FACULTY OF ENGINEERING & TECHNOLOGY

PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY

BACHELOR OF TECHNOLOGY

COMPUTER ORGANIZATION AND MICROPROCESSOR ARCHITECTURE (303105211)

LABORATORY MANUAL

INDEX

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 2150H
        MOV B,A
        LDA 2151H
        ADD B
        JNC LOOP
        INR C
LOOP:   STA 2152H
        MOV A,C
        STA 2153H
        HLT
```

OBSERVATION:

```
        Input:  2050H: 88H
                2051H: 82H
        Output: 2052H: 0AH
                 2053H: 01H
```

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:

EXPERIMENT NO: 2

AIM:

PART A: WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM MULTIPLICATION OF TWO 8 BIT NOS.

PROGRAM:

```
         MVI D,00H
         MVI A,00H
         LXI H,4150H
         MOV B,M
         INX H
         MOV C,M
LOOP:    ADD B
         JNC NEXT
         INR D
NEXT:    DCR C
         JNZ LOOP
         STA 4152H
         MOV A,D
         STA 4153H
         HLT
```

OBSERVATION:

INPUT:	FF	(4150)
	FF	(4151)
OUTPUT:	01	(4152)
	FE	(4153)

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM DIVISION OF TWO 8 BIT NOS.

PROGRAM:

```
            LXI H,4150H
            MOV B,M
            MVI C,00H
            INX H
            MOV A,M
NEXT:       CMP B
            JC LOOP
            SUB B
            INR C
            JMP NEXT
LOOP:       STA 4152H
            MOV A,C
            STA 4153H
            HLT
```

OBSERVATION:

```
INPUT:      4150H-FF
            4151H-FF
OUTPUT:     4152H-01    -----REMAINDER
            4153H-FE   -----QUOTIENT
```

CONCLUSION:

EXPERIMENT NO: 3

AIM: WRITE A PROGRAM TO ADD BLOCK OF 8-BIT DATA STORED IN MEMORY LOCATIONS.

PROGRAM:

```
        MVI C,00H
        MVI D,05H
        MVI A,00H
        LXI H,2050H

AGAIN:  ADD M
        JNC NEXT
        INR C

NEXT:   INX H
        DCR D
        JNZ AGAIN

LOOP:   STA 2055H
        MOV A,C
        STA 2056H
        HLT
```

OBSERVATION:

```
Input:  2050H: 30H (First data)
        2051H: 10H (Second data)
        2052H: 10H (Third data)
        2053H: 10H (Fourth data)
        2054H: 05H (Fifth data)
Output: 2055H: 55H (Result of addition)
        2056H: 00H (Carry, if generated)
```

CONCLUSION:

EXPERIMENT NO: 4

PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MINIMUM FROM TWO 8-BIT NUMBERS.

PROGRAM:

```
        MVI A,00H
        LXI H,4201H
        MOV A,M

        INX H
        CMP M
        JC AHEAD
        MOV A,M

AHEAD:  STA 4203H
        HLT
```

OBSERVATION:

```
    Input:  4201H: 09H
            4202H: 30H
    Output: 4203H: 09H
```

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MINIMUM FROM BLOCK OF N 8-BIT NUMBERS.

ALGORITHM:

1. Load the address of the first element of the array in HL pair.
2. Move the count to B register.
3. Increment the pointer.
4. Get the first data in Accumulator.
5. Decrement the counter.
6. Increment the pointer.
7. Compare the content of memory addressed by HL pair with that of Accumulator.
8. If carry=1, go to step 10 or if carry=0, go to step 9.
9. Move the content of memory addressed by HL to Accumulator.
10. Decrement the count.
11. Check for zero of the count. If ZF=0, go to step 6, or if ZF=1, go to next step.

12. Store the smallest data in memory.
13. Terminate the program.

PROGRAM:

```
        MVI C,04H
        MVI A,00H
        LXI H,4201H
        MOV A,M

LOOP:   INX H
        CMP M
        JC AHEAD
        MOV A,M
        DCR C
        JNZ LOOP
        STA 420AH
        JMP LAST

AHEAD: DCR C
        JNZ LOOP
        STA 420AH

LAST:   HLT
```

OBSERVATION:

```
    Input:  4201H: 20H
            4202H: 05H
            4203H: 09H
            4204H: 30H
            4205H: 23H
    Output: 430AH: 05H
```

CONCLUSION:

EXPERIMENT NO: 5

PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MAXIMUM FROM TWO 8-BIT NUMBERS.

PROGRAM:

```
        MVI A,00H
        LXI H,4201H
        MOV A,M

LOOP:   INX H
        CMP M
        JNC AHEAD
        MOV A,M

AHEAD: STA 4203H
        HLT
```

OBSERVATION:

```
    Input:  4201H: 09H
            4202H: 30H
    Output: 4203H: 30H
```

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MAXIMUM FROM BLOCK OF N 8-BIT NUMBERS.

ALGORITHM:

1. Load the address of the first element of the array in HL pair.
2. Move the count to B register.
3. Increment the pointer.
4. Get the first data in Accumulator.
5. Decrement the counter.
6. Increment the pointer.
7. Compare the content of memory addressed by HL pair with that of Accumulator.
8. If carry=0, go to step 10 or if carry=1, go to step 9.
9. Move the content of memory addressed by HL to Accumulator.
10. Decrement the count.
11. Check for zero of the count. If ZF=0, go to step 6, or if ZF=1, go to next step.
12. Store the largest data in memory.

13. Terminate the program.

PROGRAM:

```
        MVI C,04H
        MVI A,00H
        LXI H,4201H
        MOV A,M

LOOP:   INX H
        CMP M
        JNC AHEAD
        MOV A,M
        DCR C
        JNZ LOOP
        STA 4206H
        JMP LAST

AHEAD:          DCR C
        JNZ LOOP
        STA 420AH

LAST:   HLT
```

OBSERVATION:

```
    Input:  4201H: 20H
            4202H: 05H
            4203H: 09H
            4204H: 30H
            4205H: 23H
    Output: 420AH: 30H
```

CONCLUSION:

EXPERIMENT NO: 6

AIM: PART A: WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN ASCENDING ORDER.

ALGORITHM:

1. Initialize HL pair as memory pointer.
2. Get the count at 4200 into C register.
3. Copy it in D register (For bubble sort (N-1) times required).
4. Get the first value in Accumulator.
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of Accumulator and memory.
7. Decrement content of D register by 1.
8. Repeat steps 5 and 7 till the value in register D becomes zero.
9. Decrement content of register C by 1.
10. Repeat steps 3 to 9 till the value in register C becomes zero.

PROGRAM:

```
        LXI H,4201H
        MVI C,05H
REPEAT: MOV D,C
        LXI H, 4201H
LOOP:   MOV A,M
        INX H
        CMP M
        JC SKIP
        MOV B,M
        MOV M,A
        DCX H
        MOV M,B
        INX H
SKIP:   DCR D
        JNZ LOOP
        DCR C
        JNZ REPEAT
        HLT
```

OBSERVATION:

Input: 4201H: 10H

	4202H:	25H
	4203H:	12H
	4204H:	22H
	4205H:	05H
Output:	4201H:	05H
	4202H:	10H
	4203H:	12H
	4204H:	22H
	4205H:	25H

PART B: WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN DECENDING ORDER.

ALGORITHM:

1. Initialize HL pair as memory pointer.
2. Get the count at 4200 into C register.
3. Copy it in D register (For bubble sort (N-1) times required).
4. Get the first value in Accumulator.
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of Accumulator and memory.
7. Decrement content of register D by 1.
8. Repeat steps 5 and 7 till the value in register D becomes zero.
9. Decrement register C by 1.
10. Repeat steps 3 to 9 till the value in register C becomes zero.

PROGRAM:

```
        LXI H,4201H
        MVI C,05H

REPEAT: MOV D,C
        LXI H,4201

LOOP:   MOV A,M
        INX H
        CMP M
        JNC SKIP
        MOV B,M
        MOV M,A
        DCX H
        MOV M,B
        INX H

SKIP:   DCR D
        JNZ LOOP
        DCR C
        JNZ REPEAT
        HLT
```

OBSERVATION :

Input:	4201H:	10H
	4202H:	25H
	4203H:	12H
	4204H:	22H
	4205H:	05H
Output:	4201H:	25H
	4202H:	22H
	4203H:	12H
	4204H:	10H
	4205H:	05H

CONCLUSION:

EXPERIMENT NO: 7

AIM: PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BCD NUMBER INTO ITS EQUIVALENT BINARY NUMBER.

PROGRAM:

```
START:  LXI SP,FFFFH
        LXI H,2001H
        LXI B,2101H
        MOV A,M
        CALL BCDBIN
        STAX B
        HLT

BCDBIN: PUSH B
        PUSH D
        MOV B,A
        ANI 0F
        MOV C,A
        MOV A,B
        ANI F0
        JZ BCD1
        RRC
        RRC
        RRC
        RRC
        MOV D,A
        XRA A
        MVI E,0AH

SUM:    ADD E
        DCR D
        JNZ SUM

BCD1:   ADD C
        POP D
        POP B
        RET
```

OBSERVATION:

Input:	2001H:	72H
Output:	2101H:	48H

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT BCD NUMBER.

```
START: LXI SP, FFFFH
            LXI H,2001H
            MOV A, M
            CALL PWRTEN
            HLT

PWRTEN: LXI H,2011
            MVI B,64H
            CALL BINBCD
            MVI B,0AH
            CALL BINBCD
            MOV M,A
            RET

BINBCD: MVI M, FFH

NXTBUF: INR M
            SUB B
            JNC NXTBUF
            ADD B
            INX H
            RET
```

OBSERVATION:

Input:	2001H:	FFH
Output:	2011H:	02H
	2012H:	05H
	2013H:	05H

CONCLUSION:

EXPERIMENT NO: 8

AIM: PART A:WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT ASCII NUMBER.

PROGRAM:

```
         LXI SP,FFFFH
         LXI H,2250H
         LXI D,2260H
         MOV A,M
         MOV B,A
         RRC
         RRC
         RRC
         RRC
         CALL ASCII
         STAX D
         INX D
         MOV A,B
         CALL ASCII
         STAX D
         HLT

ASCII:   ANI 0FH
         CPI 0AH
         JC CODE
         ADI 07H

CODE:    ADI 30H
         RET
```

OBSERVATION:

Input:	2250H:	7EH
Output:	2260H:	37H
	2261H:	45H

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN ASCII NUMBER INTO ITS EQUIVALENT BINARY NUMBER.

PROGRAM:

```
            LXI SP,FFFFH
            LXI H,2250H
            LXI D,2260H
            MOV A,M
// TAKING ASCII NUMBER WHOSE ORIGINAL BINARY IS LESS THAN 10
            CALL ASCIIBIN
            STAX D
            INX H
            MOV A,M
// TAKING ASCII NUMBER WHOSE ORIGINAL BINARY IS LESS THAN 10
            CALL ASCIIBIN
            INX D
            STAX D
            HLT

ASCIIBIN: SUI 30H
            CPI 0AH
            RC
            SUI 07H
            RET
```

OBSERVATION:

Input:	2250H:	36H (ASCII Data 1)
	2251H:	45H (ASCII Data 2)
Output:	2260H:	06H (BINARY ANSWER FOR ASCII DATA 1)
	2261H:	0EH (BINARY ANSWER FOR ASCII DATA 2)

CONCLUSION:

EXPERIMENT NO: 9

AIM: WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF A SERIES OF EVEN NUMBERS.

PROGRAM:

```
LDA 2500H
MOV C, A: "Initialize counter"
MVI B, 00H: "sum = 0"
LXI H, 2501H: “initialize pointer"
BACK: MOV A, M: "Get the number"
ANI 01H: "Mask Bit l to Bit7"
JNZ SKIP: "Don't add if the number is ODD"
 MOV A, B: "Get the sum"
ADD M: "SUM = SUM + data"
MOV B, A: "Store result in B register"
SKIP: INX H: "increment pointer"
DCR C: "Decrement counter"
JNZ BACK: "if counter 0 repeat"
STA 2505H: "store sum"
HLT: "Stop"
```

OBSERVATION:

INPUT: 2500 H = 4H
2501 H = 20H
2502 H = 15H
2503 H = 13H
2504 H = 22H

OUTPUT: Result: 2505 H = 20+22= 42H

CONCLUSION:

EXPERIMENT NO: 10

AIM: WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF SERIES OF ODD NUMBERS.

PROGRAM:

```
LDA 2500H
MOV C, A: "Initialize counter"
LXI H, 2501H: "Initialize pointer"
MVI E, 00H: "Sum low = 0"
MOV D, E: "Sum high = 0"
BACK: MOV A, M: "Get the number"
ANI 01H: "Mask Bit 1 to Bit-7"
JZ SKIP: "Don't add if the number is even"
MOV A, E: "Get the lower byte of sum"
ADD M: "Sum = sum + data"
MOV E, A: "Store result in E register"
JNC SKIP
INR D: "Add carry to MSB of SUM"
SKIP: INX H: "Increment pointer"
HLT
```

OBSERVATION:

Inputs: 2500 H = 4H
2501 H = 9AH
2502 H = 52H
2503 H = 89H
2504 H = 3FH

OUTPUT: Result = 2505 H = 89H + 3FH= C8H

CONCLUSION: