National Institute Of Technology, Srinagar

Microprocessor Lab File

Name: Sameer Hameed
Enroll: 2021BITE058
Semester: 5th

November 1, 2023

INDEX:-

Q.1 Write a 8085 assembly code for adding is 2 8-bit numbers

Q.2 Write a 8085 assembly code for Subtracting 2 8-bit numbers.

Q.3 Write a 8085 assembly code for Multiplying 2 8-bit numbers

Q.4 Write a 8085 assembly code for Dividing two 8-bit numbers?

Q.5 Write a 8085 assembly code for to find the Largest in an array

Q.6 Code to develop a subroutine to add two floating point number.

Q.7 Program to add two double byte numbers.

Q.8 Program to divide 4byte number with the another 4 byte number.

Q.9 Program to divide 8 -bit number with 8-bit number upt to fractional coefficient of 16 bit

Q.10 Program to Adding first N natural number and store result on memory location x.

Q.11 Program to Sort an array using bubble sort;

Q.12 Program to find the factorial of a number;

Q.13 Program to find the 1's and 2's compliment of a giving number;

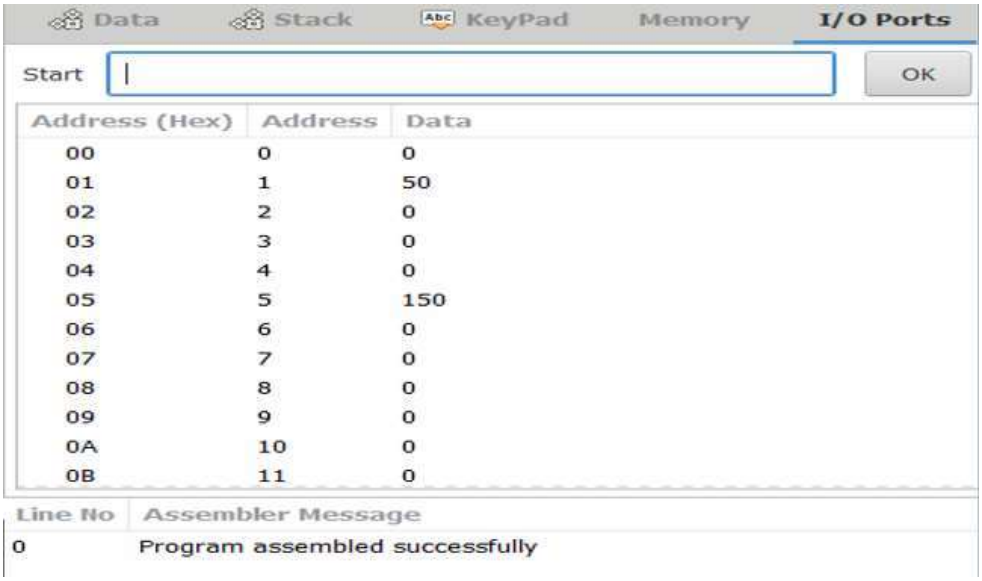
Q.13 Program to find addition of 2 four BYTE number;

Q.14 Program to find addition of 2 four BYTE number;

Q.1 Write a 8085 assembly code for adding is 2 8-bit numbers

```
;there is 2 8-bit numbers.oneis stored in memory location 0026H and the other is
;is stored in input port with address 01H.Add these two number and
;store the result and carry at memory location 0027H and 0028H
;respectively .Also dispaly the result at output port 05H and the carry in port 06H
mvi c,00H
LXI H,0026H
MOV B,H
IN 01H
ADD B
JNC STORE
INR C
STORE: INX H
OUT 05H
MOV A,C
OUT 06H
Hlt
```

Output:



Q.2 Write a 8085 assembly code for Subtracting 2 8-bit numbers.

```
;subtracting two 8-bit no. first no.is stored at mem.add.
;0050H
; and the second no.is stored in mem.address 0051H.
;stored the final result at the memory add 0053H

LXI H,0050H
MOV A,M;
INX H;
MOV B,M;
SUB B;
STA 0053H;
HLT;
```

Data | Stack | KeyPad | **Memory** | I/O Ports

Start | OK

Address (Hex)	Address	Data
004C	76	0
004D	77	0
004E	78	0
004F	79	0
0050	80	33
0051	81	15
0052	82	0
0053	83	18
0054	84	0
0055	85	0
0056	86	0
0057	87	0
0058	88	0

Line No	Assembler Message
0	Program assembled successfully

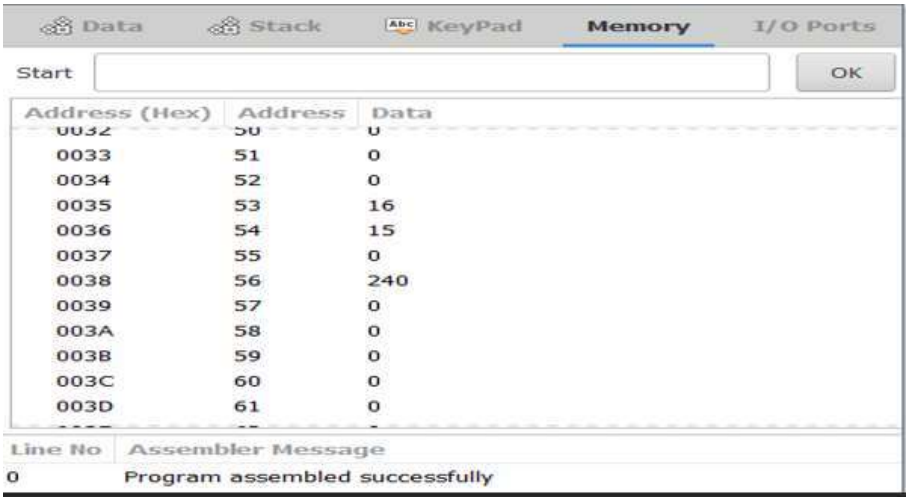
Q.3 Write a 8085 assembly code for Multiplying 2 8-bit numbers

;perform the multiplication of two 8-bit numbers stored at memory locations 0035H and 0036H repectively.

;store the result at memory location at 0037H and 0038H.

```
LXI H,0035H

MOV B,M

INX H

XRA A

MOV C,A

LOOP: ADD M

JNC SKIP

INR C

SKIP: DCR B

JNZ LOOP

INX H

MOV M,C

INX H

MOV M,A

Hlt
```

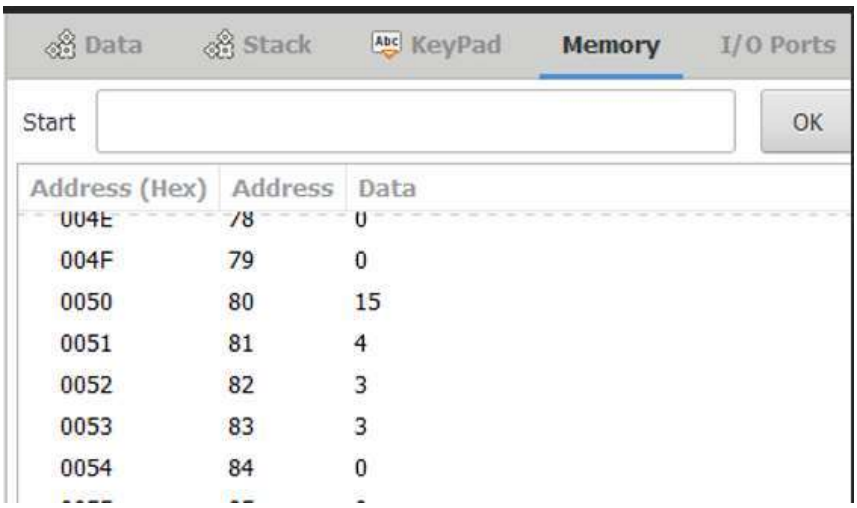
Output:-



Q.4 Write a 8085 assembly code for Dividing two 8-bit numbers?

```
;first no.is stored in 0050H -DIVIDENT ,second no. is stored in 0051H -DIVISOR
;we have to stored the remainder in 0052H and stored the quotient in 0053H
LXI H,0050H;
MOV A,M;
INX H;
MOV B,M;
MVI C,00H;
LOOP: CMP B;
JC SKIP;
SUB B;
INR C;
JMP LOOP;
SKIP: STA 0052H;
MOV A,C;
STA 0053H;
HLT;
```

Output:-

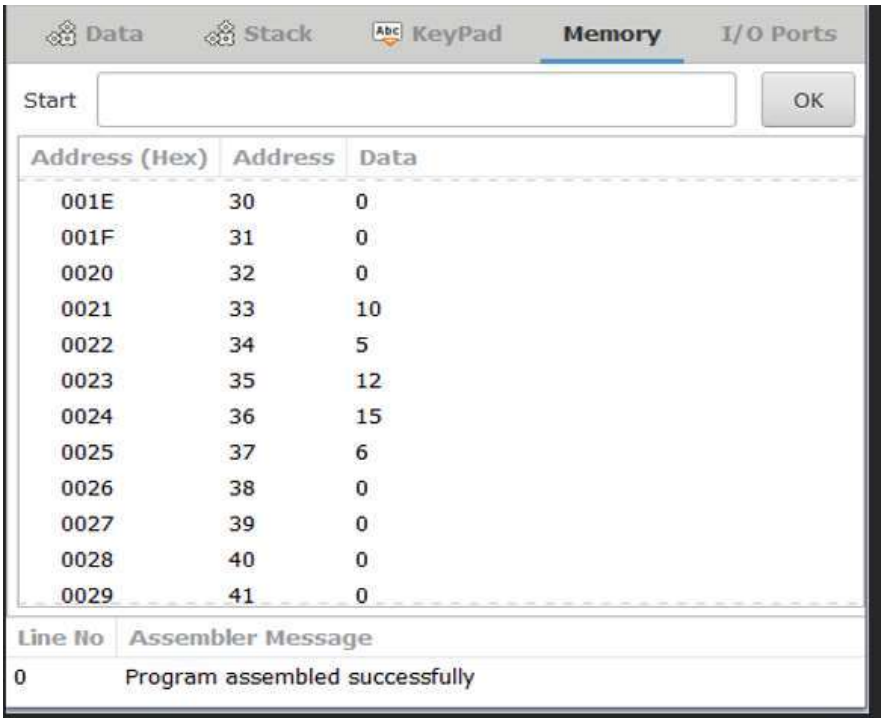


Q.5 Write a 8085 assembly code for to find the Largest in an array

```
;Largest element in an array stored in memory
;at 0020H the size of array let take 5
```

```
;0021H - 10
;0022H -5
;0023H -12
;24 - 15
;25H - 6
;result = 15 at 0024H and stored in 0030H
;we should also stored the 16 bit address ie 0024H
;msb stored in 0031H and lsb stored in 0032H
   LXI H,0020H
      MOV C,M
      INX H
      MOV B,M
      SHLD 0031H
      DCR C
LOOP:  INX H
      MOV A,M
      CMP B
      JC SKIP
      SHLD 0031H
      MOV B,A
      SKIP:  DCR C
      JNZ LOOP
      LXI H,0030H
      MOV M,B
      HLT
```

Output:-



Data | Stack | KeyPad | Memory | I/O Ports

Start | OK

Address (Hex)	Address	Data
001E	30	0
001F	31	0
0020	32	0
0021	33	10
0022	34	5
0023	35	12
0024	36	15
0025	37	6
0026	38	0
0027	39	0
0028	40	0
0029	41	0

Line No	Assembler Message
0	Program assembled successfully

Q.6 Code to develop a subroutine to add two floating point number.

```
 ; Subroutine to add two floating-point quantities
; Input: HL points to the first floating-point number
;      DE points to the second floating-point number

ADD_FLOATS:  LXI H,0030;
   LXI D,0032;
   ; Load the exponents into registers A and B
   MOV A, M   ; Load exponent of the first number
   INX H      ; Move HL to the mantissa of the first number
   MOV B, M   ; Load exponent of the second number
   INX D      ; Move DE to the mantissa of the second number
   ; Calculate the difference in exponents
   SUB B
   MOV B, A   ; Store the difference in B
   ; Shift the mantissa of the number with the smaller exponent to the right
   MVI C, 0   ; Initialize C to 0 for later use
LOOP_SHIFT:  RRC       ; Right shift A (mantissa of first number)
```

RRC ; Right shift C (carry flag holds bit from A)

DCR B ; Decrement B (exponent difference)

JNZ LOOP_SHIFT

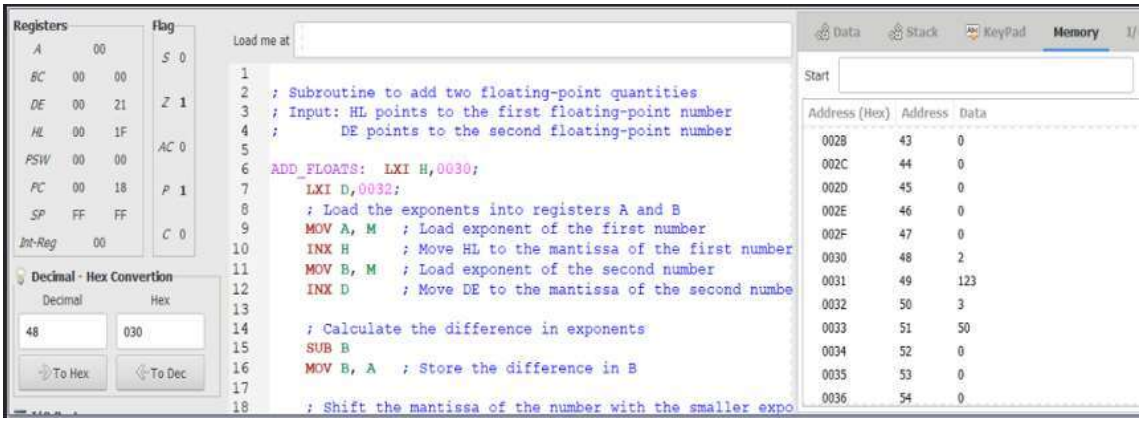
MOV A, M ; Load mantissa of the first number

