ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 242E

DIGITAL CIRCUITS LABORATORY
HOMEWORK REPORT

HOMEWORK NO	:	2
HOMEWORK DATE	:	25.03.2023
LAB SESSION	:	FRIDAY - 14.00
GROUP NO	:	G3

GROUP MEMBERS:

150200054	:	ASLI YEL
150190028	:	SEVİM EFTAL AKŞEHİRLİ

SPRING 2021

Contents

1 PRELIMINARY

1.1 Flip-Flops and Their Significance in Digital Circuits

Flip-flops are simple memory units that can hold a single bit of data as long as the device is powered. They are activated by a clock signal, and can have one or more data inputs, a single output, and control inputs, depending on their type. The Q output displays the current state or value of the memory unit, and the next value of the output (Q(t+1)) is determined by the current state, current inputs, and clock signal (CLK).

Flip-flops are highly beneficial components in modern electronics. Their capability to store data even in the absence of a power supply makes them suitable for memory devices. Additionally, they are valuable for regulating data flow by holding it in a register until it is required. Moreover, they can produce complex waveforms and clock signal, enabling their usage in a wide range of electronic circuits. Overall, flip-flops are crucial in electronics by providing efficient data storage, processing, and control.

1.2 Differences between Latches and Flip-Flops

To point out the differences between Latch and Flip-flops:
A latch is an electronic circuit that has two stable states and can be set or reset by an input signal, and its output changes continuously based on the inputs. In contrast, a flip-flop is a circuit that has two stable states and changes its output state only when it receives a clock signal.
Latches are used in applications that require fast response times, as they are asynchronous circuits that don't rely on a clock signal. Flip-flops, which are synchronous circuits, are used in applications that require timing and signal synchronization.
Latches are simpler and require fewer components than flip-flops, but they are less reliable and more sensitive to glitches and noise. The output of a latch changes immediately with the input signal, while the output of a flip-flop changes only on the rising or falling edge of the clock signal, making flip-flops less prone to errors caused by input changes that happen between clock cycles.

1.3 How an SR Latch Works and Its Input Functionalities

An SR latch is a single bit memory device build up with two cross-coupled NOR or NAND gates, which form a feedback loop. It has two stable states, conventionally named as Q and Q$_{\mathrm{N}}$, which are the complements of each other. SR states for Set and Reset, two primary inputs of an SR latch.

All other latches and flip-flops can be built from this fundamental memory device by adding external gates. The input S is used to store a “1” to the latch,input S = 1 “sets” the output to Q = 1. The input R is used to write a “0” to the latch, input R = 0 “resets” the output to Q = 0. If both inputs are “0”, the SR latch preserves its state. Both inputs should not be “1” at the same time because it will make Q and Q(t+1) same and it is forbidden case. The next value of the output Q(t+1) depends on the inputs and current output Q current state. Characteristic equation for S-R latch is: Q(t+1) = S + Q(t)R’ (SR = 0) [1]. (442)

1.4 Truth Table of an SR Latch without an Enable Input

S	R	Q(t)	Q(t+1)		
0	0	0	0	Q(t)	No Change
0	0	1	1		
0	1	0	0	0	Reset
0	1	1	0		
1	0	0	1	1	Set
1	0	1	1		
1	1	0	ϕ	ϕ	Forbidden Inputs
1	1	1	ϕ		

Table 1: Truth table for an SR Latch without an Enable Input

When an SR latch is implemented with two cross-coupled NOR gates like in Figure1, it has a truth table as the Table1 above. However when we implement an SR latch with only two cross-coupled NAND gates like in the Figure2, what we need to pay attention is this time setting inputs as $\bar{S}$ and $\bar{R}$, not as S and R, gives the same truth table. Therefore, by adding two additional NAND gates like in the Figure3, an SR latch which consists of only NAND gates and matches with same truth table with inputs S and R can be implemented.

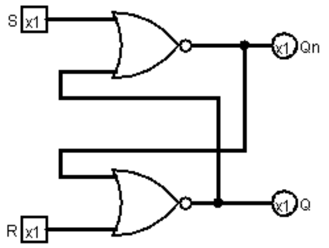


Figure 1: An SR Latch with two NOR Gates

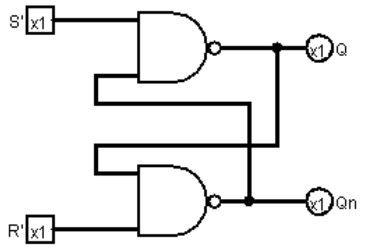


Figure 2: An SR Latch with two NAND Gates

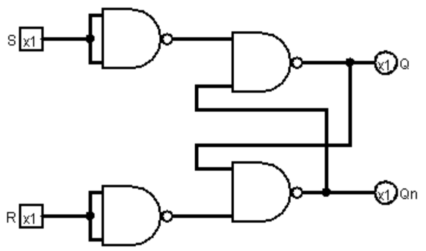


Figure 3: An SR Latch with only NAND Gates

1.5 Truth Table of an SR Latch with an Enable Input

E	S	R	Q(t)	Q(t+1)		
0	ϕ	ϕ	0	0	Q(t)	Not Enabled
0	ϕ	ϕ	1	1		
1	0	0	0	0	Q(t)	No Change
1	0	0	1	1		
1	0	1	0	0	0	Reset
1	0	1	1	0		
1	1	0	0	1	1	Set
1	1	0	1	1		
1	1	1	0	ϕ	ϕ	Forbidden Inputs
1	1	1	1	ϕ		

Table 2: Truth table for an SR Latch with an Enable Input

1.6 Truth Table of a D Flip-Flop

D	CLK	Q(t)	Q_N(t)	Q(t+1)		Q_N(t+1)	
0	rising	0	1	0	0	1	1
0	rising	1	0	0		1	
1	rising	0	1	1	1	0	0
1	rising	1	0	1		0	
ϕ	0	0	1	0	Q(t)	1	Q_N(t)
ϕ	0	1	0	1		0	
ϕ	1	0	1	0	Q(t)	1	Q_N(t)
ϕ	1	1	0	1		0	

Table 3: Truth table for a D Flip-Flop

1.7 Truth Table of a JK Flip-Flop

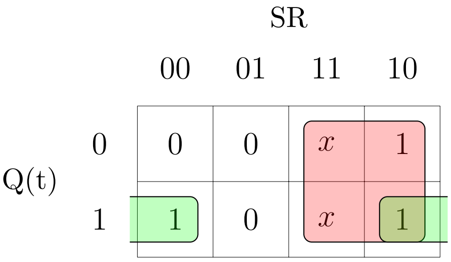
J	K	CLK	Q(t)	Q_N(t)	Q(t+1)		Q_N(t+1)	
0	0	rising	0	1	0	Q(t)	1	Q_N(t)
0	0	rising	1	0	1		0	
0	1	rising	0	1	0	0	1	1
0	1	rising	1	0	0		1	
1	0	rising	0	1	1	1	0	0
1	0	rising	1	0	1		0	
1	1	rising	0	1	1	Q_N(t)	0	Q(t)
1	1	rising	1	0	0		1	
ϕ	ϕ	0	0	1	0	Q(t)	1	Q_N(t)
ϕ	ϕ	0	1	0	1		0	
ϕ	ϕ	1	0	1	0	Q(t)	1	Q_N(t)
ϕ	ϕ	1	1	0	1		0	

Table 4: Truth table for a JK Flip-Flop

2 EXPERIMENT

2.1 Part 1 - SR Latch (only NAND Gates)

In order to find the characteristic equation of an SR Latch, we created a Karnaugh Map for Q(t+1) = f(S, R, Q(t)) according to the true generating points of truth table we constructed earlier (Table1 from section 1.4).



Q(t) \ SR	00	01	11	10
0	0	0	x	1
1	1	0	x	1

Figure 4: Karnaugh Map for F_1

This Karnaugh Map has two prime implicants, which are also distinguished prime implicants. Therefore, we were able to write the cahacteristic equation of an SR Latch simply by ORing this two prime implicants. What we got is:

Q(t+1) = S + Q(t)$\bar{R}$ (SR = 0)

SR = 0 expression is also added to imply that S and R cannot both be 1 at the same time.

The internal structure of the SR Latch we implemented is as follows.

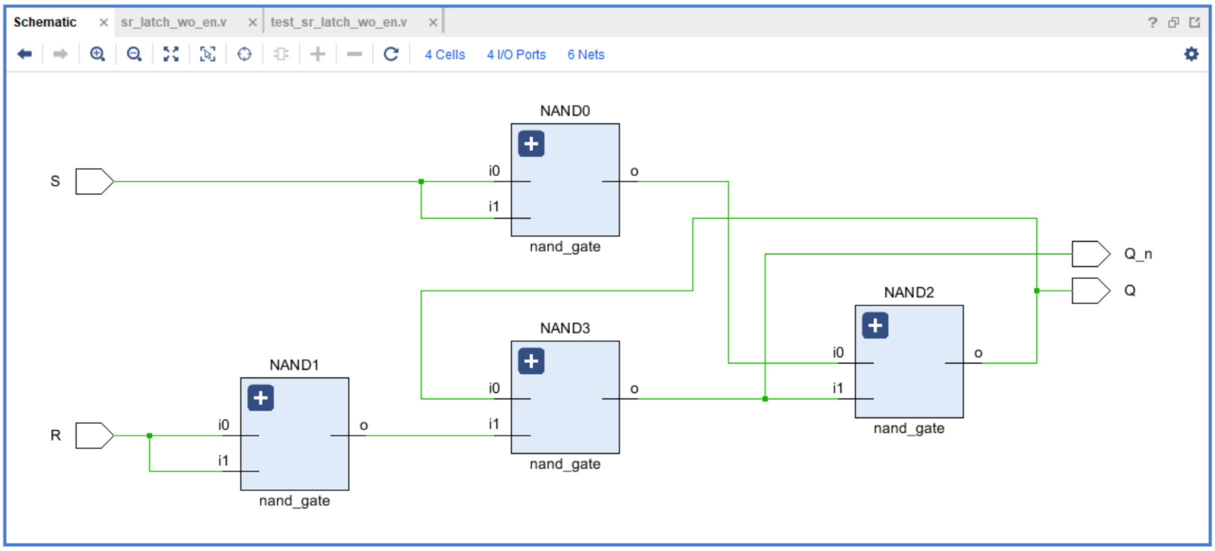


Figure 5: RTL Schematic of SR Latch without Enable Input

And here are the simulation results we observed to verify that it works correctly.

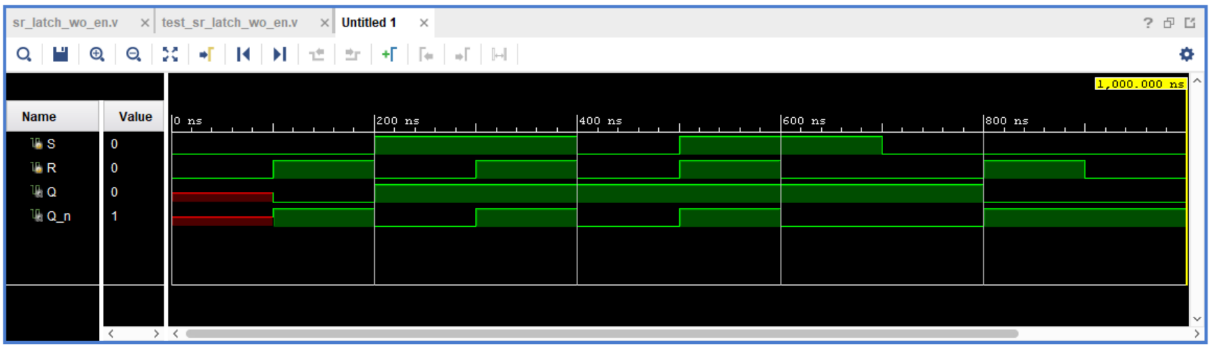


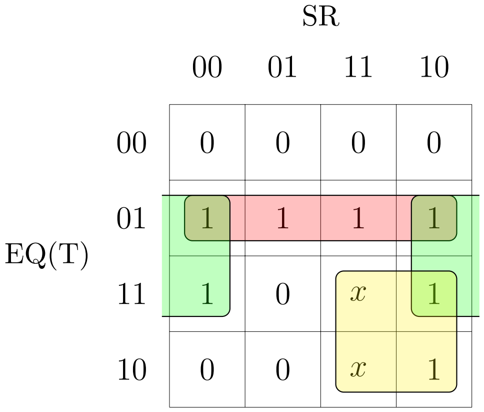
Figure 6: Simulation of SR Latch without Enable Input

We concluded that the SR Latch we implemented works as it should work.

We also can see that both outputs Q and Qn gets the value of "1" when we input S=1 and R=1. The fact that Q and Qn are meant to be the complement of each other clarifies why this input combination is not allowed.

2.2 Part 2 - SR Latch with Enable input (only NAND Gates)

In order to find the characteristic equation of an SR Latch with Enable, we created a Karnaugh Map for Q(t+1) = f(S, R, E, Q(t)) according to the true generating points of truth table we constructed earlier (Table2 from section 1.5).



EQ(T) \ SR	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	0	x	1
10	0	0	x	1

Figure 7: Karnaugh Map for F_1

This Karnaugh Map has three prime implicants, which are also distinguished prime implicants. Therefore, we were able to write the cahacteristic equation of an SR Latch with Enable input simply by ORing this three prime implicants. What we got is:

$Q(t+1) = ES + Q(t)\bar{R} + \bar{E}Q(t)$

The internal structure of the SR Latch with Enable input we implemented is as follows.

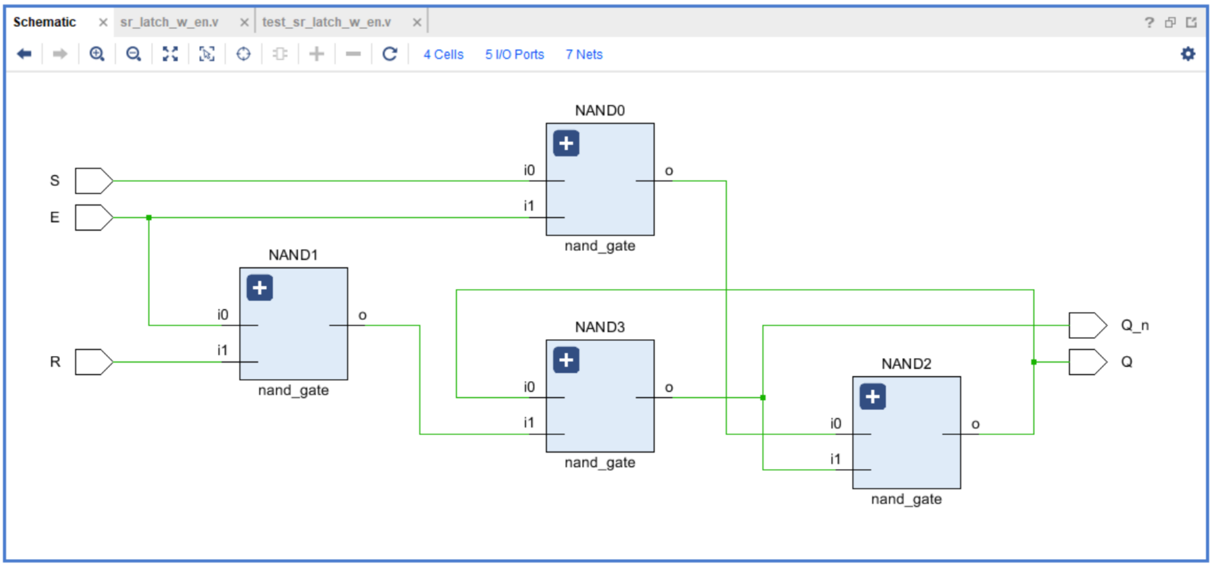
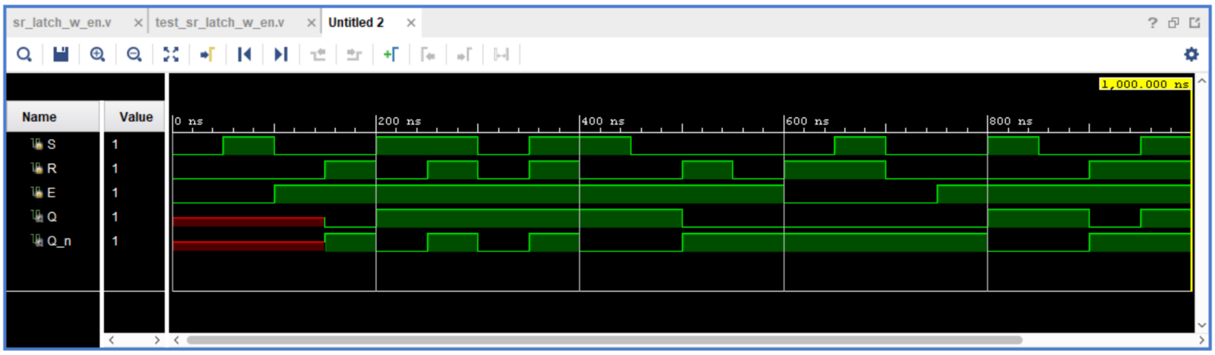


Figure 8: RTL Schematic of SR Latch with Enable Input

And here are the simulation results we observed to verify that it works correctly.



Name	Value
S	1
R	1
E	1
Q	1
Q_n	1

Figure 9: Simulation of SR Latch with Enable Input

We concluded that the SR Latch with Enable input we implemented works as it should work.

We can see that when Enable is set to "1" this latch operates just like the basic SR Latch which does not have an Enable input. On the other hand, when Enable is set to "0" the latch preserves its value regardless of the S and R inputs. This is the only difference revealed by the existence of the Enable input.

2.3 Part 3 - D Flip-Flop from D-Latches

In order to implement a negative edge triggered D flip flop, first we implemented D latches with Enable input.

The structure and the simulation results of implemented D Latch, which basically nothing but an SR latch with inverted inputs, is as follows.

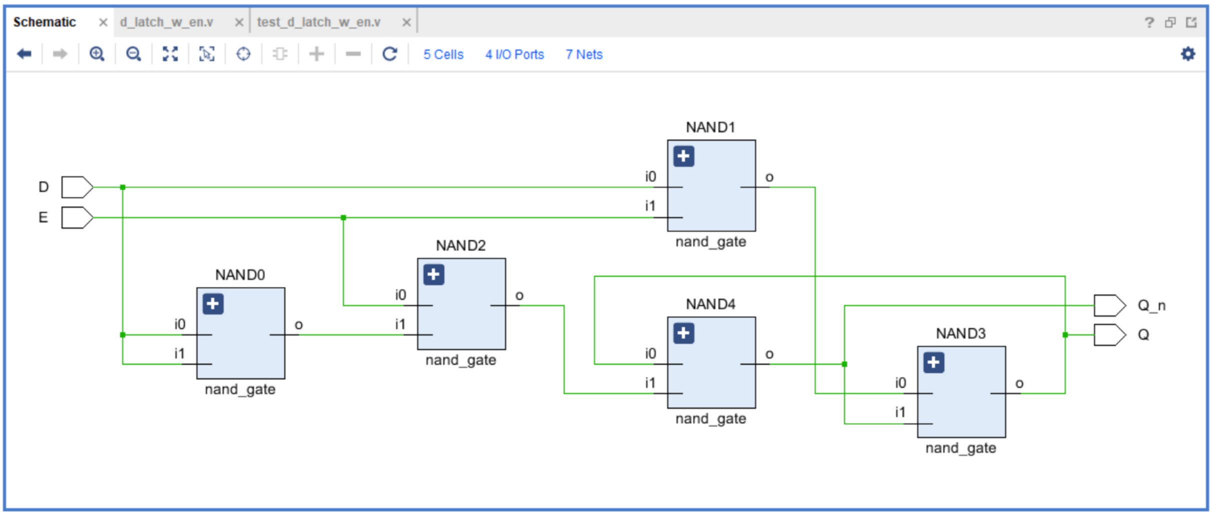
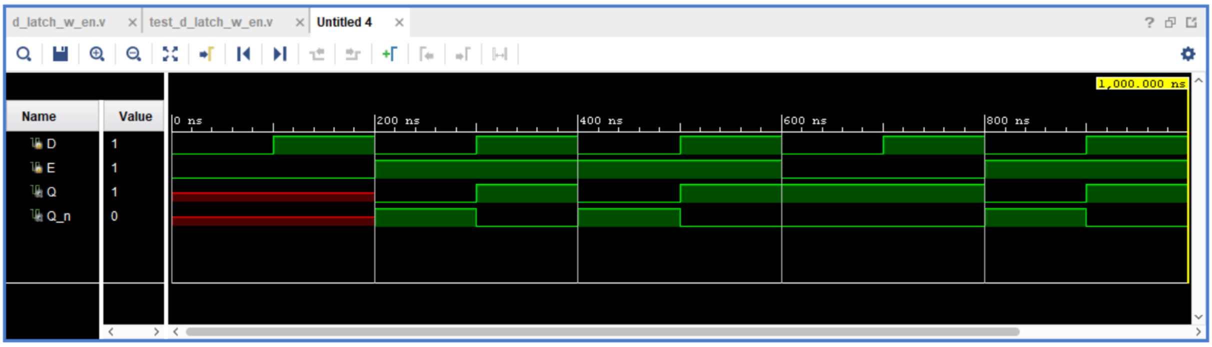


Figure 10: RTL Schematic of D Latch with Enable Input



Name	Value
D	1
E	1
Q	1
Q_n	0

Figure 11: Simulation of D Latch with Enable Input

Then with the technique called Master-Slave configuration, an edge triggered D flip flop can be structured by implementing a circuit which consists of two D latches whose Enable inputs are fed by the clock signal (or the inversion of the clock signal). The first Latch is called Master and the second one Slave. Sending inverted clock signal to the Enable input of Slave, while sending it directly to the Enable input of Master is the key point that makes this circuit negative edge triggered.

Here is the structure of the negative edge triggered D flip flop implemented from D Latches.

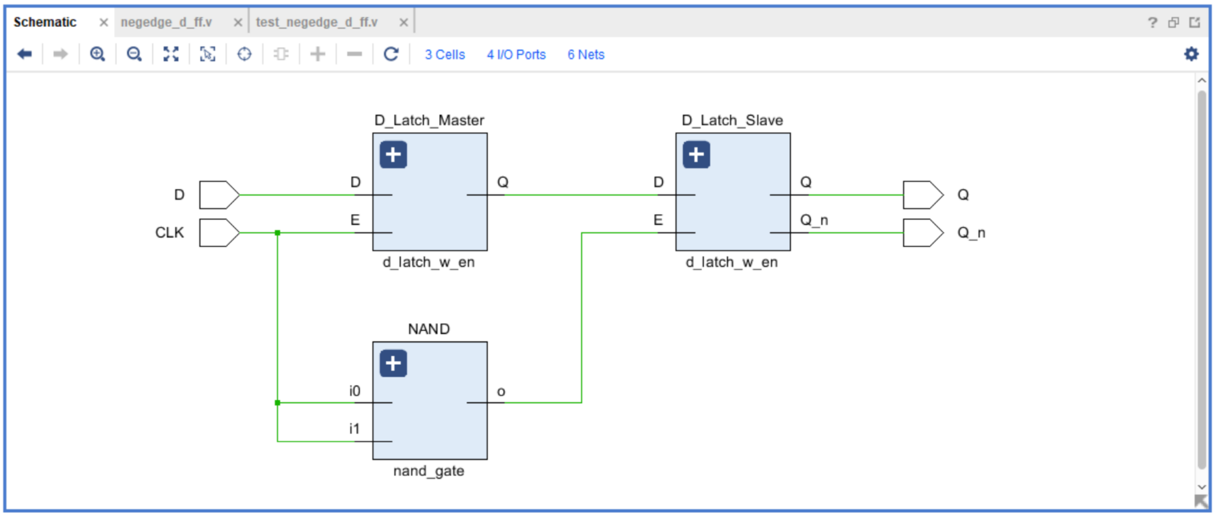


Figure 12: RTL Schematic of D Flip-Flop

And the simulation results we observed to verify that it works correctly is as follows.

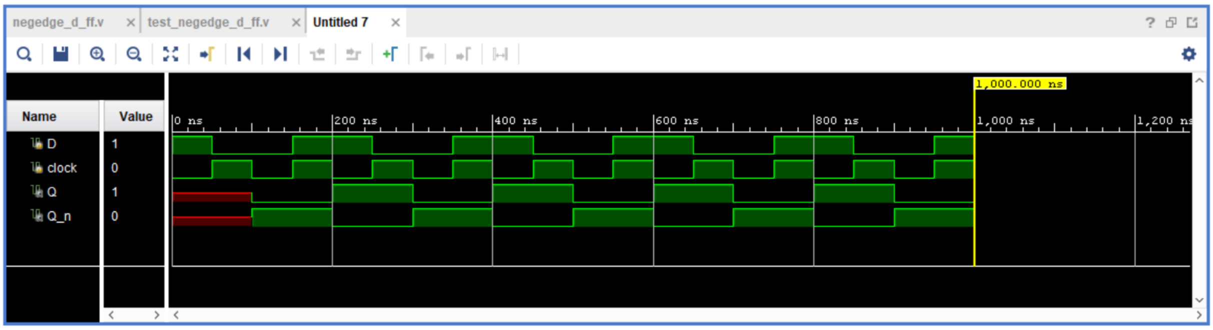


Figure 13: Simulation of D Flip-Flop

It is seen that the flip flop works as expected and clock is only effective at the falling edge.

When clock is high first latch (Master) is enabled, it stores the value of input D; however, the second latch cannot change its state. Then when the clock is low, Slave is enabled, it stores the output of Master and forwards it as output of D flip flop.

The circuit consists of two D flip-flops connected together. When the clock is high, the D input is stored in the first latch, but the second latch cannot change its state. When the clock is low, the first latch's output is stored in the second latch, but the first latch cannot change state. And Master cannot change its state during this time, until the next high clock signal.

In conclusion, this flip flop can only alter its state when there is a transition in the clock signal from a high state to a low state.

2.4 Part 4 - JK Flip-Flop (only NAND Gates)

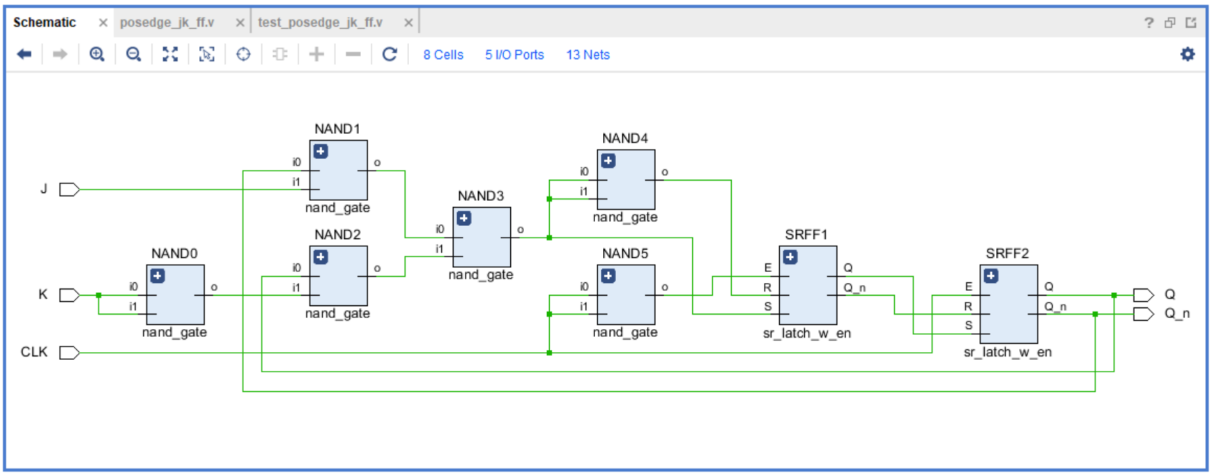


Figure 14: RTL Schematic of JK Flip-Flop

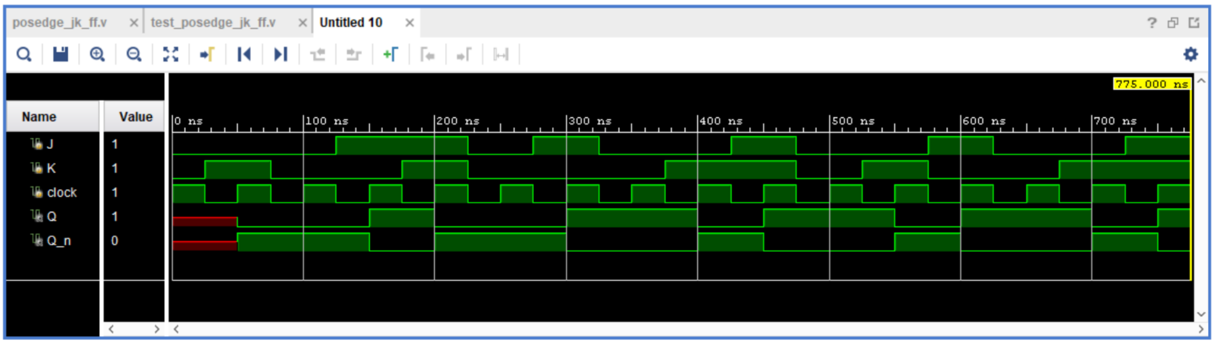


Figure 15: Simulation of JK Flip-Flop

2.5 Part 5 - Asynchronous Up Counter (JK Flip-Flop)

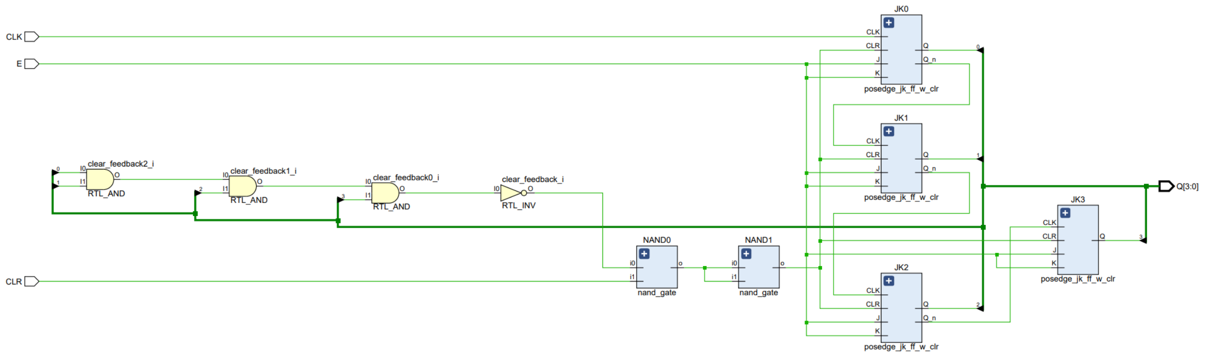
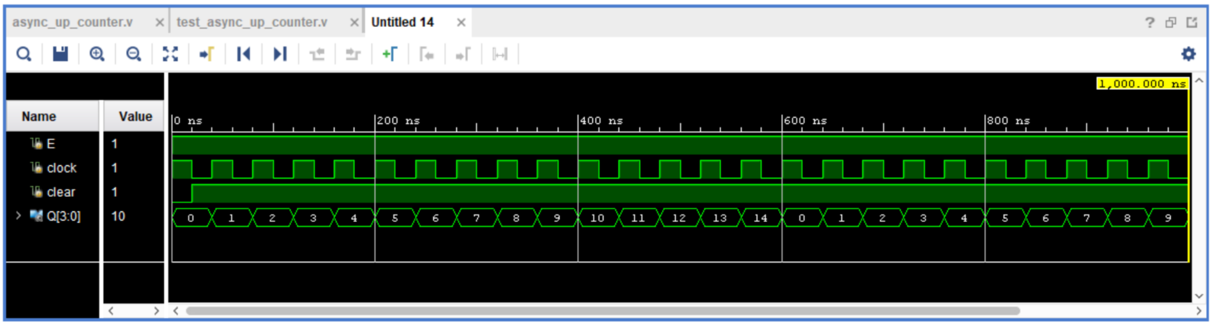


Figure 16: RTL Schematic of Asynchronous Up Counter



Name	Value
E	1
clock	1
clear	1
Q[3:0]	10

Figure 17: Simulation of Asynchronous Up Counter

Q[3](t)	Q[2](t)	Q[1](t)	Q[0](t)	Q[3](t+1)	Q[2](t+1)	Q[1](t+1)	Q[0](t+1)
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	0	0

Table 5: Truth table for an Asyncronous Up Counter

2.6 Part 6 - Synchronous Up Counter (JK Flip-Flop)

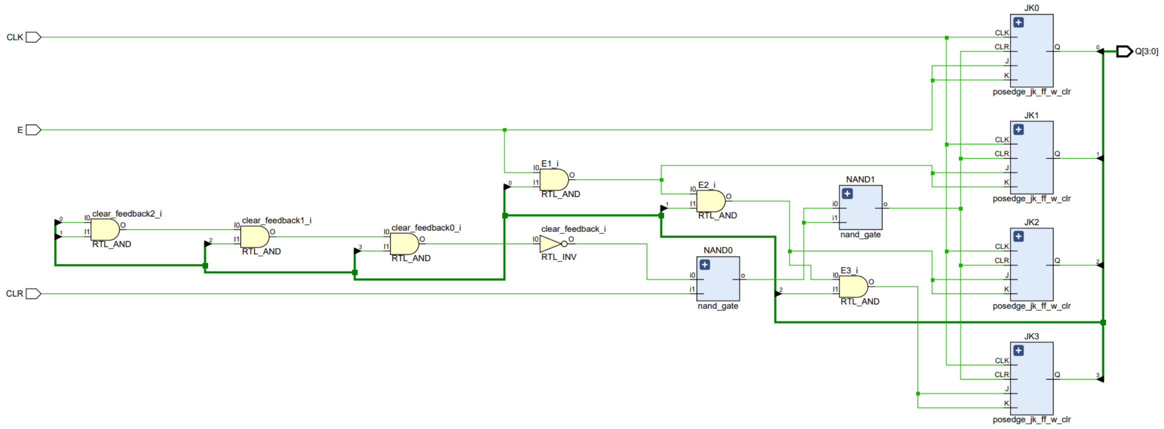
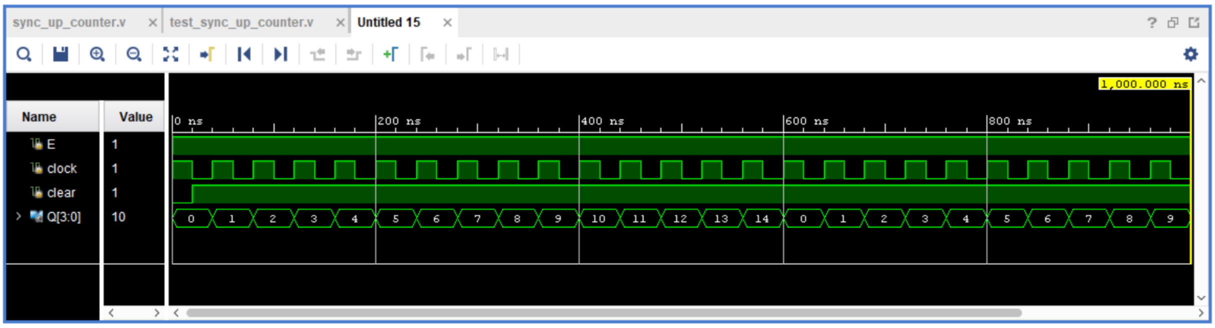


Figure 18: RTL Schematic of Synchronous Up Counter


sync_up_counter.v × test_sync_up_counter.v × Untitled 15 ×

Name	Value
E	1
clock	1
clear	1
Q[3:0]	10

Figure 19: Simulation of Synchronous Up Counter

Q[3](t)	Q[2](t)	Q[1](t)	Q[0](t)	Q[3](t+1)	Q[2](t+1)	Q[1](t+1)	Q[0](t+1)
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	0	0

Table 6: Truth table for an Syncronous Up Counter

2.7 Part 7 - Positive Edge Triggered Pulse Generator

REFERENCES

[1] - Mano, M. M., Ciletti, M. D. Digital design: with an introduction to the Verilog HDL, VHDL, and SystemVerilog. (No Title).

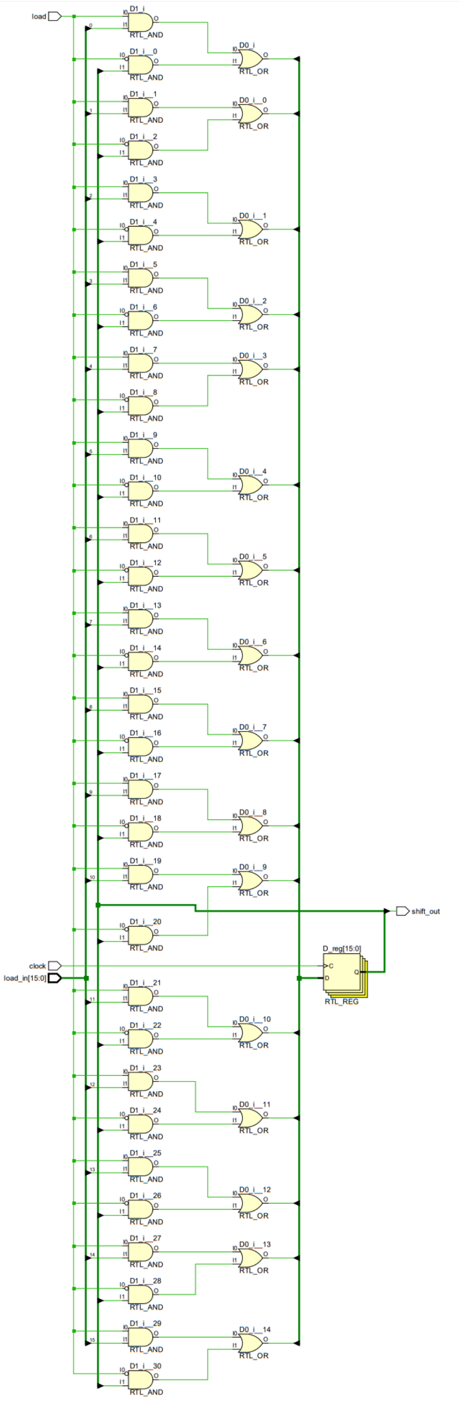
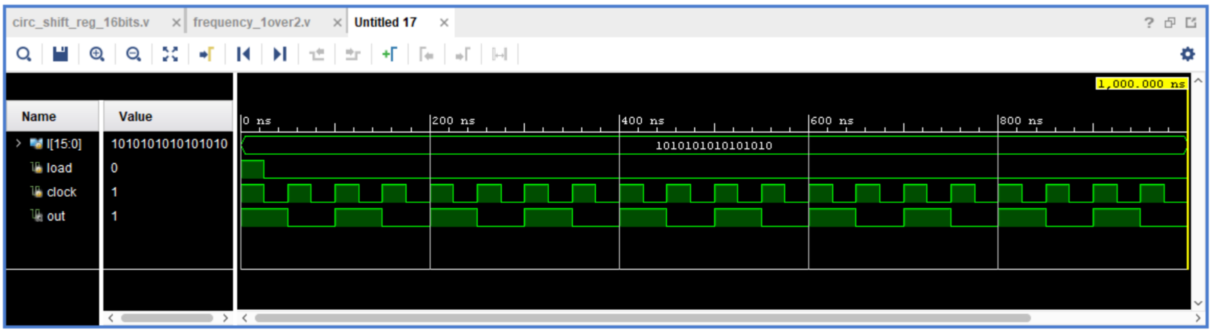
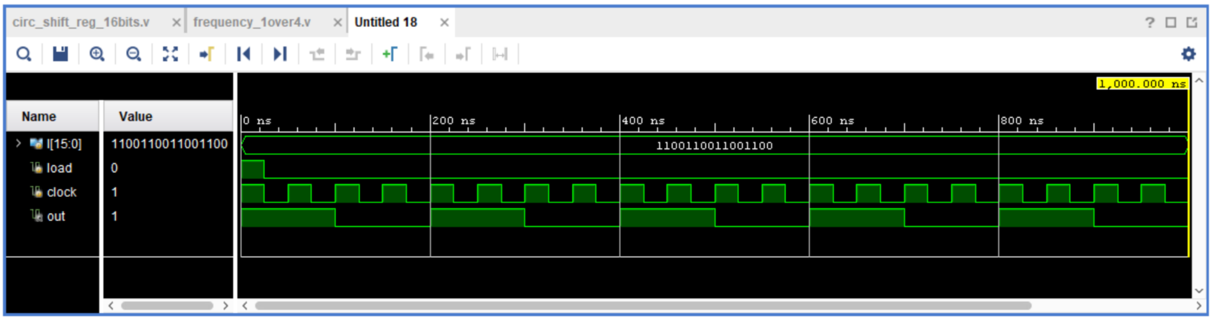


Figure 20: RTL Schematic of 16BIT Circular Shift Register with Load



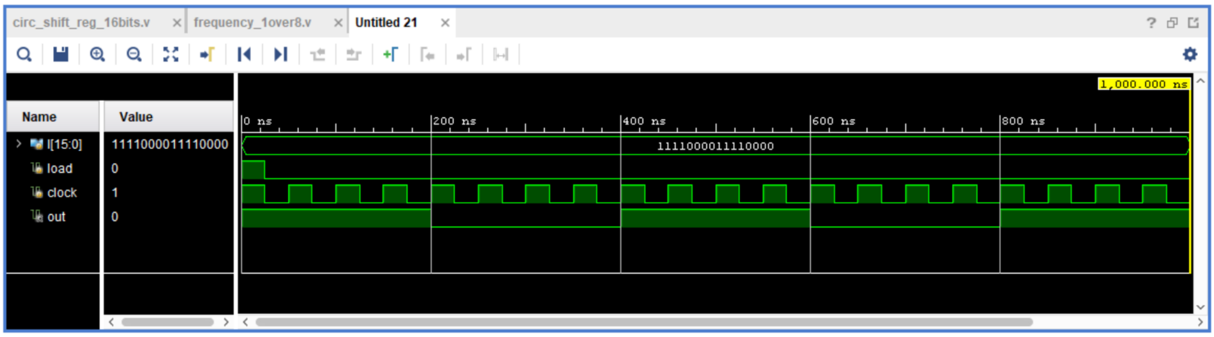
Name	Value
I[15:0]	1010101010101010
load	0
clock	1
out	1

Figure 21: Simulation of Shift Register with 1/2 Frequency of Clock Signal



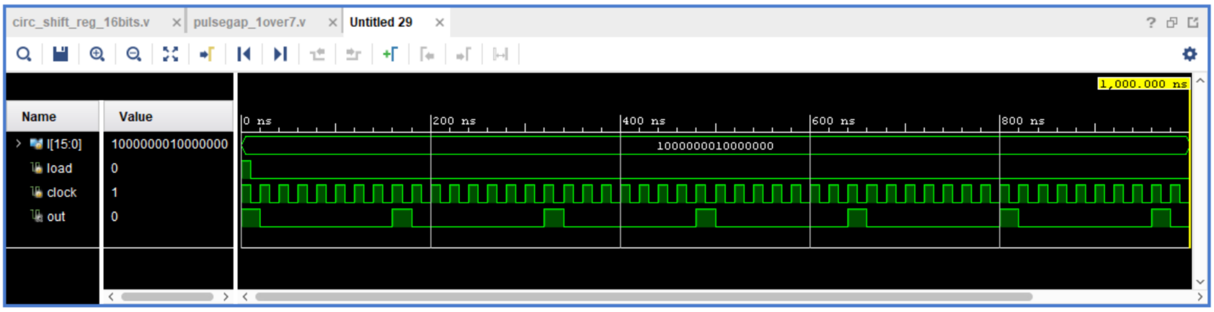
Name	Value
I[15:0]	1100110011001100
load	0
clock	1
out	1

Figure 22: Simulation of Shift Register with 1/4 Frequency of Clock Signal



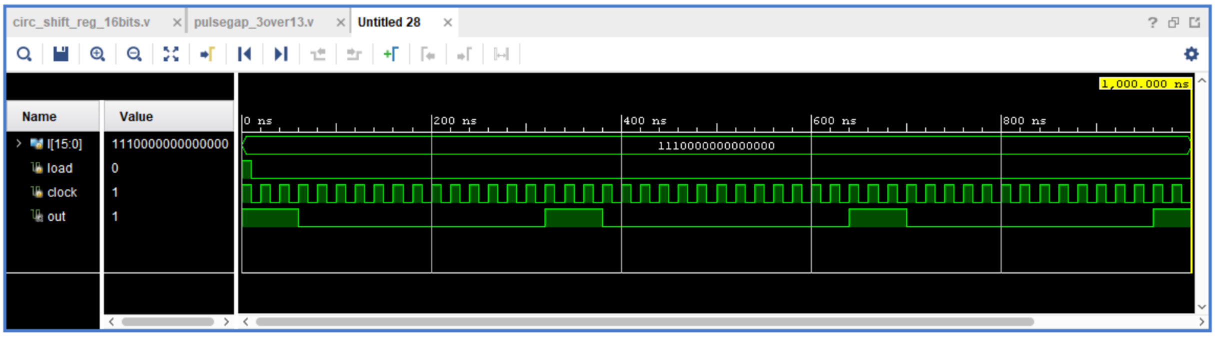
Name	Value
I[15:0]	1111000011110000
load	0
clock	1
out	0

Figure 23: Simulation of Shift Register with 1/8 Frequency of Clock Signal



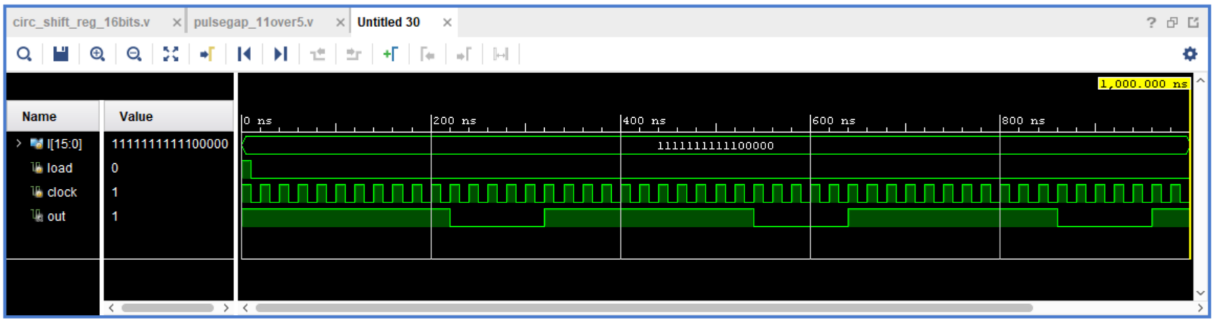
Name	Value
I[15:0]	1000000010000000
load	0
clock	1
out	0

Figure 24: Simulation of Shift Register with 1/7 Pulse-Gap Duration Rate



Name	Value
I[15:0]	1110000000000000
load	0
clock	1
out	1

Figure 25: Simulation of Shift Register with 3/13 Pulse-Gap Duration Rate



Name	Value
I[15:0]	1111111111100000
load	0
clock	1
out	1

Figure 26: Simulation of Shift Register with 11/5 Pulse-Gap Duration Rate