F8H | 00 | F8 | X | X | X | X | X |
| MOV C,A | 00 | F8 | 00 | X | X | X | X |
| MOV D,B | 00 | F8 | 00 | F8 | X | X | X |
| ADD D | F8 | F8 | 00 | F8 | 1 | 0 | 0 |
| HLT | F8 | F8 | 00 | F8 | 1 | 0 | 0 |



| * | Address | Label | Mnemonics | Hexcode | Bytes | M-Cycles | T-States |
|---|---|---|---|---|---|---|---|
| √ | 0000 | | MVI A,00 | 3E | 2 | 2 | 7 |
| | 0001 | | | 00 | | | |
| √ | 0002 | | MVI B,F8 | 06 | 2 | 2 | 7 |
| | 0003 | | | F8 | | | |
| √ | 0004 | | MOV C,A | 4F | 1 | 1 | 4 |
| √ | 0005 | | MOV D,B | 50 | 1 | 1 | 4 |
| √ | 0006 | | ADD D | 82 | 1 | 1 | 4 |
| √ | 0007 | | HLT | 76 | 1 | 2 | 5 |

Start From → 0000

| Register | Value | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Accumulator | F8 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Register B | F8 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Register C | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register D | F8 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Register E | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 3E | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 80 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | F880 |
| Program Counter(PC) | 0007 |
| Clock Cycle Counter | 31 |
| Instruction Counter | 6 |

| SOD | SID | INTR | TRAP | R7.5 | R6.5 | R5.5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For SIM instruction

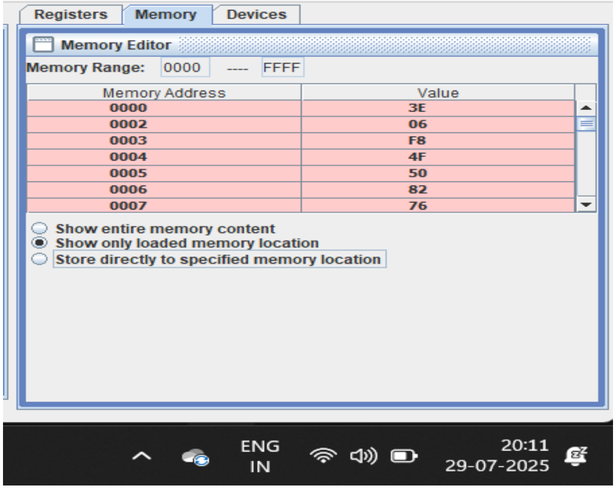
| SOD | SDE | * | R7.5 | MSE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

No. Converter Tool :

| Hexadecimal | Decimal | Binary |
|---|---|---|
| 0 | 0 | 0 |

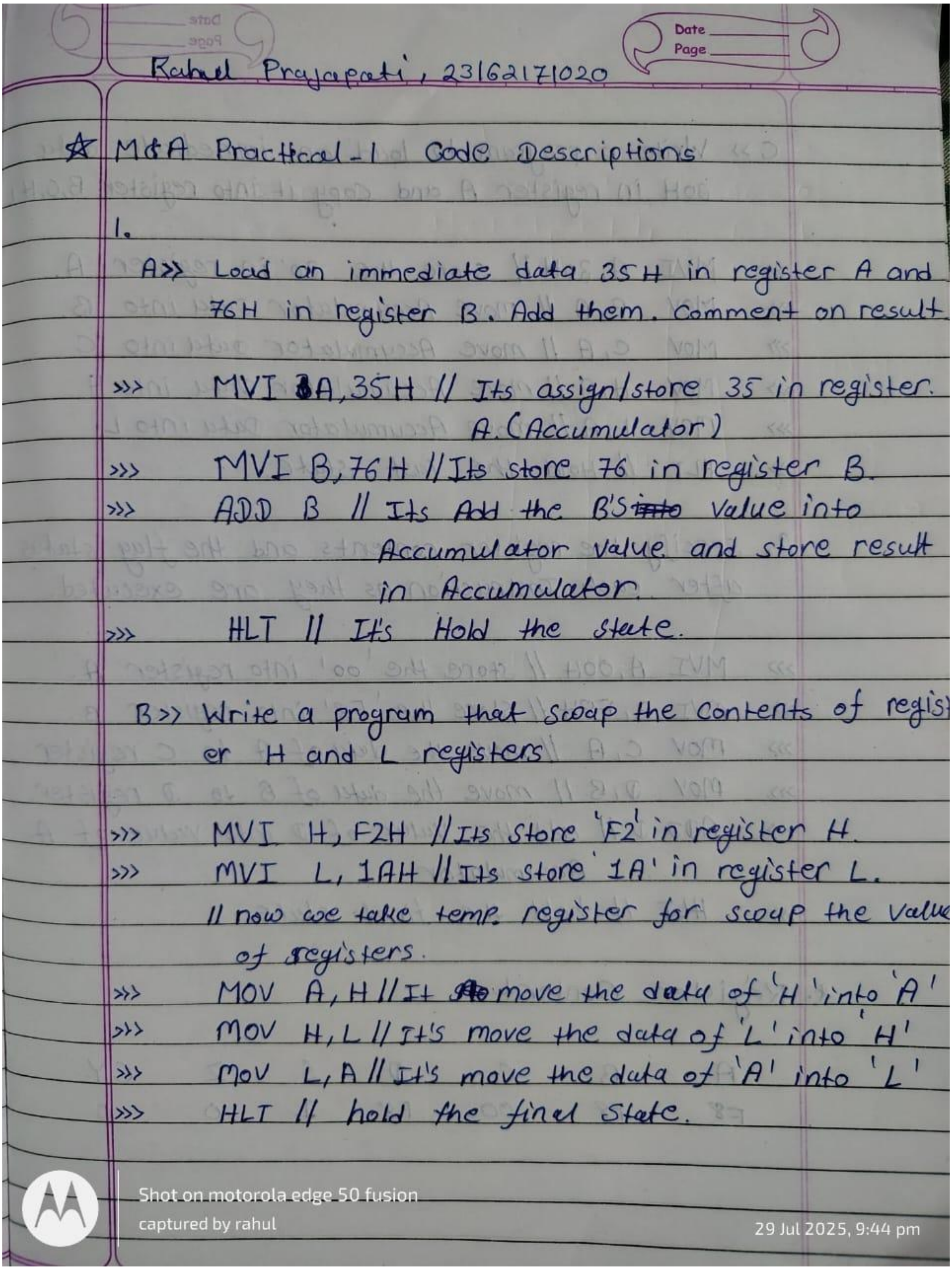


Memory Range: 0000 ---- FFFF

| Memory Address | Value |
|---|---|
| 0000 | 3E |
| 0002 | 06 |
| 0003 | F8 |
| 0004 | 4F |
| 0005 | 50 |
| 0006 | 82 |
| 0007 | 76 |

○ Show entire memory content
● Show only loaded memory location
○ Store directly to specified memory location

