



**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
SONEPAT**

Microprocessor and Interfacing Lab
(CSC508)

Practical Lab File

Submitted To:

Dr. Rajiv Verma

Submitted By

Dipankar Yadav

Roll No.- 12111070

Branch: CSE

Semester: 5

Session: 2021-25

| Practical o. | Topic | Page Number |
|-------------------------|---|------------------------|
| 1. | Addition of two 8 bit numbers | 5 |
| 2. | Subtraction of two 8 bit numbers | 6 |
| 3. | Addition with a carry of two 8 bit numbers | 7 |
| 4. | Subtraction with a borrow of two 8 bit numbers | 8 |
| 5. | Multiplication of two 8 bit numbers using repeated addition | 9 |
| 6. | Multiplication of two 8 bit numbers using bit rotation | 10 |
| 7. | Division of two 8 bit numbers using repeated addition | 11 |
| 8. | Division of two 8 bit numbers using bit rotation | 12 |

| | | |
|-----|--|-------|
| 9. | Addition of two BCD without using DAD | 13-14 |
| 10. | Maximum number in an array | 15 |
| 11 | Square of a 8 bit number | 16 |

Practical 1: - Write a program for addition of two 8 bit numbers

CODE:

MVI A,01H

MVI B,05H

ADD B

STA 0000H

HLT

OUTPUT:

Memory view

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | 06 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Practical 2: - Write a program for subtraction of two 8 bit numbers

CODE:

MVI A,11H

MVI B,05H

SUB B

STA 0000H

HLT

OUTPUT:

main.asm

```
1 MVI A,11H
2 MVI B,05H
3 SUB B
4 STA 0000H
5 HLT
6
7
```

Load at 0x0800

Memory View

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | 0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 006 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Practical 3: - Write a program for addition with a carry of two 8 bit numbers

CODE:

```
MVI A, 00H
MVI B, 23H
MVI C, 98H
MVI D, 45H
MVI E, 22H
MOV A, C
ADD E
MOV C, A
STA 0061H
MOV A, B
ADC D
STA 0060H
hlt
```

OUTPUT:

Load at 0x0800

main.asm

```
1 ;<Program title>
2 jmp start
3 ;data
4 ; code
5 start: nop
6 MVI A, 00H
7 MVI B, 23H
8 MVI C, 98H
9 MVI D, 45H
10 MVI E, 22H
11 MOV A, C
12 ADD E
13 MOV C, A
14 STA 0061H
15 MOV A, B
16 ADC D
17 STA 0060H
18 hlt
```

Memory View

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|----|----|----|----|----|----|----|----|----|----|
| 000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 006 | 68 | BA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Your location: the re

Practical 4: - Write a program for subtraction of two 8 bit numbers with borrow.

CODE:

MVI A, 8DH

MVI B, 7DH

MOV C, A

SUB B

JC SUBTRACT

SUBTRACT: STA RESULT

RESULT: DB 00H

HLT;

OUTPUT:

| Memory View | | | | | | | | | | | | | | | |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Practical 5: - Write a program for multiplication of two 8 bit numbers using repeated addition.

CODE:

```
MVI b,02h
MVI c,04h
MVI a, 00h
MVI d,00h
loop: add b
jnc skip
inc d
skip: dcr c
jnz loop
mov b,d
mov c,a
mov a,c
sta 0000h
hlt
```

OUTPUT:

| Memory view | | 0x Address | | | | | | | | | | | | | | | |
|-------------|----|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 000 | 08 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 006 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Practical 6: - Write a program for multiplication of two 8 bit numbers using bit rotation method

CODE:

```
MVI D,06H
MVI A,05H
LXI H,0000H
```

```
LOOP: RRC
      JNC SKIP
      DAD D
```

```
SKIP: XCHG
      DAD H
      XCHG
      DCR C
      JNZ LOOP
      HLT
```

OUTPUT:

| | |
|--|---|
| A/PSW | 0x 05 56 |
| BC | 0x 00 00 |
| DE | 0x 00 00 |
| HL | 0x 1E 00 |
| SP | 0x FF FF |
| PC | 0x 08 14 |
| Flags  | |
| 7 |  |

| main.asm | |
|----------|-------------|
| 1 | MVI D,06H |
| 2 | MVI A,05H |
| 3 | LXI H,0000H |
| 4 | |
| 5 | LOOP: RRC |
| 6 | JNC SKIP |
| 7 | DAD D |
| 8 | |
| 9 | SKIP: XCHG |
| 10 | DAD H |
| 11 | XCHG |
| 12 | DCR C |
| 13 | JNZ LOOP |
| 14 | HLT |

Practical 7: - Write a program for division of two 8 bit numbers by repeated addition method.

