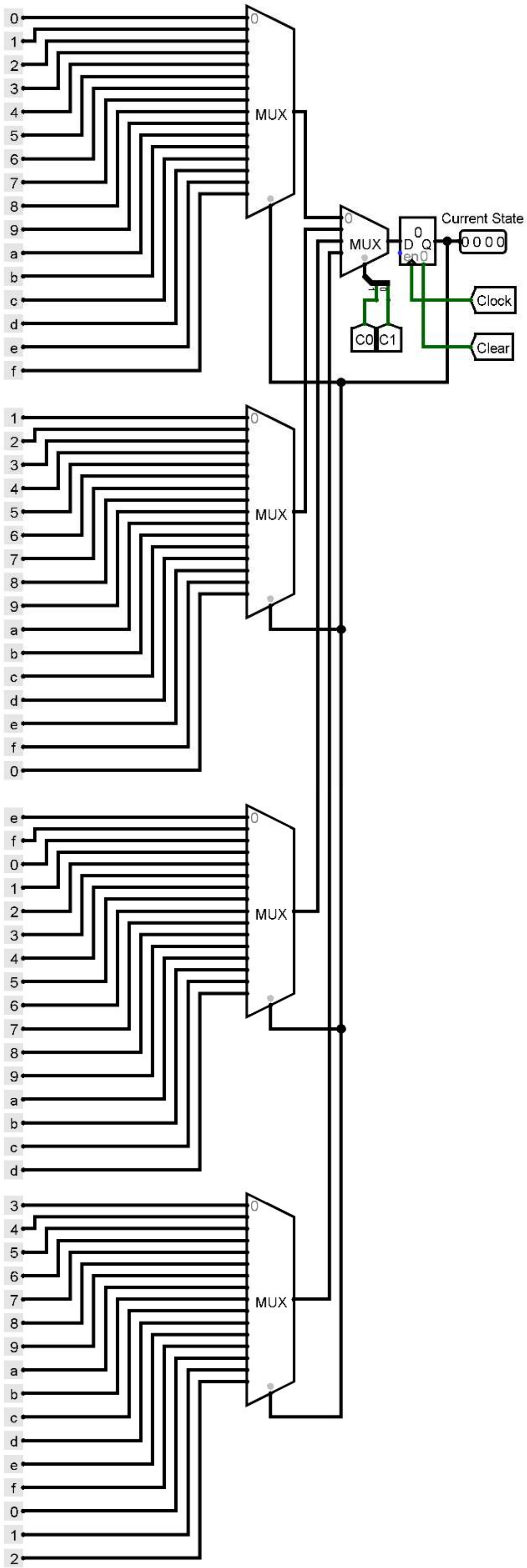
CS226 - Switching Theory Lab - 11

Name: M. Maheeth Reddy **Roll No.:** 1801CS31

Ans 1: **Circuit Diagram**

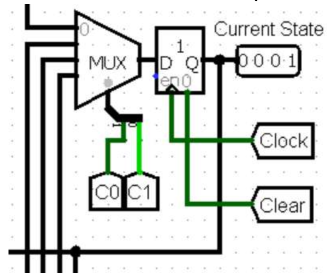


Test Run

Initially, Counter reads 0

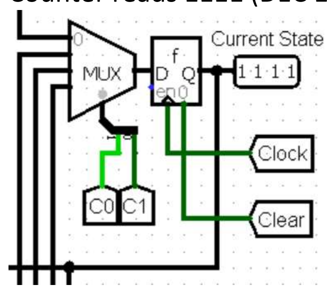
Step 1: Set C1 = 0, C0 = 1, and toggle the clock.

- Value stored is incremented by 1
- Counter reads 0001 (DEC 1)



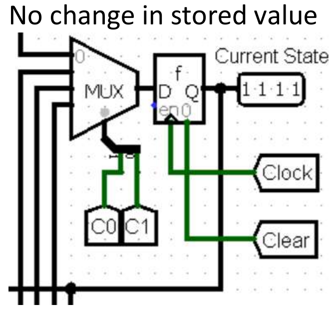
Step 2: Set C1 = 1, C0 = 0, and toggle the clock.

- Value stored is decremented by 2
- Counter reads 1111 (DEC 15)



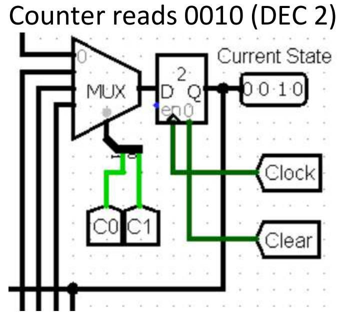
Step 3: Set C1 = 0, C0 = 0, and toggle the clock

- No change in stored value



Step 4: Set C1 = 1, C0 = 1 and toggle the clock

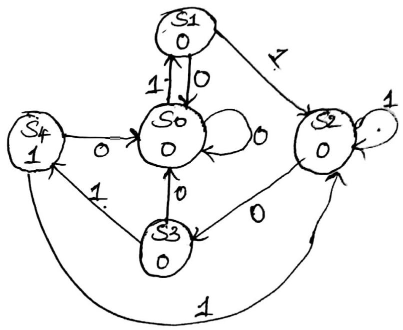
- Value stored is incremented by 3
- Counter reads 0010 (DEC 2)



Ans 2:

Moore FSM to detect 1101

State Diagram



State Encoding

State	Code
S0	000
S1	001
S2	010
S3	011
S4	100

Output Table

Current State S_2	S_1	S_0	Output Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1

Implementation using JK Flip Flop:

State Transition Table

Current State S_2	S_1	S_0	Inputs A	Next State S_2'	S_1'	S_0'	J2	K2	J1	K1	J0	K0
0	0	0	0	0	0	0	0	X	0	X	0	X
0	0	0	1	0	0	1	0	X	0	X	1	X
0	0	1	0	0	0	0	0	X	0	X	X	1
0	0	1	1	0	1	0	0	X	1	X	X	1
0	1	0	0	0	1	1	0	X	X	0	1	X
0	1	0	1	0	1	0	0	X	X	0	0	X
0	1	1	0	0	0	0	0	X	X	1	X	1
0	1	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	0	0	X	1	0	X	0	X
1	0	0	1	0	1	0	X	1	1	X	0	X
1	0	1	0	0	0	0	X	1	0	X	X	1
1	0	1	1	0	0	0	X	1	0	X	X	1
1	1	0	0	0	0	0	X	1	X	1	0	X
1	1	0	1	0	0	0	X	1	X	1	0	X
1	1	1	0	0	0	0	X	1	X	1	X	1
1	1	1	1	0	0	0	X	1	X	1	X	1

K-Map Simplification

S_2S_1 \ S_0A	00	01	11	10
00	X	X	1	1
01	X	X	1	1
11	X	X	1	1
10	X	X	1	1

$$\boxed{K_0 = 1}$$

S_2S_1 \ S_0A	00	01	11	10
00	X	X	X	X
01			1	1
11	1	1	1	1
10	X	X	X	X

$$\boxed{K_1 = S_2 + S_0}$$

S_2S_1 \ S_0A	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	1	1	1	1
10	1	1	1	1

$$\boxed{K_2 = 1}$$

S_2S_1 \ S_0A	00	01	11	10
00		1	X	X
01	1		X	X
11			X	X
10			X	X

$$J_0 = \bar{S}_2\bar{S}_1A + \bar{S}_2S_1\bar{A}$$

$$\boxed{J_0 = \bar{S}_2(S_1 \oplus A)}$$

S_2S_1 \ S_0A	00	01	11	10
00			1	
01	X	X	X	X
11	X	X	X	X
10		1		

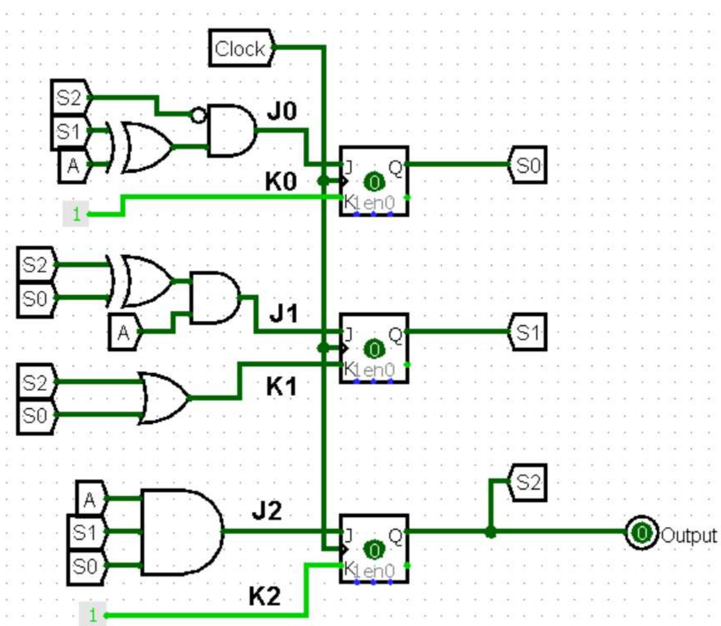
$$J_1 = \bar{S}_2S_0A + S_2\bar{S}_0A$$

$$\boxed{J_1 = A(S_0 \oplus S_2)}$$

S_2S_1 \ S_0A	00	01	11	10
00				
01			1	
11	X	X	X	X
10	X	X	X	X

$$\boxed{J_2 = S_1S_0A}$$

Logic Diagram



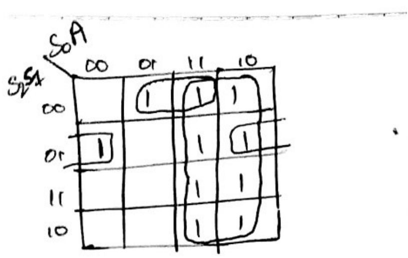
Implementation using T Flip-Flop

State Transition Table

Moore FSM State Transition Table

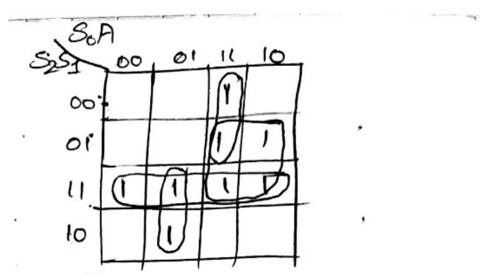
Current State S_2	S_1	S_0	Inputs A	Next State S_2'	S_1'	S_0'	T2	T1	T0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0	1
0	0	1	1	0	1	0	0	1	1
0	1	0	0	0	1	1	0	0	1
0	1	0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0	1	1
0	1	1	1	1	0	0	1	1	1
1	0	0	0	0	0	0	1	0	0
1	0	0	1	0	1	0	1	1	0
1	0	1	0	0	0	0	1	0	1
1	0	1	1	0	0	0	1	0	1
1	1	0	0	0	0	0	1	1	0
1	1	0	1	0	0	0	1	1	0
1	1	1	0	0	0	0	1	1	1
1	1	1	1	0	0	0	1	1	1

K-Map Simplification



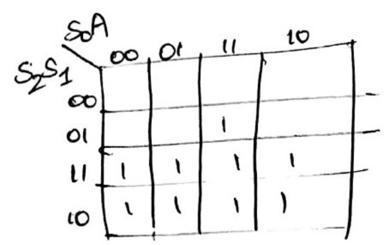
$T_0 = \bar{S_2}\bar{S_1}A + \bar{S_2}S_1\bar{A}$

$\boxed{T_0 = \bar{S_2}(S_1 \oplus A)}$



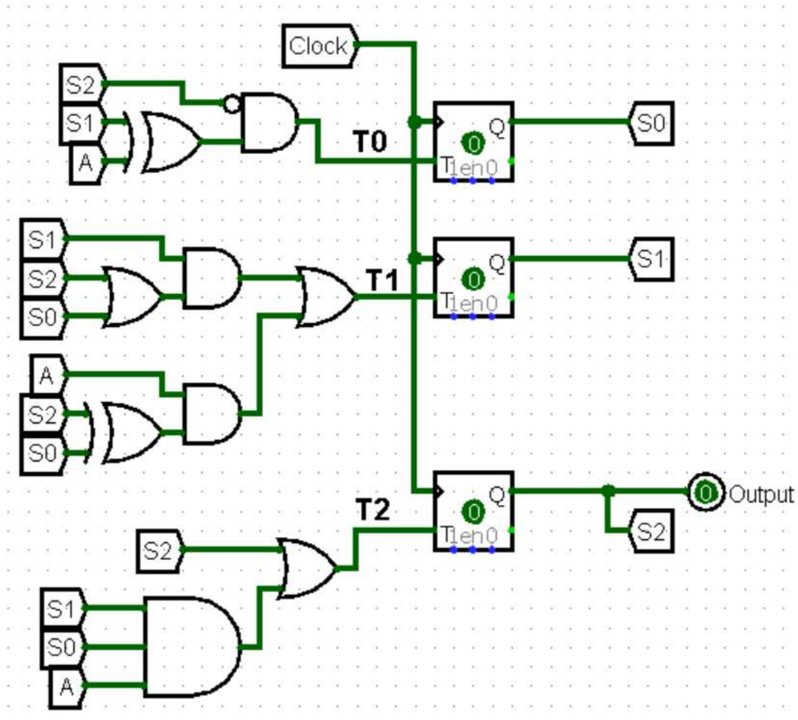
$T_1 = S_1S_0 + S_2S_1 + S_2\bar{S_0}A + \bar{S_2}S_0A$

$\boxed{T_1 = S_1(S_2+S_0) + A(S_2 \oplus S_0)}$



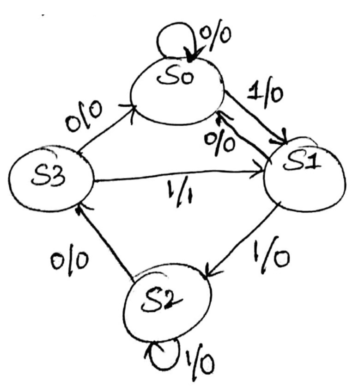
$\boxed{T_2 = S_2 + S_1S_0A}$

Logic Diagram



Mealy FSM to detect 1101

State Diagram



State Encoding

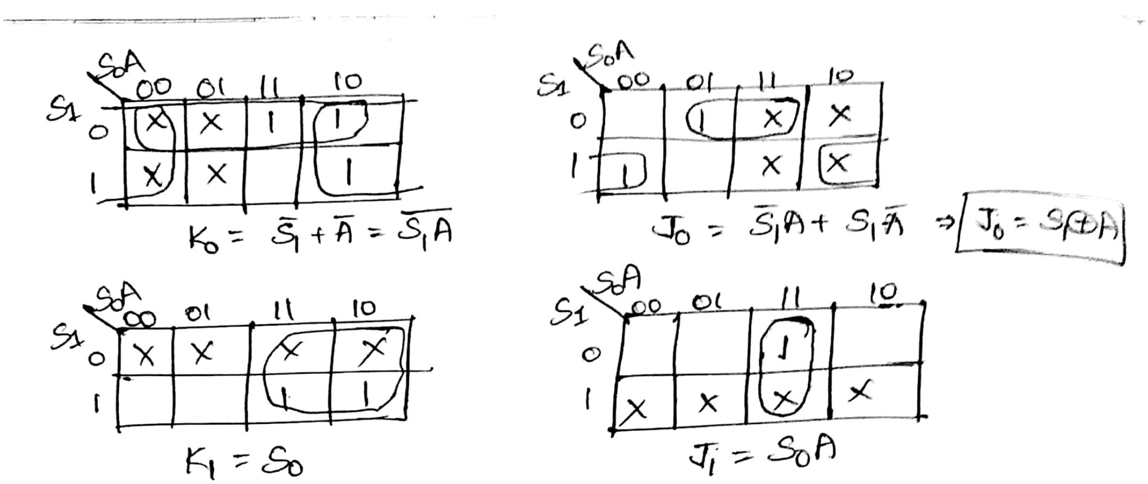
State	Code
S0	00
S1	01
S2	10
S3	11

Implementation using JK Flip Flop:

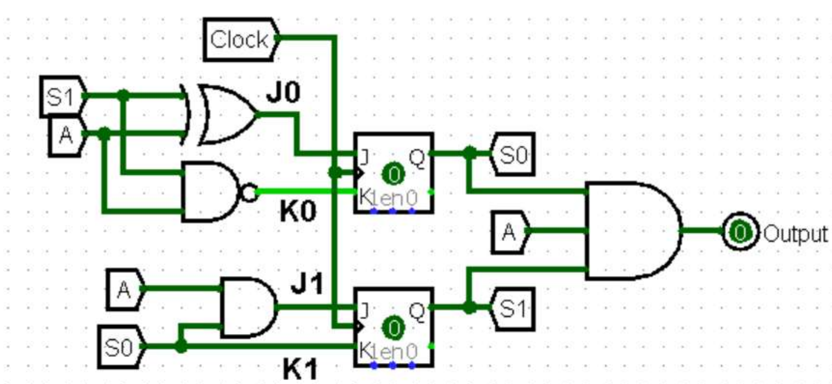
State Transition Table

Current State S_1	S_0	Input A	Next State S_1'	S_0'	Output Y	J1	K1	J0	K0
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	0	X	1	X
0	1	0	0	0	0	0	X	X	1
0	1	1	1	0	0	1	X	X	1
1	0	0	1	1	0	X	0	1	X
1	0	1	1	0	0	X	0	0	X
1	1	0	0	0	0	X	1	X	1
1	1	1	0	1	1	X	1	X	0

K-Map Simplification (Output $Y = S_1 S_0 A$)



Logic Diagram

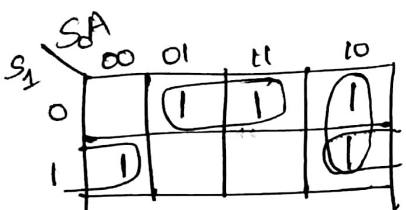


Implementation using T Flip Flop:

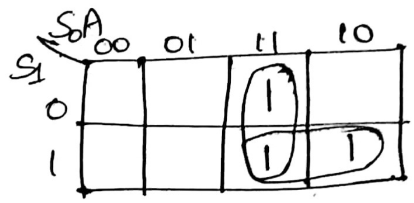
State Transition Table

Current State S_1	Current State S_0	Input A	Next State S_1'	Next State S_0'	Output Y	T_1	T_0
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	0	0	1
0	1	1	1	0	0	1	1
1	0	0	1	1	0	0	1
1	0	1	1	0	0	0	0
1	1	0	0	0	0	1	1
1	1	1	0	1	1	1	0

K-Map Simplification (Output $Y = S_1 S_0 A$)

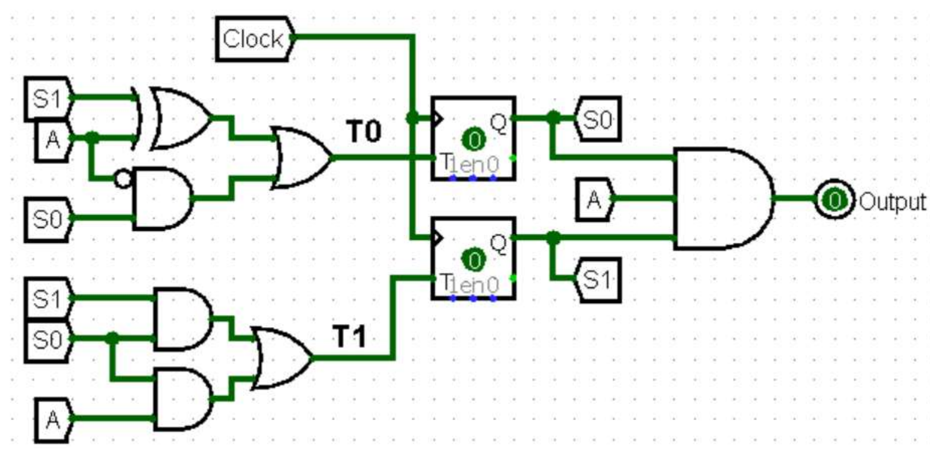


$T_0 = \bar{S_1}A + \bar{S_0}\bar{A} + S_1\bar{A}$



$T_1 = S_0A + S_1S_0$

Logic Diagram



Ans 3:

Average of history of temperatures stored in 4 8-bit registers

Method for Division: Shift the bits of the sum twice to the right. This is equivalent to division by 4. We are calculating only quotient but not remainder.

Circuit Notation

Clock stands for Clock

Hist_En, if high, enables entry of a new temperature

New Value, to input new temperature reading for storing in registers

Sum_En, if high, calculates sum of temperatures stored in current clock cycle

Load, if high loads sum of temperatures of current clock cycle, into the shift register

Shift, is the bit that is placed at shift register MSB while shifting bits for division

Registers 1,2,3,4 store history of temperatures

Reset_hist, erases all data stored in registers 1,2,3,4

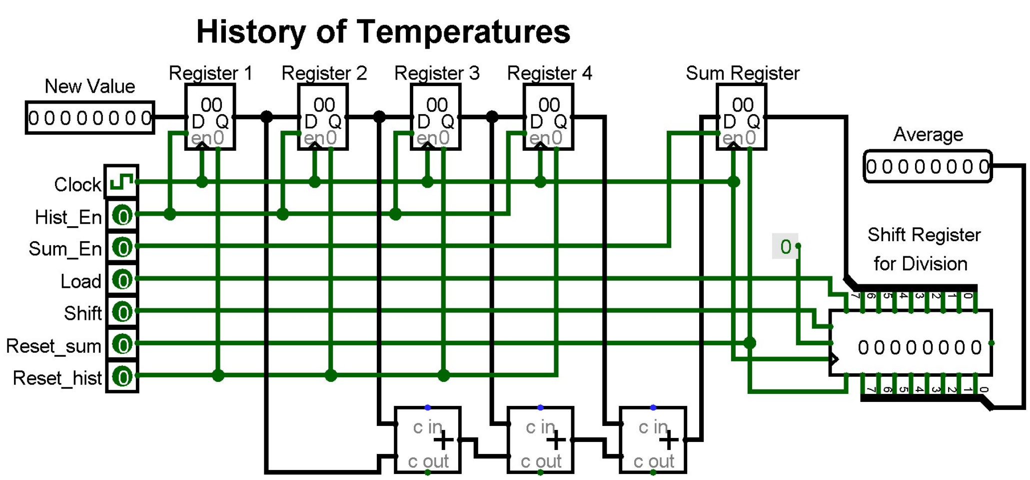
Reset_sum, erases sum for calculating it once again for different temperatures in the next iteration

Sum Register stores the sum of temperatures stored in registers 1 to 4

Shift Register is used for dividing the sum in Sum Register

Average pin shows average of the temperatures in registers 1 to 4

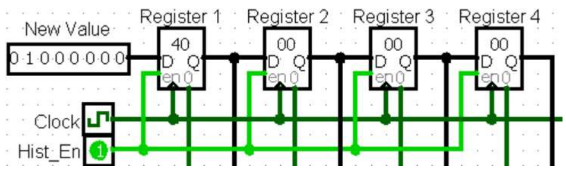
Circuit Diagram



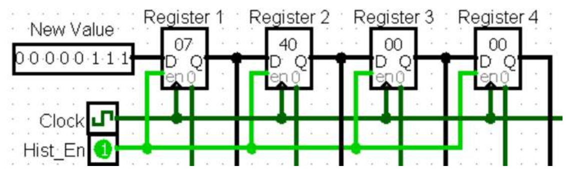
Test Run

1. Initially, there is nothing stored in the registers.
2. Set **New_Value** = 0100 0000, **Hist_En** = 1, and toggle the clock. The value is stored in Register 1.

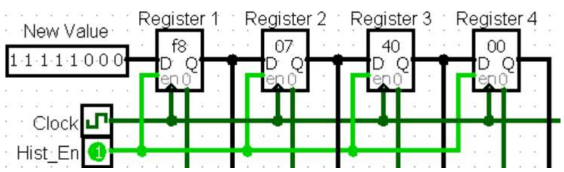
3. Set **New_Value** = 0000 0111 and toggle the clock. The value is stored in Register 1 and previous value is shifted to register 2.
4. Set **New_Value** = 1111 1000, and toggle the clock. The value is stored in Register 1 and previous values are shifted towards right side.
5. Set **New_Value** = 1001 1001, and toggle the clock. The value is stored in Register 1 and previous values are shifted towards right side.



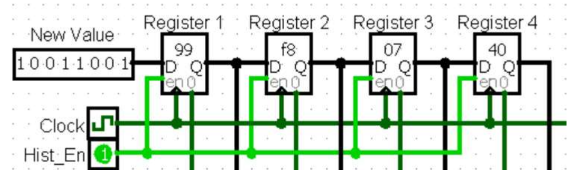
After Step 1



After Step 2

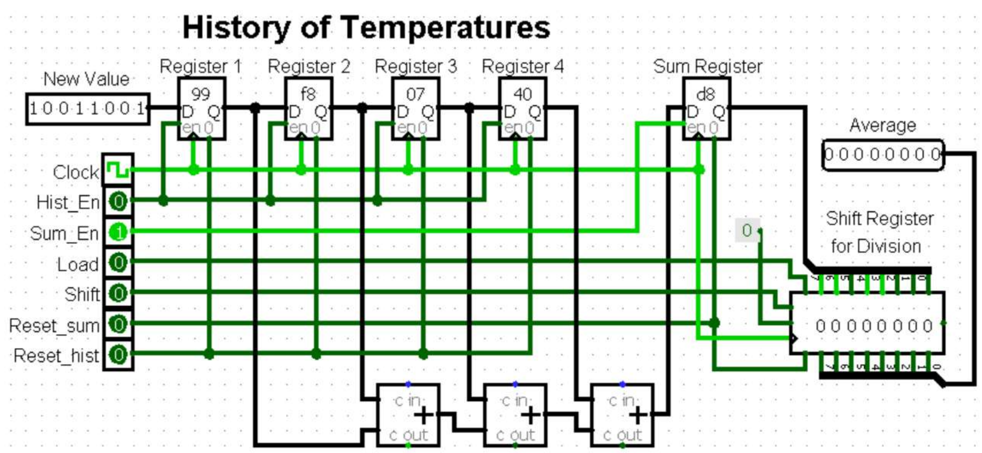


After Step 3

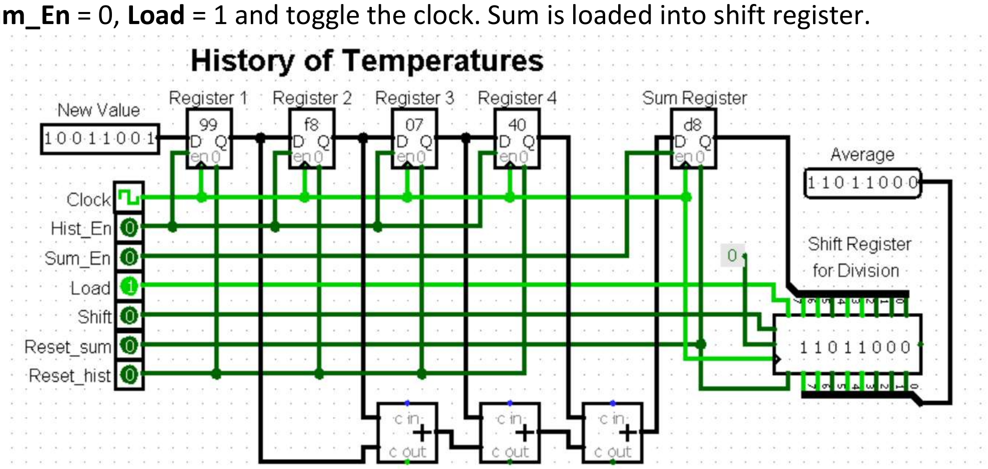


After Step 4

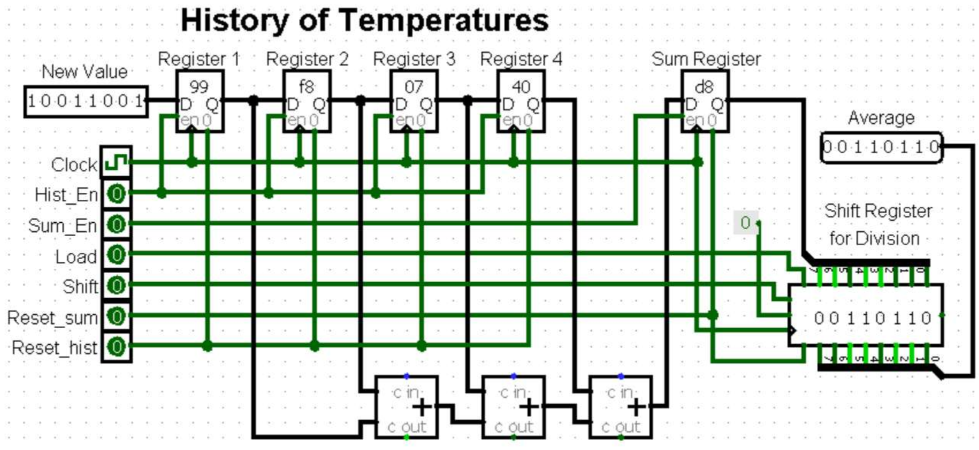
6. Now, set **Hist_En** = 0, **Sum_En** = 1 and toggle the clock. The sum of all temperatures present in the registers at this moment, is stored in Sum Register.



7. Set **Sum_En** = 0, **Load** = 1 and toggle the clock. Sum is loaded into shift register.



8. Set **Load** = 0, **Shift** = 1 and toggle the clock twice. The output shows the average values of the temperatures stored in the registers for current clock cycle. Set Shift = 0 before further use.



9. To store a new temperature value, set **New_Value** to desired value, **Hist_En** = 1, and toggle the clock.
10. To re-calculate the average, set **Reset_sum** = 1 to clear previous sum, **Sum_En** = 1, and toggle the clock. Now load the new sum into shift register and calculate the average as mentioned in steps 7,8,9.
11. To erase all stored values after step 8, set **Reset_sum** = 0 and **Reset_hist** = 0.