➢ HANDWRITTEN CODE DESCRIPTIONS:-



Date
Page

Rahul Prajapati, 23162171020

☆ M&A Practical-1 Code Descriptions

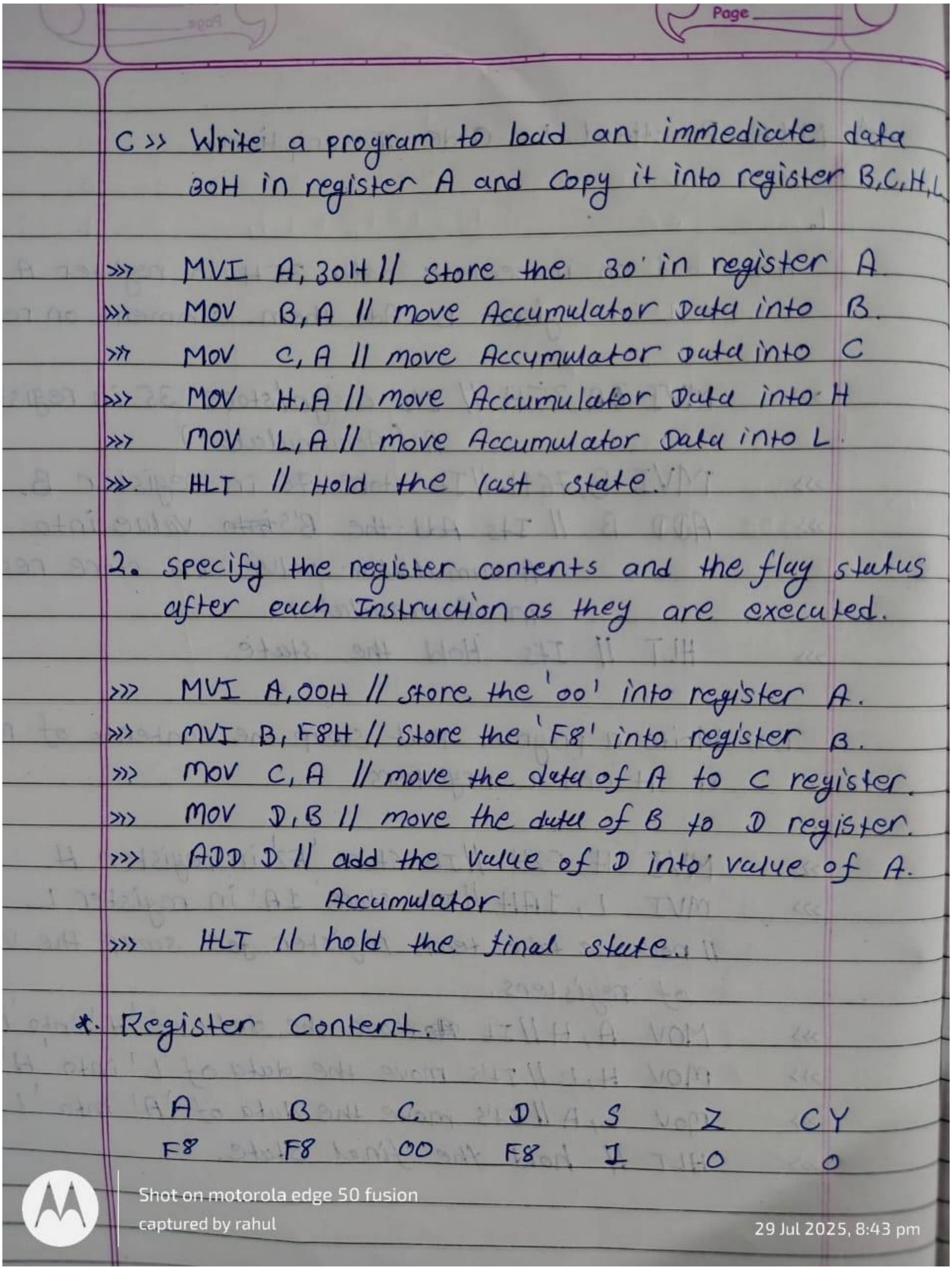
1.

A>> Load an immediate data 35H in register A and 76H in register B. Add them. Comment on result.

```
>>> MVI A,35H // Its assign/store 35 in register.
                    A.(Accumulator)
>>> MVI B,76H //Its store 76 in register B.
>>> ADD B // Its Add the B'S ~~into~~ value into
                 Accumulator value and store result
                 in Accumulator.
>>> HLT // It's Hold the state.
```

B>> Write a program that scoap the contents of regis
er H and L registers.

```
>>> MVI H, F2H //Its Store 'F2' in register H.
>>> MVI L, 1AH //Its store '1A' in register L.
// now we take temp. register for scoup the value
   of registers.
>>> MOV A, H //It move the data of 'H' into 'A'
>>> MOV H,L // It's move the data of 'L' into 'H'
>>> MoV L,A // It's move the data of 'A' into 'L'
>>> HLT // hold the final State.
```



C >> Write a program to load an immediate data 30H in register A and Copy it into register B,C,H,L

```
>>>  MVI A; 30H // store the 30' in register A.
>>>  MOV B, A // move Accumulator Data into B.
>>>  MOV C, A // move Accumulator Data into C
>>>  MOV H, A // move Accumulator Data into H
>>>  MOV L, A // move Accumulator Data into L.
>>>  HLT // Hold the last state.
```

2. specify the register contents and the flag status after each Instruction as they are executed.

```
>>>  MVI A, 00H // store the '00' into register A.
>>>  MVI B, F8H // Store the 'F8' into register B.
>>>  MOV C, A // move the data of A to C register.
>>>  MOV D, B // move the data of B to D register.
>>>  ADD D // add the value of D into value of A.
                   Accumulator
>>>  HLT // hold the final state.
```

*. Register Content.

| A | B | C | D | S | Z | CY |
|---|---|---|---|---|---|---|
| F8 | F8 | 00 | F8 | 1 | 0 | 0 |