CODE:

```
MVI A, 27H
MVI B, 05H
MVI C, 00H
MVI D, 00H
LOOP: SUB B
      JC DONE
INR C
JMP LOOP
DONE: MOV E, A
HLT
```

OUTPUT:

| | |
|--------------|----------|
| A/PSW | 0x FF 97 |
| BC | 0x 05 07 |
| DE | 0x 00 FF |
| HL | 0x 00 00 |
| SP | 0x FF FF |
| PC | 0x 08 12 |

Flags 

main.asm

```
1 MVI A, 27H
2 MVI B, 05H
3 MVI C, 00H
4 MVI D, 00H
5 LOOP: SUB B
6       JC DONE
7 INR C
8 JMP LOOP
9 DONE: MOV E, A
10 HLT
```

Practical 8: - Write a program for division of two 8 bit numbers using by bit rotation method.

CODE:

```

MVI E,00H;
LHLD 0008H;
LDA 0007H;
MOV B,A;
MVI C,08H;
NEXT:DAD H;
MOV A,E;
RLC
MOV E,A;
MOV A,H;
SUB B;
JC SKIP;
MOV H,A;
INR E;
SKIP:DCR C;
JNZ NEXT;
MOV A,E;
STA 0033H;
MOV A,H;
STA 0034H;
HLT;

```

OUTPUT:

```

main.asm
1 start: nop
2     MVI E,00H;
3     LHLD 0008H;
4     LDA 0007H;
5     MOV B,A;
6     MVI C,08H;
7     NEXT:DAD H;
8     MOV A,E;
9     RLC
10    MOV E,A;
11    MOV A,H;
12    SUB B;
13    JC SKIP;
14

```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A |
|-----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | FF | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 006 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Practical 9: - Write an assembly language program to perform addition of 2 BCD numbers without using DAD in an 8085 microprocessor..

CODE:

```
LXI H,0000H
```

```
MOV B, M  
INX H  
MOV C,M  
MVI A, 00H
```

```
MOV A,B  
ADD C  
MOV B,A
```

```
ANI 0FH  
CPI 09H  
JNC M  
MOV A,B  
JMP END
```

```
M: JNZ ABC  
    MOV A,B  
    JMP END
```

```
ABC: ADI 06H  
    MOV C,A  
    ANI F0H  
    MOV D,A  
    MOV A,C  
    SUB D  
    MOV C,A  
    MOV A,B  
    ANI F0H  
    ADD D  
    CPI 90H  
    JNC M2  
    ADD C  
    JMP END
```

```
M2: JNZ XYZ  
    ADD C  
    JMP END
```

```

XYZ: ADI 60H
      ADD C
      MOV B,A
      MVI A,01H
      STA 0001H
      MOV A,B
      JMP END






```

```

END: STA 0002H
      HLT

```

Output:

Load at 0x0800

main.asm


```


1 LXI H,0000H
2
3 MOV B, M
4 INX H
5 MOV C,M
6 MVI A, 00H
7
8
9 MOV A,B
10 ADD C
11 MOV B,A
12
13 ANI 0FH
14 CPI 09H
15 JNC M
16 MOV A,B
17 JMP END
18
19 M: JNZ ABC
20 MOV A,B

```

Memory View

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|-----|----|----|----|----|----|----|----|
| 000 | 08 | 09 | 11 | 00 | 00 | 00 | 00 |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 006 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00B | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00C | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00D | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00E | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00F | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Start Address at: 0x 



10. Write an assembly language program to find the maximum value in an array in an 8085 microprocessor.

Code:

```
LXI H,0001H
LDA 0000H
MOV D,A
MVI A,0H
LOOP: CMP M
JNC VALUPDT
MOV A,M
VALUPDT: INX H
DCR D
JNZ LOOP
MOV M,A
HLT
```

Output:

Load at 0x0800

main.asm

```
1 LXI H,0001H
2 LDA 0000H
3 MOV D,A
4 MVI A,0H
5 LOOP: CMP M
6 JNC VALUPDT
7 MOV A,M
8 VALUPDT: INX H
9 DCR D
10 JNZ LOOP
11 MOV M,A
12 HLT
```

Memory View

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|----|----|----|----|----|----|----|----|
| 000 | 03 | 0C | FD | 87 | FD | 00 | 00 | 0C |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 006 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 00B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 00C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 00D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 00E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |
| 00F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0C |

Start Address at: 0x

11. Write an assembly language program to find the square of an 8 bit number in an 8085 microprocessor.

Code:

```
LHLD 0000H
XCHG
LDA 0000H
LXI H,0000
MVI C,08H
Loop:DAD H
RAL
JNC Ahead
DAD D
Ahead:DCR C
JNZ Loop
SHLD 0001H
HLT
```

Output:



The screenshot displays an 8085 assembly simulator interface. On the left, a code editor window titled 'main.asm' contains the following assembly code:

```
1 LHLD 0000H
2 XCHG
3 LDA 0000H
4 LXI H,0000
5 MVI C,08H
6 Loop:DAD H
7 RAL
8 JNC Ahead
9 DAD D
10 Ahead:DCR C
11 JNZ Loop
12 SHLD 0001H
13 HLT
14
```

On the right, a 'Memory View' window shows the memory contents. The address 0000 is highlighted, and the value 05 19 is displayed in the first two columns. The rest of the memory is filled with 00.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|----|----|----|----|----|----|----|----|
| 000 | 05 | 19 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 005 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 006 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |