TOPIC:
Familiarization with 8085 instruction

OBJECTIVES

i) To demonstrate the basic understanding of assembly language programming using the 8085 microprocessor simulator.

ii) To become familiar with the architecture and instruction set of 8085 microprocessor.

iii) To perform data exchange, arithmetic and logical expression

iv) To perform data exchange between two memory locations and data movement within memory.

INTRODUCTION

8085 is a powerful tool to analyze and optimize the performance, power and reliability of microprocessor designs. Arithmetic group instructions in 8085 play a crucial role in mathematical operation whereas logical instructions are used to manipulate and process data stored in memory or registers. In this week's lab we are going to learn to move data between two address. Additionally we are also going to learn about how division and multiplication are performed.

Program:

1) WAP to move 10 byte of data from starting address 9500H to 9600H.

⇒
```
MVI B, 0AH
LXI H, 9500H
LXI D, 9600H
J1 : MOV A, M
STAX D
INX H
INX D
DCR B
JNZ J1
HLT
```

Here in the above program. MVI loads 10 register B LXI H, 9500H and LXI P, 9600H loads immediate data to register pair. Then loop J1, is used. MOV A, M is used to transfer data of memory to accumulator. STAX D stores the data by specified memory location INX increases register pair by 1. DCR B decreases register pair by 1. JNZ is used to jump on if no 0 is represented by carry flag.

Output

Before execution

	0	1	2	3	4	5	6	7	8	9
950	10	11	12	13	14	15	16	17	18	19

After execution

	0	1	2	3	4	5	6	7	8	9
960	10	11	12	13	14	15	16	17	18	19

2) Write a program to multiply two numbers

→
```
MVI A, 00H
MVI C, 06H
MVI B, 03H
J1: ADDC
DCR B
JNZ J1
STA 2200H
HLT
```

Here; MVI A, 00H (i.e 06H, 03H) is used to load the second byte (8 bit immediate data) into the register specified. The loop J1 is used. ADD is used to add the contents of Register to contents of accumulator and store the result in accumulator. DCR B decrease the contents or register by 1. JNZ is used to jump on if no zero. Here, zero is represented by carry flag. At last STA is used to store the contents of accumulator to specified address.

Output

- Before execution

	0	1	2
220	00	00	00

- After execution

	0	1	2
220	12	00	00

3) WAP to divide two numbers

⇒
```
MVI A, 0CH
MVI B, 02H
MVI C, 00H
J1 : CMP B
  JC J3
  SUB B
  INR C
 JMP J1
J3 : MOV A, C
STA 3300H
  HLT
```

Here; MVI instruction is used to load 8 bit data to register pair. Then J1 loop is used where CMP is used which generates the complement. JC is used to jump on carry (if (CY = 1)). SUB B subtracts the content of register with contents of accumulator INR increases the register pair by 1. JMP loads the program counter by 16 bit address and jumps to specified memory location. MOV instruction copy contents of specified register to memory. STA is used to store contents of accumulator to specified address

- Output

→ Before execution

	0	1	2
330	00	00	00

→ After execution

	0	1	2
330	06	00	00

CONCLUSIONS

Through these lab exercises, we were able to solve problems involving data transfer instructions we were able to learn how to exchange the contents of specified memory location. We learnt about various instruction and flags

These instructions facilitate the movement and manipulation of data between registers and memory locations. In conclusion. This week's lab helped us to understand the concepts that are required to multiply and divide two numbers.