ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 242E

DIGITAL CIRCUITS LABORATORY HOMEWORK REPORT

HOMEWORK NO	:	2
HOMEWORK DATE	:	13.04.2022 - 27.04.2022
LAB SESSION	:	FRIDAY - 14.00
GROUP NO	:	G08

GROUP MEMBERS:

150200705 : HELİN ASLI & AKSOY

150180089 : HARUN & ÇİFCİ

SPRING 2022

Contents

1 INTRODUCTION [10 points]

In this experiment, we implemented and examined data storage elements: latches and flip-flops using Verilog without using always block in the Part 1 and 2. We simulated all parts separately with Vivado. For this experiment, we revised our previous knowledge from BLG231E-Digital Circuits course.

2 MATERIALS AND METHODS [40 points]

1. Flip-flops are useful for storing data. They are the smallest block for storing elements. They can store a bit. The data can be changed with different inputs.

2. 2.The difference between a flip flop and latch is the clock signal. Latch changes the output instantly when the input changes. Flip flop, on the other hand, changes the output according to the clock signal.

3. As the name indicates (Set Reset Latch), it sets the output to a certain value or resets it to 0. To build this latch, we need two 2-input NOR gates. First NOR gate is connected with S and output to find the inverse of output. After that, second NOR gate is connected with inverse of output and the input R. If SET is 1 and RESET is 0, the output is set to 1. If SET is 0 and RESET is 1, the output is resetted to 0. If both are 0, output does not change. Both inputs shouldn't be 1 because that case is ignored in SR Latch.

4. SR latch which does not have an Enable input.

S	R	Q	Q'
1	0	1	0
0	1	0	1
0	0	Q	Q
1	1	forbidden	forbidden

Figure 1: S-R Latch without enable truth table

5. SR latch which has an Enable input

E	S	R	Q	Q'
0	0	0	latch	latch
0	0	1	latch	latch
0	1	0	latch	latch
0	1	1	latch	latch
1	0	0	latch	latch
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

Figure 2: S-R Latch with enable truth table

6. The truth table of a JK flip flop

J	K	CLK	Q(t+1)	QN(t+1)
x	x	0	Q(t)	QN(t)
x	x	1	Q(t)	QN(t)
0	0	↓	Q(t)	QN(t)
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	QN(t)	Q(t)

Figure 3: The truth table of a JK flip flop

7. The truth table of a D flip flop

E	D	Q	Q'
0	0	LATCH	LATCH
0	1	LATCH	LATCH
1	0	0	1
1	1	1	0

Figure 4: the truth table of a D flip flop

3 RESULTS [15 points]

3.1 PART 1

We have implemented an SR latch module with S and R inputs and with Q and Q neg outputs without an Enable input only using 2-input NOR gates. Using the truth table we have constructed in the Preliminary section, we have written the characteristic equation of the latch as Q(t + 1) = f(S; R; Q(t)). We have constructed a K-map in order to find the characteristic equation of SR latch without enable input. From this K-map we can see that S and Q(t) * R' are our prime implicants. And the equation is Q(t+1) = S + Q(t) * R' . And for the disallowed inputs, which are the inputs S = 1, R = 1, it is not certain that how will our circuit behave.

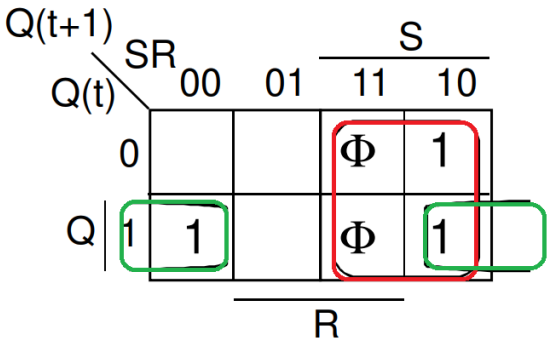

Figure 5: Equation

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

Figure 6: Equation of SR latch

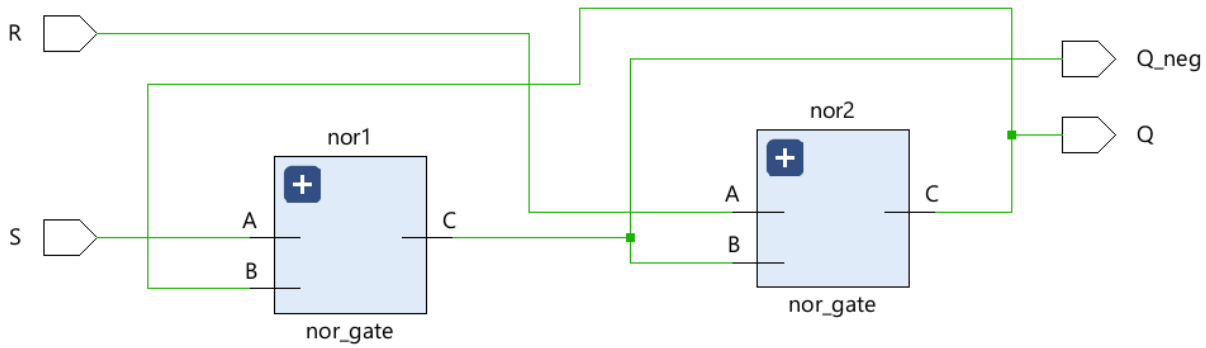

Figure 7: Schema of circuit designed for Part 1

Q(t)	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	*forbidden*
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	*forbidden*

Figure 8: Truth Table of circuit designed for Part 1

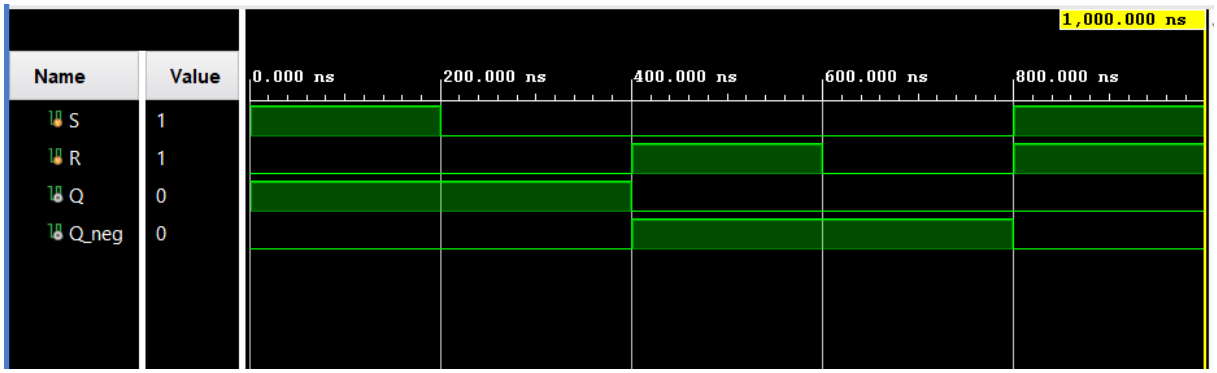

Figure 9: Simulation of designed SR-Latch

3.2 PART 2

We have implemented an SR latch module with S, R and Enable inputs and with Q and Q neg outputs by using only NAND gates. Using the truth table we have constructed in the Preliminary section, we have written the characteristic function of the latch as Q(t + 1) = f(S; R; E; Q(t)). From the truth table it can be seen that when Enable input is zero, Q does not change, and for the Enable = 1, our S and R values are altered. Characteristic equation is: Q(t+1) = (S * E) + Q(t) * (R * E)' And for the disallowed inputs, which are the inputs S = 1 and R = 1, both Q' and Q outputs have been 1 and in this state, the value in the latch is uncertain.

