Name: K.V.R saicharan
Roll no: 3200227720012
1st semester
Sub: computer organisation lab

12.) a) Design and Implement flip-flops ?

Flip-flops:

Flip flops are an application of logic gates. A flip-flop circuit can remain in a binary state indefinitely (as long as power is delivered to the circuit) until directed by an input signal to switch states.

S-R flip-flop stands for set reset flip-flops.

The set reset I flip-flop consists of two NOR gates and also two NAND gates. These flip-flops are also called S-R Latch The design of these flip flops also includes two inputs, called the set [S] and Reset [R]. There are also two outputs, Q and Q'.

Clocked sr flipflop: The operation of a basic flip-flop can be modified by providing an additional control input that determines when the state of the circuit is to be changed.

The limitation with a S-R flip-flop using NOR and nano gate is the invalid state. This problem can be overcome by using a stable SR flip-flop that can change outputs when certain invalid states are met, regardless of the condition of either the Set or the Reset inputs.

A clock pulse is given to the inputs of the AND Gate. If the value of the clock pulse is '0', the outputs of both the AND Gates remain

'0'.

12.)B.) Write a 8085 program for multiplication of two 8 bit numbers.

Aim:

Algorithm 1.

We are taking adding the number 43 seven(7) times in this example. 2. As the multiplication of two 8 bit numbers can be maximum of 16 bits so we need register pair to store the result.

Program:

Address	HEX Codes	Labels	memories	comments
7000	21,00,80		LXI H,8000H	Load first operand address
7003	46		MOV B,M	Store first operand
7004	23		INX H	Increase HL pair
7005	A7		XRA A	clear accumulator
7006	47		MOV C A	store 00H at register
7007	86	LOOP	ADD M	Add memory element
7008	D2,0C,70		jnc skip	when carry flag is 0
700B	0C		inr C	Increase C when carry is 1
700C	05	skip	DCR B	decrese B register
700D	C2,07,70		jnz Loop	jumpto loop

Output:

address	data
. .	.
. .	.

8pt;">.

8050	93
8051	D0

12.)C.) Write a 8085 program for Smallest number from an array

aim:

Algorithm:

1. Load the address of the first element of the array in HL pair
2. Move the count to B - reg.
3. Increment the pointer.
4. Get the first data in A - reg.
5. Decrement the count.
6. Increment the pointer.
7. Compare the content of memory addressed by HL pair with that of A - reg.
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 A - reg.
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:

```
LXI H,5000  ;  Set pointer for array
MOV B,M     ;  Load the Count
INX H
MOV A,M     ;  Set 1st element as largest
```

```
data
DCR B ;          Decrement the count
LOOP: INX H
CMP M ;          If A-reg < M go to AHEAD
JC AHEAD
MOV A,M ;        Set the new value as smallest
AHEAD DCR B
JNZ LOOP ;       Repeat comparisons till count = 0
sta 6000 ;       Store the largest value at 6000
HLT
```

input: Data 0: 05H in memory location 5000
Data 1: 0AH in memory location 5001 Data 2: 71H in memory location 5002 Data 3: 17H in memory location 5003 Data 4: 26H in memory location 5004 Data 6: 7EH in memory location 5005
Output: Largest Data 0AH in memory location 6000