

- Practical - I :-

Page _____
Date _____

Aim :- Write the working of 8085 simulator GINUSIM8085 and basic architecture of 8085.

- 8085 is an 8-bit microprocessor designed by Intel in 1977.
- It has below shown configurations :-
 - 8-bit data bus.
 - 16-bit address bus.
 - 16-bit program counter.
 - 16-bit stack pointer
 - Six 8-bit registers paired together namely : BC, DE, HL .
- It is used in mobile phones, washing machines, microwave, ovens, etc.
- 8085 microprocessor has following functional units ,

(1) Accumulator :-

- It is an 8-bit register used to perform arithmetic logic, I/O load/store operations.
- It is connected to internal data bus of ALU.

(2) Arithmetic & Logic Unit (ALU) :-

- As the name suggests, it performs arithmetic &

logical operations like Addition, Subtraction, AND, OR, NOT, etc. on 8-bit data.

(3) General Purpose registers :-

- These are basically 6 general purpose registers in 8085 microprocessor. They are B, C, D, E, H & L.
- Each register is capable to hold 8-bit data. These registers can work in pairs to hold 16-bit data & their pairing combination is like B-C, D-E & H-L

(4) Program counter :-

- It is a 16-bit register used to store the memory address location at the next instruction to be executed.

(5) Stack pointer :-

- It is a 16-bit register working like stack which increments and decrements by 2 while doing operations.

(6) Temporary registers :-

- It is an 8-bit register which holds temporary data of various operations.

(7) Flag register :-

→ It is an 8-bit register having five 1-bit flip-flops which holds either 0 or 1 depending upon the result stored in the Accumulator.

→ These are the set of 5 flip-flops :-

- (i) Sign (s)
- (ii) Zero (z)
- (iii) Auxiliary carry (Ac)
- (iv) Parity (P)
- (v) Carry (c).

(8) Instruction registers and Decoders :-

→ It is an 8-bit register when an instruction is fetched from memory then it is stored in the instruction register.

(9) Interrupt Control :-

→ When a microprocessor is executing a main program and whenever an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program.

(10) Serial Input / Output control :-

→ It controls the serial data communication by using these two instructions :-

SID (Serial Input Data)

SOD (Serial Output Data).

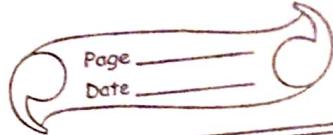
(11) Address buffer and address data buffer :-

→ The content stored in the stack pointer and program counter is loaded into the address buffer and address data buffer to communicate with the CPU.

(12) Address bus & Data bus :-

- Databus carries the data to be stored.
- It is bidirectional, whereas address bus carries the location where it should be stored and it is unidirectional.
- It is used to transfer the data of Address I/O devices.

-: Practical - 2 :-



D) Aim:- Write the 8085 program to add two 8-bit numbers stored in registers.

Program:-

```
MVI B, 04H ; Move value 04H to register B  
MOV A, B ; Move value from register B to A  
MVI C, 01H ; Move value 01H to register C.  
ADD C ; Add  $\Rightarrow A \leftarrow A + C$   
  
MOV D, A ; Move value from register A to D  
HLT
```

Output :- Before execution

| Register | Register |
|----------|----------|
| A | |
| B 04 | C 01 |
| D | E |

After execution

| Register | Register |
|----------|----------|
| A 05 | |
| B 04 | C 01 |
| D 05 | E |

Aim :- Write the 8085 program to add two 8-bit numbers stored in memory locations.

Program :- LDA 2000H ; Load Acc. with value stored at 2000H
MOV B,A ; Move value from Acc to register B.

LDA 2001H ; Load Acc. with value stored at 2001H
ADD B ; Add $\Rightarrow A \leftarrow A + B$.

STA 2005H ; Store Acc. content at 2005H location
in memory.
HLT.

Output :-

① Before execution

| (H) | Address | Data |
|------|---------|------|
| 2000 | 04 | |
| 2001 | 05 | |
| 2002 | | |
| 2003 | | |
| 2004 | | |
| 2005 | | |

② After execution

| (H) | Address | Data |
|-----|---------|------|
|-----|---------|------|

| | |
|------|----|
| 2000 | 04 |
|------|----|

| | |
|------|----|
| 2000 | 04 |
|------|----|

| | |
|------|----|
| 2001 | 05 |
|------|----|

| | |
|------|----|
| 2001 | 05 |
|------|----|

| | |
|------|--|
| 2002 | |
|------|--|

| | |
|------|--|
| 2002 | |
|------|--|

| | |
|------|--|
| 2003 | |
|------|--|

| | |
|------|--|
| 2003 | |
|------|--|

| | |
|------|--|
| 2004 | |
|------|--|

| | |
|------|--|
| 2004 | |
|------|--|

| | |
|------|--|
| 2005 | |
|------|--|

| | |
|------|----|
| 2005 | 09 |
|------|----|

3) Aim :- Write the 8085 program to add 16-bit numbers.

Program :-

| | |
|-----------|---|
| LDA 2001H | ; Load Acc. with value at 2001H |
| MOV B,A | ; Move value of Acc. to register B |
| LDA 2003H | |
| ADD B | ; Add $\Rightarrow A \leftarrow A + B$ |
| STA 2005H | ; Store the Acc. value at 2005H location |
| LDA 2000H | |
| MOV B,A | |
| LDA 2002H | |
| ADC B | ; Add with carry to value in register B and Accumulator |
| STA 2004H | |

HLT.

Output :-

| | |
|------------------------------|-----------------------------|
| ① <u>Before execution :-</u> | ② <u>After execution :-</u> |
|------------------------------|-----------------------------|

| (H) Address | Data | (H) Address | Data |
|----------------|------|----------------|------|
| 2000 | 13 | 2000 | 13 |
| 2001 | 45 | 2001 | 45 |
| 2002 | 31 | 2002 | 31 |
| 2003 | 54 | 2003 | 54 |
| 2004 | | 2004 | 44 |
| 2005 | | 2005 | 99 |

Aim :- Write the 8085 program to subtract two 8-bit numbers stored in registers.

Program :-

| | |
|------------|---|
| MVI B, 05H | ; Move value 05H into register B |
| MOV A, B | ; move value from register B to A. |
| MVI C, 02H | ; Move value 02H to register C. |
| SUB C | ; Subtract $\Rightarrow A \leftarrow A - C$ |
| MOV D, A | ; move value of Acc. to register D. |
| HLT. | |

Output :-

① Before execution :-

Registers Data Registers Data.

| | | | |
|---|----|---|----|
| A | | | |
| B | 05 | C | 02 |
| D | | E | |

② After execution :-

Registers Data Registers Data.

| | | | |
|---|----|---|----|
| A | 03 | | |
| B | 05 | C | 02 |
| D | 03 | E | |

5) Aim: Write the 8085 program to subtract two 8-bit numbers without using SUB instruction.

Program:

MVI B, 03H ; Move value 03H to register B.
MOV A, B ; Move value of register B to A.

MVI C, 07H ; Move value 07H to register C

CMA ; Complement the value of AC.
INR A ; Increment the value of AC by 1.
ADD C ; Add $\Rightarrow A \leftarrow A + C$.

MOV D, A ; Move value of AC to register D.
HLT.

Output:

① Before execution

Register Data Register Data

A

B 03 C 07

D

E

② After execution

Register Data Register Data

A 04

B 03 C 07

D 04 E

-:- Practical - 3 :-

Page _____
Date _____

Write the 8085 program to perform all logical operations...

D) Aim:- To perform AND operation of two 8-bit numbers stored on register.

Program:- MVI B, 04H ; Move value 04H to register B
MOV A, B ; Move value from register B to A

MVI C, 02H ; Move value 02H to register C
ANA C ; AND operation \Rightarrow A \leftarrow A.C

MOV D, A ; Move value from A to register D
HLT

Output:- ① Before execution

Registers Data Registers Data

A

B 04 C 02

D

E

② After execution

Registers Data Registers Data

A 00

B 04 C 02

D 00 E

Aim:- To perform OR operation of two 8-bit nos. without using ORA instruction.

Program:- MVI B, 02H ; Move value 02H to register B
MVI C, 03H ; Move value 03H to register A

MOV A, B ; Move value from register B to A

CMA ; Complement of value in A

MOV B, A ;

MOV A, C

CMA

MOV C, A

ANA B ; ANDing of value in B with A.

CMA

MOV D, A

HLT

Output:- ○ Before execution

Register Data Register Data

A

B 02

C 03

D

E

○ After execution

Register Data Register Data

A 03

B FD

C FC

D 03

E

③ Aim :- To perform XOR operation of two 8-bit nos. stored on memory location 2000H and 2001H.

Program:-

| | |
|--------------|------------------------------------|
| LXI H, 2000H | ; HL pointer points to 2000H loc. |
| MOV B, M | ; Value at 2000H will be move to B |
| MOV A, B | ; Move value in register B to Acc |

| | |
|----------|----------------------------------|
| INX H | ; Increment HL pointer. |
| MOV C, M | ; Move value at HL pointer to C. |

| | |
|-----------|----------------------------------|
| XRA C | ; X-OR operation of C with A. |
| STA 2005H | ; Store the Acc. value to 2005H. |
| HLT | |

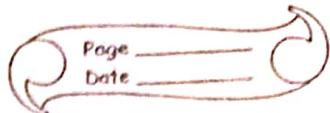
Output:-

| | |
|--------------------|-------------------|
| ① Before execution | ② After execution |
|--------------------|-------------------|

| (H) Address | Data |
|----------------|------|
| 2000 | 02 |
| 2001 | 08 |
| 2002 | |
| 2003 | |
| 2004 | |
| 2005 | |

| (H) Address | Data |
|----------------|------|
| 2000 | 02 |
| 2001 | 08 |
| 2002 | |
| 2003 | |
| 2004 | |
| 2005 | 0A |

- Practical - 4 :-



Aim:- Write the 8085 program to mask lower four bits of an 8-bit number stored in a register.

Program:-

| | |
|-------------|---------------------------------|
| MVI A, 25H | ; Move value 25H to Accumulator |
| MVI B, OFOH | ; Move value F0H to register B |

ANA B ; AND operation $\Rightarrow A \leftarrow A \cdot B$

MOV D,A

HLT

Output:-

① Before execution

| | | | |
|----------|------|----------|------|
| Register | Data | Register | Data |
|----------|------|----------|------|

| | | | |
|---|----|---|--|
| A | 25 | C | |
| B | F0 | D | |
| D | | E | |

② After execution

| | | | |
|----------|------|----------|------|
| Register | Data | Register | Data |
|----------|------|----------|------|

| | | | |
|---|----|---|--|
| A | 20 | C | |
| B | F0 | D | |
| D | 20 | E | |

Aim :- Write the 8085 program to set upper four bits of an 8-bit no. stored in a register.

Program :-

- MVI A, 25H ; Move value 25H to Accumulator
- MVI B, 0FH ; Move value 0FH to register B
- MVI C, 0FDH ; Move value F0H to register C

ANA B ; AND operation $\Rightarrow A \leftarrow A \cdot B$

ORA C ; OR operation $\Rightarrow A \leftarrow A + C$

MOV D, A ; Move the Acc. value to register D

HLT

Output :-

① Before execution

Register Data Register Data

A 25

B OF C F0

D E

② After execution

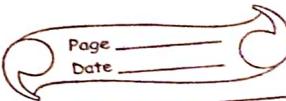
Register Data Register Data

A F5

B OF C F0

D F5 E

-: Practical - 5 :-



Aim :- Write the 8085 program to transfer block of data from one address to another.

Program :-

```
LXI H, 2000H ; HL pointer points at 2000H  
LXI D, 4000H ; DE pointer points at 4000H  
MVI C, 05H
```

LOOP:
MOV A, M ; Data at HL pointer moves to Ac.
STAX D ; Stores the Ac. value at DE pointer location.
INX H ; Increment HL pointer
INX D ; Increment DE pointer
DCR C ; Decrement value at C.
JNZ LOOP ; Jump if C ≠ 0 to label LOOP

HLT

Output:- ① Before execution :-

| <u>Address (H)</u> | <u>Data</u> |
|--------------------|-------------|
| 2000 | 3 |
| 2001 | 4 |
| 2002 | 5 |
| 2003 | 6 |
| 2004 | 7 |
| 2005 | |
| ; | |
| 4000 | |
| 4001 | |
| 4002 | |
| 4003 | |
| 4004 | |

② After execution :-

| <u>Address (H)</u> | <u>Data</u> |
|--------------------|-------------|
| 2000 | 3 |
| 2001 | 4 |
| 2002 | 5 |
| 2003 | 6 |
| 2004 | 7 |
| ; | |
| 4000 | 3 |
| 4001 | 4 |
| 4002 | 5 |
| 4003 | 6 |
| 4004 | 7 |

Aim :- Write 8085 program to transfer block of data from one address to another in reverse order.

Program :-

```
LXI H, 2000H ; HL pointer points at 2000H
LXI D, 4004H ; DE pointer points at 4004H
MVI C, 05H
```

LOOP :-

```
MOV A, M ; Moves value at HL pointer to Acc.
STAX D ; Stores the Acc. value at DE pointer location.
INX H
DCX D ; Decrement DE pointer.
DCR C
JNZ LOOP ; Jump if C≠0 or label LOOP.
```

HLT.

Output :-

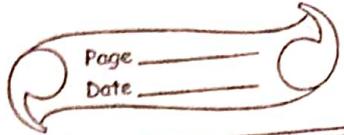
① Before execution :-

| Address(H) | Data |
|------------|------|
| 2000 | 3 |
| 2001 | 4 |
| 2002 | 5 |
| 2003 | 6 |
| 2004 | 7 |
| i | |
| 4000 | |
| 4001 | |
| 4002 | |
| 4003 | |
| 4004 | |

② After execution :-

| Address(H) | Data |
|------------|------|
| 2000 | 3 |
| 2001 | 4 |
| 2002 | 5 |
| 2003 | 6 |
| 2004 | 7 |
| i | |
| 4000 | 7 |
| 4001 | 6 |
| 4002 | 5 |
| 4003 | 4 |
| 4004 | 3 |

- Practical - 6 :-



Aim :- Write 8085 program to multiply two 8-bit nos.

Program :-

LXI H, 2000H ; HL pointer points at 2000H
MOV A, M ; Move data at HL to Acc.

LXI H, 2001H ; HL pointer points at 2001H
MOV B, M ; Move data at HL to register B

DCR B ; Decrement data of B by 1
DCX H ; Decrement HL pointer

LOOP: ADD M ; $A \leftarrow A + M$

DCR B

JNZ LOOP ; Jump if B ≠ 0 at label LOOP

STA 2002H ; Store the value of Acc. at 2002H

HLT

Output :- ① Before execution :- ② After execution :-

| Address ^(H) | Data |
|------------------------|------|
| 2000 | 2 |
| 2001 | 4 |
| 2002 | |

| Address ^(H) | Data |
|------------------------|------|
| 2000 | 2 |
| 2001 | 4 |
| 2002 | 8 |