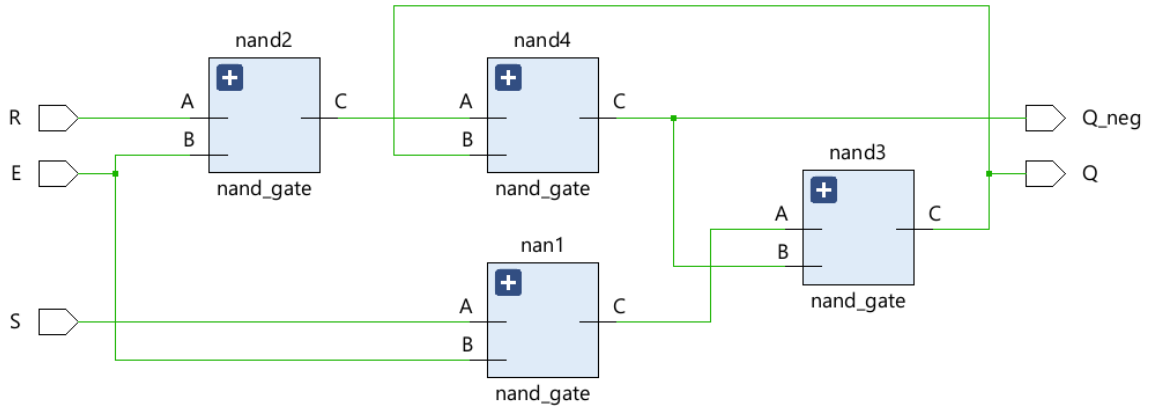

Figure 10: Schema of circuit designed for Part 2

E	S	R	Q(t+1)
0	X	X	Q(t)
1	1	0	1
1	0	1	0
1	0	0	Q(t)
1	1	1	*forbidden*

Figure 11: Truth Table of circuit designed for Part 2

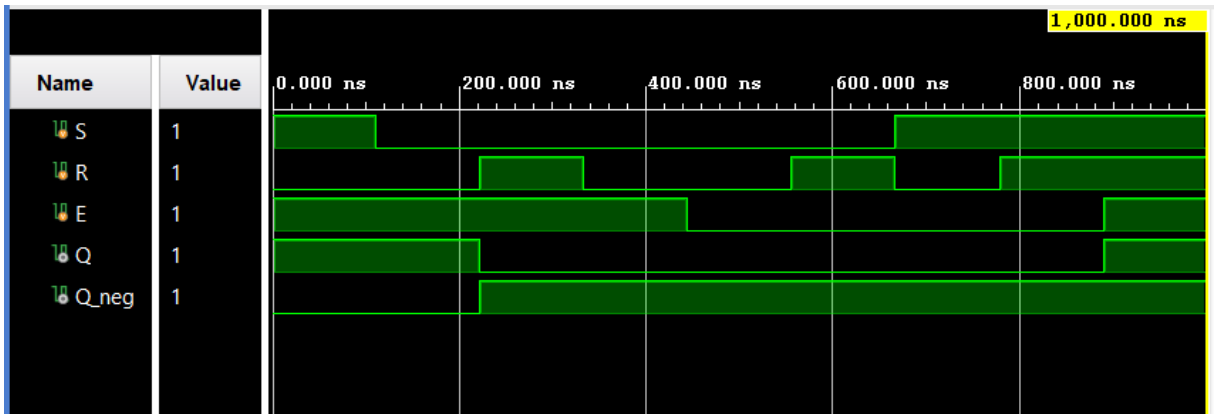

Figure 12: Simulation of designed SR-Latch

3.3 PART 3

We have implemented a positive edge triggered D flip-flop module from D-Latches with D input and for Q and Q neg outputs using D latches with Enable input. We have implemented the D latches with Enable using only 2-input NAND gates as a separate module. D-Flip flop is only effective at rising edges as can be seen in simulation figure:

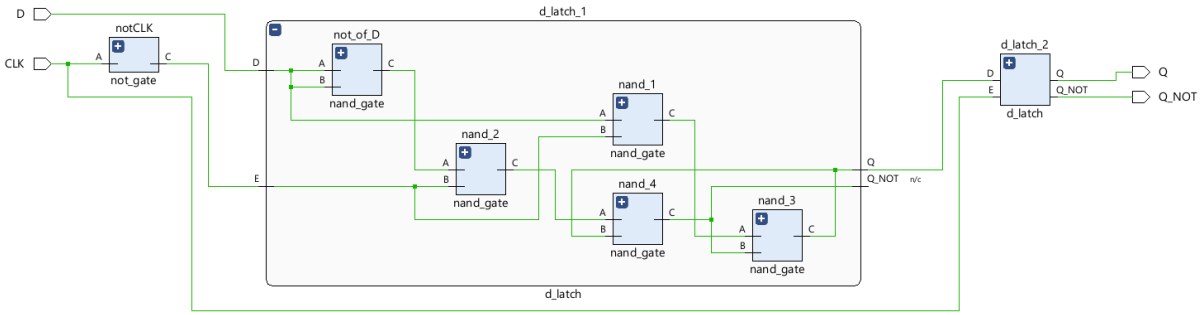

Figure 13: Schema of circuit designed for Part 3

E	D	Q	Q'
0	0	LATCH	LATCH
0	1	LATCH	LATCH
1	0	0	1
1	1	1	0

Figure 14: Truth Table of circuit designed for Part 3

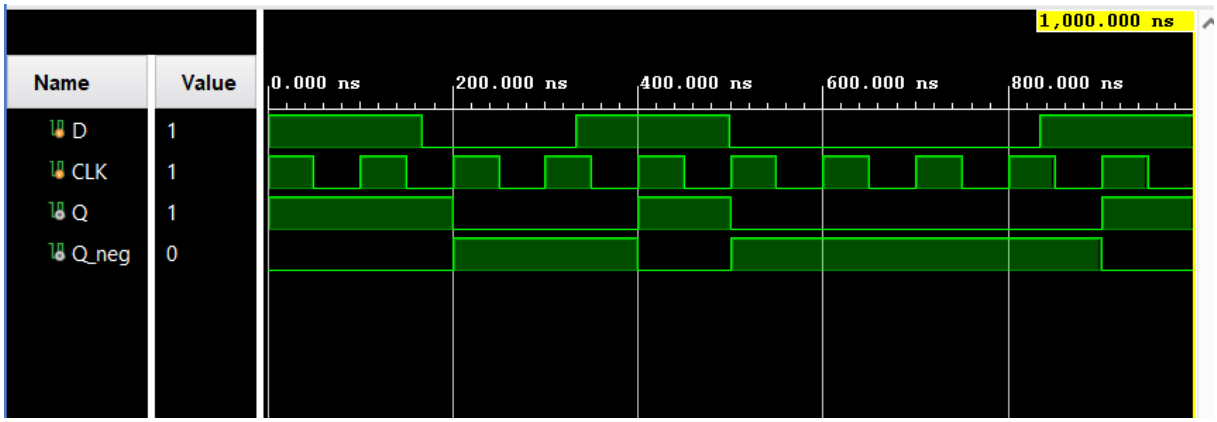

Figure 15: Simulation of designed D-Flip-flop

3.4 PART 4

We have implemented an edge triggered JK flip-flop module from SR flip-flop with J, K and Clock inputs and for Q and Q neg outputs. We have implemented the JK flip flop by using only 2-input NAND gates as a separate module.

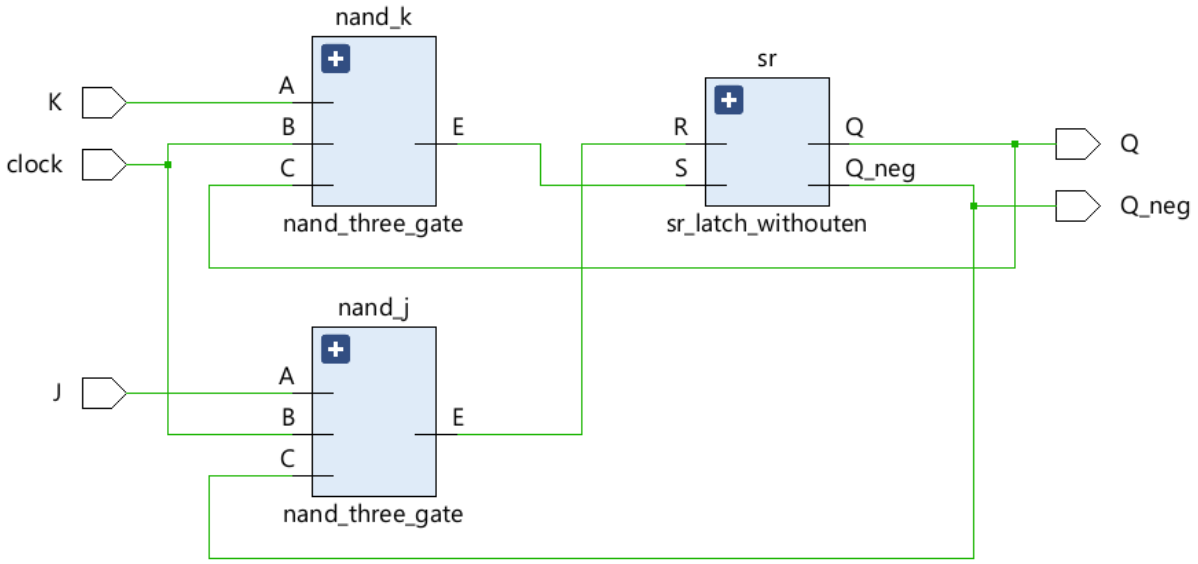

Figure 16: Schema of circuit designed for Part 4

J	K	CLK	Q(t+1)	QN(t+1)
x	x	0	Q(t)	QN(t)
x	x	1	Q(t)	QN(t)
0	0	↓	Q(t)	QN(t)
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	QN(t)	Q(t)

Figure 17: Truth Table of circuit designed for Part 4

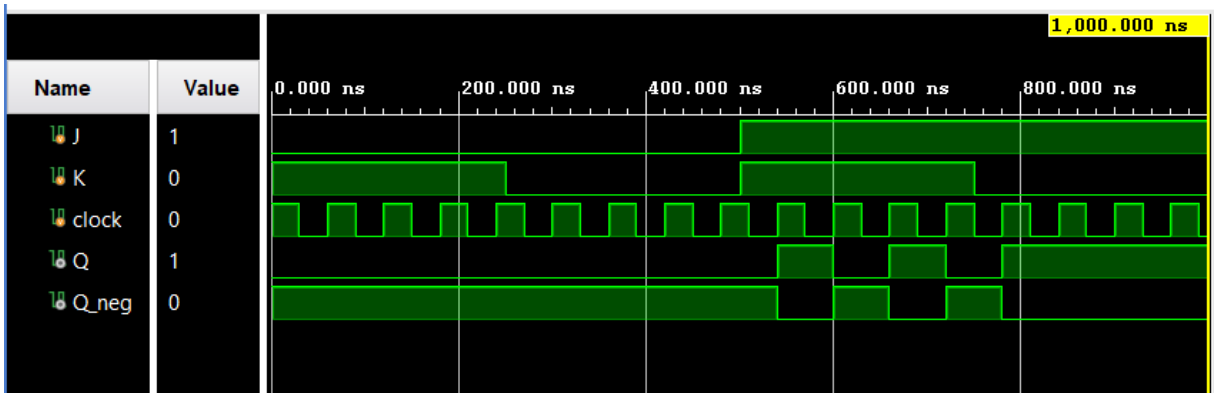

Figure 18: Simulation of designed JK Flip-flop

3.5 PART 5

We have implemented a 4-bit asynchronous up counter using JK flip flops. For each flip flop, input J, K and clock and get values from output Q. We have implemented the counter count between 0-14 in decimal. Also it resets the counter when it reaches 15 in decimal.

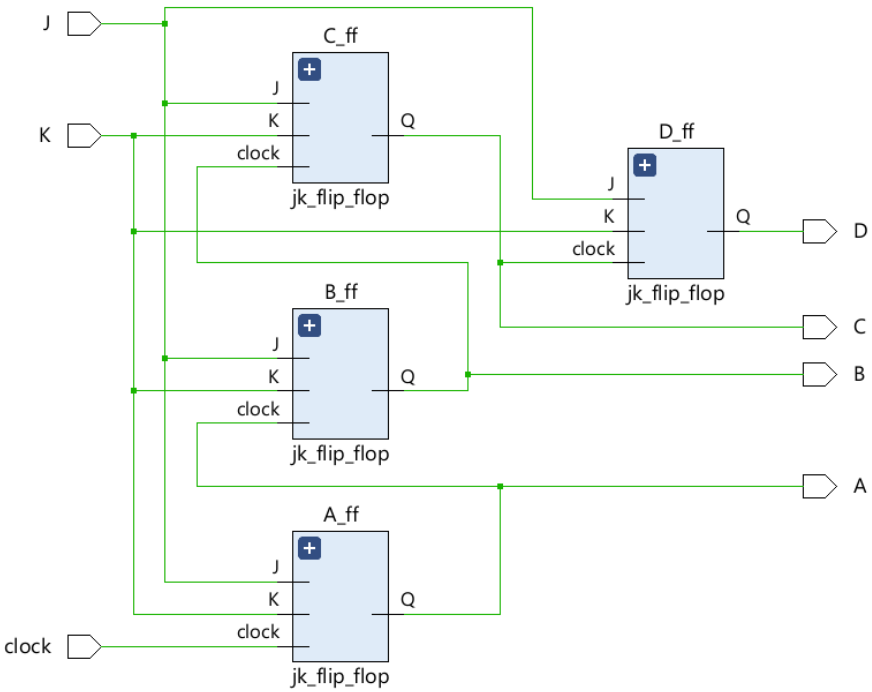

Figure 19: Schema of circuit designed for Part 5

Clock	rst	bin_out(3)	bin_out(2)	bin_out(1)	bin_out(0)
X	0	0	0	0	0
↑	1	0	0	0	0
↑	1	0	0	0	1
↑	1	0	0	1	0
↑	1	0	0	1	1
↑	1	0	1	0	0
↑	1	0	1	0	1
↑	1	0	1	1	0
↑	1	0	1	1	1
↑	1	1	0	0	0
↑	1	1	0	0	1
↑	1	1	0	1	0
↑	1	1	0	1	1
↑	1	1	1	0	0
↑	1	1	1	0	1
↑	1	1	1	1	0
↑	1	1	1	1	1

Figure 20: Truth Table of circuit designed for Part 5

3.6 PART 6

In this part, we have implemented a 4-bit synchronous up counter using JK flip flops. The counter performs the same operation as the counter we have created in Part 5.

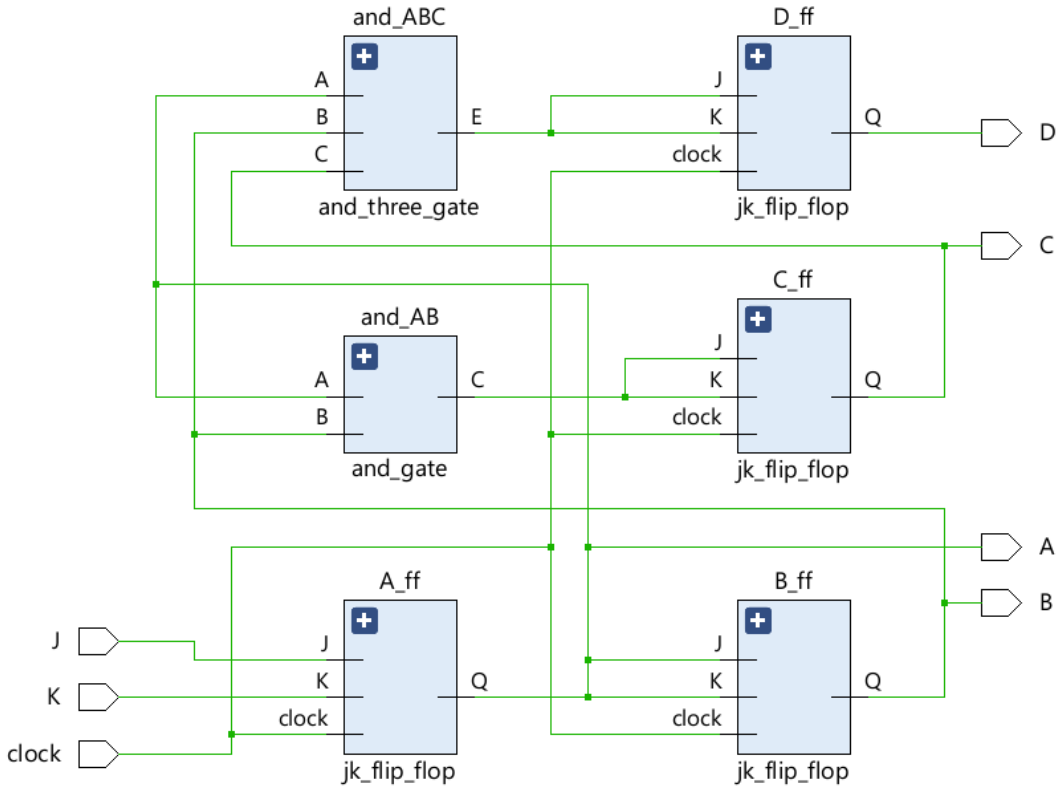

Figure 21: Schema of circuit designed for Part 6

Count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

Figure 22: Truth Table of circuit designed for Part 6

3.7 PART 7

In this part, we have implemented a positive edge triggered pulse generator using a circular shift register. The circuit takes 8-bit input for the loaded value, 1-bit input for the clock signal, 1-bit input for the load flag and give 1-bit output. Basically, when Load=0 (that is, shift=1), with the clock signal, circular shift operation is done; when Load=1, with the clock signal, a 8-bit input value is loaded. We have designed our circuit in a way that the output of this circuit is the most significant bit (MSB) of the loaded value.

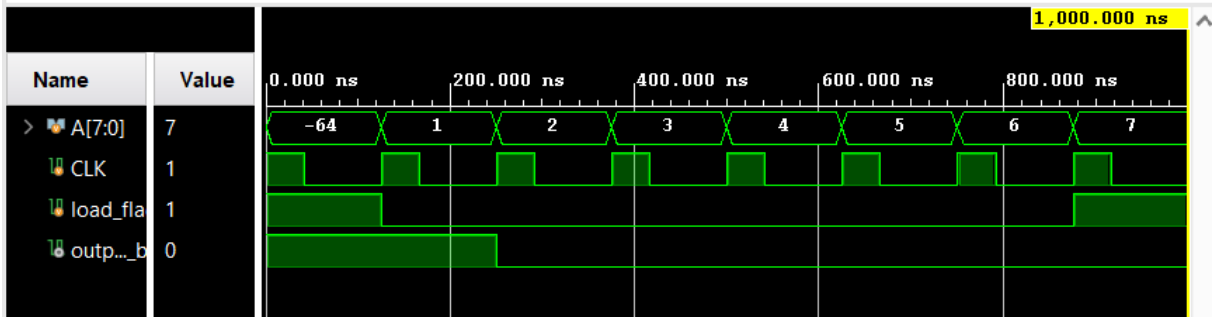

Figure 23: Simulation result with the 1/2 frequency of clock signal and 1/3 pulse-gap duration rate

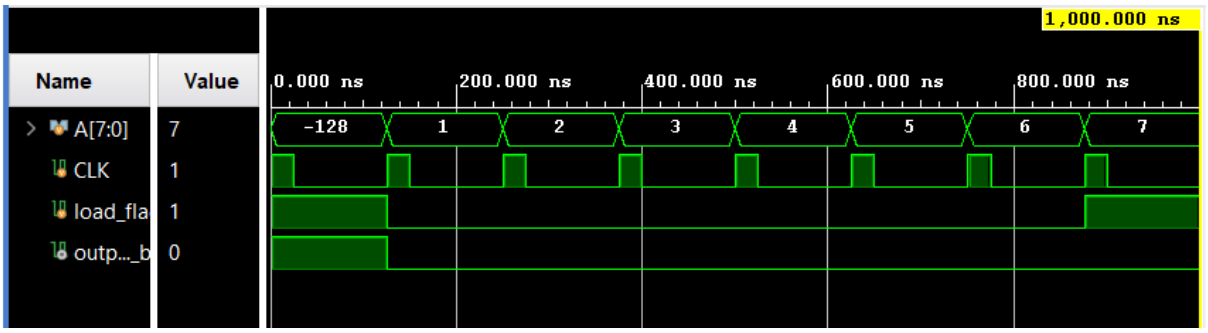

Figure 24: Simulation result with the 1/4 frequency of clock signal and 1/7 pulse-gap duration rate

- P.S. Since 4/13 Pulse Gap cannot be provided with 8 bits, we rounded it to the nearest value 4/12.

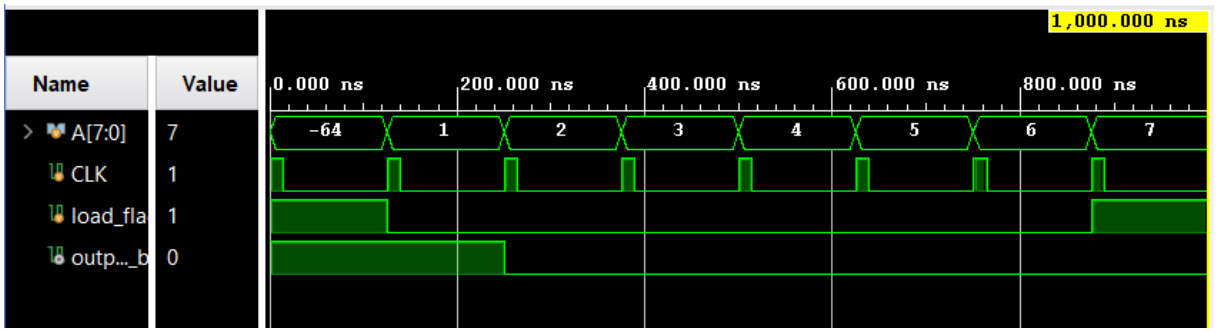

Figure 25: Simulation result with the 1/8 frequency of clock signal and 1/3 pulse-gap duration rate

4 DISCUSSION [25 points]

In the preliminary, firstly we have introduced what flip-flop is and why they are useful. Secondly, we have tried to explain what are the differences between latch and flip-flop. Then, we have observed how an SR-Latch works and what are the functionalities of the input variables are. Then we have constructed truth table of SR-Latch without and with Enable input. After that, we have constructed truth tables of JK and D Flip-Flop.

For the experiment section, In the first and second parts, we have implemented an SR-Latch module. First one was designed with only NOR gates and doesn't have an enable input. Second one was designed with only NAND gates and has enable input. Using the truth tables we have constructed at preliminary section we have written the characteristic functions of latches we designed. Then we have explained how we found the equation and how do latches behave. We discussed the differences between SR Latch with and without enable input.

In the third part, we have implemented a positive edge triggered D-Flip-Flop module from D-Latches. We implemented the D-Latch with enable inputs and using only NAND gates.

In the forth part, we have implemented a positive edge triggered JK-Flip-Flop module from SR Flip-Flop and with using only NAND gates.

In the fifth and sixth part, we have implemented a 4-bit asynchronous up counter and a synchronous up counter with using JK Flip-Flops. Then we added truth tables of designed up counters.

In the last part, we have implemented a positive edge triggered pulse generator using a circular shift register. The takes 8-bit input for the loaded value, 1-bit input for the clock signal, 1-bit input for the load flag and give 1-bit output. Basically, when Load=0 (that is, shift=1), with the clock signal, circular shift operation is done; when Load=1, with the clock signal, a 8-bit input value is loaded.

We have designed our circuit in a way that the output of this circuit is the most significant bit (MSB) of the loaded value.

We have simulated our design with given frequencies of clock signals and pulse-gap duration rate.

5 CONCLUSION [10 points]

In this task we have worked with utilities of modular design of Vivado. We have designed and simulated latches, flip-flops, counters and positive edge triggered pulse generator. During the design and simulation process we have faced with some problems. Especially with testing of JK Flip-Flop and counters. We have overcame with those issues by searching and asking to teaching assistants. Eventually we have became familiar with behaviours of latches and flip flops and we learned what can we produce with usage of flip-flops. In conclusion, we suffered, we learned, we practised.