ADD C ; Add the shifted mantissa of the second number

MOV M, A ; Store the result in the mantissa location

HLT;

Output:



Q.7 Program to add two double byte numbers.

; First 16-bit number (LSB in memory locations 2000H, MSB in 2001H)

LXI H, 2000H ; Load the address of the first number into HL

; Second 16-bit number (LSB in memory locations 2002H, MSB in 2003H)

LXI D, 2002H ; Load the address of the second number into DE

MOV A, M ; Load the LSB of the first number into the accumulator

ADD E ; Add the LSB of the second number

STAX D ; Store the result in memory (LSB of the result)

INX H ; Increment the HL pointer to point to the MSB of the first number

INX D ; Increment the DE pointer to point to the MSB of the second number

MOV A, M ; Load the MSB of the first number into the accumulator

ADC E ; Add the MSB of the second number along with the carry

STAX D ; Store the result in memory (MSB of the result)

HLT ; Halt the program

Output:-

004D	77	0
004E	78	0
004F	79	0
0050	80	30
0051	81	2
0052	82	112
0053	83	85
0054	84	50
0055	85	2
0056	86	0
0057	87	0

Line No	Assembler Message
0	Program assembled successfully

Q.8 Program to divide 4byte number with the another 4 byte number.

LXI H, 0050H ; Load the address of the dividend in memory into H

MOV A, M ; Load the first byte of the dividend into A

INX H ; Move to the next byte

MOV B, M ; Load the second byte of the dividend into B

INX H ; Move to the next byte

```
MOV C, M     ; Load the third byte of the dividend into C
INX H        ; Move to the next byte
MOV D, M     ; Load the fourth byte of the dividend into D

LXI H, 0060H ; Load the address of the divisor in memory into H
MOV E, M     ; Load the first byte of the divisor into E
INX H        ; Move to the next byte
MOV H, M     ; Load the second byte of the divisor into H
INX H        ; Move to the next byte
MOV L, M     ; Load the third byte of the divisor into L
INX H        ; Move to the next byte
MOV M, A     ; Load the fourth byte of the divisor into memory

LXI D, 5000H ; Load the address where the quotient will be stored
MVI A, 00H   ; Initialize A register to 0
MVI B, 00H   ; Initialize B register to 0
MVI C, 00H   ; Initialize C register to 0
MVI D, 00H   ; Initialize D register to 0

; Start the division loop
LOOP:  MOV A, M  ; Load the current byte of the dividend into A
   MOV H, E  ; Load the current byte of the divisor into H
   SUB H     ; Subtract the divisor byte from the dividend byte
   MOV M, A  ; Store the result back in memory
   INX D     ; Move to the next byte in the quotient
   JNC NOBORROW  ; If there's no borrow from the subtraction, don't borrow
   INR C     ; Increment C if there was a borrow
NOBORROW:  INX H     ; Move to the next byte in the divisor
   INX H     ; Move to the next byte in the dividend
   DCR L     ; Decrement L (number of bytes processed)
   JNZ LOOP  ; Continue the loop until all bytes are processed
```

HLT ; Halt the program

Output:-

Address (Hex)	Address	Data
0059	89	0
005A	90	0
005B	91	0
005C	92	0
005D	93	0
005E	94	2
005F	95	2
0060	96	2
0061	97	2
0062	98	2
0063	99	0

Line No	Assembler Message
0	Program assembled successfully

Q.9 Program to divide 8 -bit number with 8-bit number upt to fractional coefficient of 16 bit

```
MVI B, 52H  ; Load the dividend (8-bit number) into register B
MVI C, 17H  ; Load the divisor (8-bit number) into register C

MVI A, 00H   ; Initialize the high byte of quotient (D) to zero
MVI D, 00H   ; Initialize the low byte of quotient (A) to zero

XRA A   ; Clear the carry flag

DIV_LOOP: RLC B   ; Rotate the bits of B through carry (dividend)
   RLC D   ; Rotate the bits of D through carry (16-bit quotient)
```

```
    MOV E, A ; Copy the low byte of the quotient (A) to E for comparison

    CMP C   ; Compare C with A (divisor)
    JC NO_SUBTRACTION ; If no subtraction needed, skip

    SUB C   ; Subtract the divisor from A
    MOV A, E ; Restore the low byte of the quotient from E

    INR D   ; Increment the high byte of the 16-bit quotient (D)

NO_SUBTRACTION:  DCR C   ; Decrement the divisor
    JNZ DIV_LOOP ; If the divisor is not zero, continue

HLT     ; Halt the program
```

Q.10 Program to Adding first N natural number and store result on memory location x.

```
;GIVING THE VALUE OF N =10
;TAKING VALUE OF X = 0050H
MVI C,0AH;
MVI A,00H;
LOOP:  MOV B,C;
ADD B;
DCR C;
JNZ LOOP;
STA 0050H;
```

HLT;

Output:-

Address (Hex)	Address	Data
004C	76	0
004D	77	0
004E	78	0
004F	79	0
0050	80	55
0051	81	0
0052	82	0
0053	83	0
0054	84	0
0055	85	0
0056	86	0
0057	87	0
0058	88	0

Line No	Assembler Message
0	Program assembled successfully

Q.11 Program to Sort an array using bubble sort;

START: LXI H,0080H;

MVI D,00H;

MOV C,M;

DCR C;

INX H;

CHECK: MOV A,M;

INX H;

CMP M;

JC NEXTBYTE;

JZ NEXTBYTE;

MOV B,M;

MOV M,A;

DCX H;

```
MOV M,B;
INX H;
MVI D,01H;
NEXTBYTE: DCR C;
JNZ CHECK;
MOV A,D;
CPI 01H;
JZ START;
HLT;
```

Output:-

Address (Hex)	Address	Data
007D	125	0
007E	126	0
007F	127	0
0080	128	5
0081	129	1
0082	130	5
0083	131	11
0084	132	32
0085	133	50
0086	134	0
0087	135	0
0088	136	0

Line No	Assembler Message
0	Program assembled successfully

Q.12 Program to find the factorial of a number;

```
LXI H,0050H;
MOV B,M;
MVI D,01H;
FACTORIAL: CALL MULTIPLY;
```

```
DCR B;
JNZ FACTORIAL;
INX H;
MOV M,D;
HLT;
MULTIPLY: MOV E,B;
MVI A,00H;
MULTIPLY_LOOP: ADD D;
DCR E;
JNZ MULTIPLY_LOOP;
MOV D,A;
RET;
```

Output:-

Address (Hex)	Address	Data
004C	76	0
004D	77	0
004E	78	0
004F	79	0
0050	80	5
0051	81	120
0052	82	0
0053	83	0
0054	84	0
0055	85	0
0056	86	0
0057	87	0
0058	88	0

Line No	Assembler Message
0	Program assembled successfully

Q.13 Program to find the 1's and 2's compliment of a giving number;

```
LDA 0050H;
MVI B,01H;
CMA;
```

STA 0051H;

ADD B;

STA 0052H;

HLT;

Address (Hex)	Address	Data
004E	78	0
004F	79	0
0050	80	4
0051	81	251
0052	82	252
0053	83	0
0054	84	0
0055	85	0
0056	86	0
0057	87	0
0058	88	0
0059	89	0

Line No	Assembler Message
0	Program assembled successfully

Output:-

Q.14 Program to find addition of 2 four BYTE number;

;taking value 10,10,10,10 in 0100H to 0103H and

;taking value 10,20,30,40 in 0200H to 0203H

LDA 0100H;

LXI H,0200H;

MOV B,M;

ADC B;

STA 0300H;

INX H;

LDA 0101H;

MOV B,M;

ADC B;

STA 0301H;

INX H;

LDA 0102H;

MOV B,M;

ADC B;

STA 0302H;

INX H;

LDA 0103H;

MOV B,M;

ADC B;

STA 0303H;

HLT;

Address (Hex)	Address	Data
004E	78	0
004F	79	0
0050	80	4
0051	81	251
0052	82	252
0053	83	0
0054	84	0
0055	85	0
0056	86	0
0057	87	0
0058	88	0
0059	89	0

Line No	Assembler Message
0	Program assembled successfully

Output:-