FACULTY OF ENGINEERING & TECHNOLOGY

PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY

BACHELOR OF TECHNOLOGY

COMPUTER ORGANIZATION AND MICROPROCESSOR ARCHITECTURE (303105211)

LABORATORY MANUAL

INDEX

Sr. No.	Aim	Start Date	Performance Date	Sign	Marks
1	**PART A:** ADDITION OF TWO 8 BIT NUMBERS USING 8085. **PART B:** WRITE A PROGRAM TO ADD TWO 16-BIT NUMBERS STORED IN REGISTERS OR MEMORY LOCATIONS. **PART C:** 8 BIT SUBTRACTION				
2	**PART A:** WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM MULTIPLICATION OF TWO 8 BIT NOS. **PART B:** WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM DIVISION OF TWO 8 BIT NOS.				
3	WRITE A PROGRAM TO ADD BLOCK OF 8-BIT DATA STORED IN MEMORY LOCATIONS.				
4	**PART A:** WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MINIMUM FROM TWO 8-BIT NUMBERS. **PART B:** WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MINIMUM FROM BLOCK OF N 8-BIT NUMBERS.				
5	**PART A:** WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MAXIMUM FROM TWO 8-BIT NUMBERS. **PART B:** WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MAXIMUM FROM BLOCK OF N 8-BIT NUMBERS.				
6	**PART A:** WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN ASCENDING ORDER. **PART B:** WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN DECENDING ORDER.				
7	**PART A:** WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BCD NUMBER INTO ITS EQUIVALENT BINARY NUMBER. **PART B:** WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT BCD NUMBER.				
8	**PART A:**WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT ASCII NUMBER. **PART B:** WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN ASCII NUMBER INTO ITS EQUIVALENT BINARY NUMBER.				
9	WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF A SERIES OF EVEN NUMBERS.				
10	WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF SERIES OF ODD NUMBERS				

EXPERIMENT NO.1

AIM: TO PERFORM

PART A: ADDITION OF TWO 8 BIT NUMBERS USING 8085.

ALGORITHM:

1. Start the program by loading the first data into Accumulator.
2. Move the data to a register (B register).
3. Get the second data and load into Accumulator.
4. Add the two register contents.
5. Check for carry.
6. Store the value of sum and carry in memory location.
7. Terminate the program.

PROGRAM:

```
        MVI C,00H
        LDA 0020H
        MOV B,A
        LDA 0021H
        ADD B
        JNC LOOP
        INR C
LOOP:   STA 2152H
        MOV A,C
        STA 0022H
        HLT
```

OBSERVATION:

1 **MVI:** When we want to load the data to any register then we use MVI.

2 **LDA**: LDA command is use to load to data into the accumulator from the given memory location.

3 **MOV:** it means the data which is stored in the source register will be copied into destination register.

4 **ADD B:** ADD is the numeric for addition. The source register b whose value will be added to accumulator. The content of register B are added to the content of accumulator.

5 **JNC:** It is a conditional jump instructor that checks the carry flag in the flag register. The JNC instruction directs the programs to jump to the specified label if the carry flag is zero. If the carry flag is one the program continues with the next instruction in sequence.

6 **STA:** It is used to store the data of the accumulator into a specified memory location.

7 **HLT:** it is used to end the program.

PART B: WRITE A PROGRAM TO ADD TWO 16-BIT NUMBERS STORED IN REGISTERS OR MEMORY LOCATIONS.

```
        MVI C,00H
        LDA 2052H
        MOV B,A
        LDA 2054H
        ADD B
        STA 2055H
        LDA 2051H
        MOV D,A
        LDA 2053H
        ADC D
        JNC LOOP
        INR C

LOOP:   STA 2056H
        MOV A,C
        STA 2057H
        HLT
```

OBSERVATION:

Input: 2051H: 81H (First number's upper byte)
2052H: 80H (First number's lower byte)
2053H: 81H (Second number's upper byte)
2054H: 20H (Second number's lower byte)

Output: 2055H: A0H (Lower byte of result)
2056H: 02H (Upper byte of result)
2057H: 01H (Carry)

PART C: 8 BIT SUBTRACTION

```
        MVI C,00H
        LXI H,4200H
        MOV A,M
        INX H
        MOV B,M
        SUB B
        JNC LOOP
        INR C
        CMA
        INR A
LOOP:   STA 4202H
        MOV A,C
        STA 4203H
        HLT
```

CONCLUSION: