Author: Sławomir Batruch
Date: 30.04.2021

I declare that this piece of work that is a basis for the grading of edc1 laboratory 3 task was performed on my own without any help (indirect or direct) from other students.

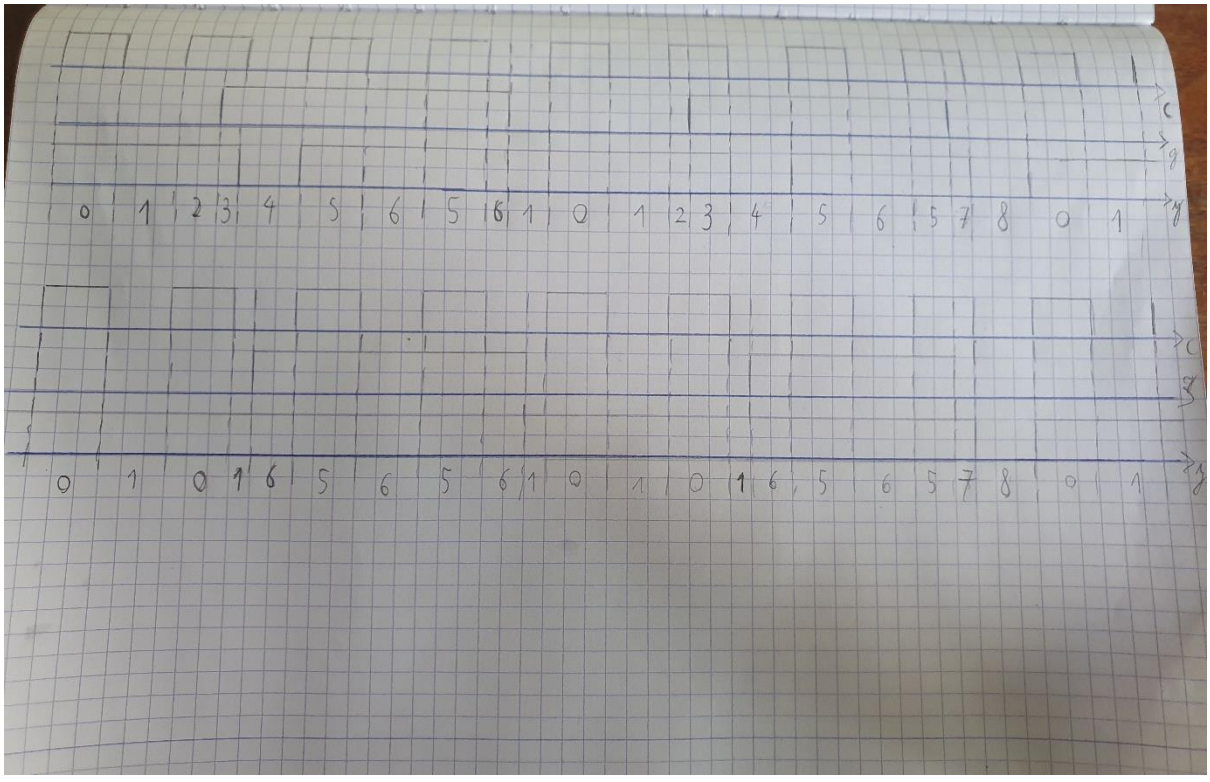
Sławomir Batruch; 303827

EDC1 – Lab3 report

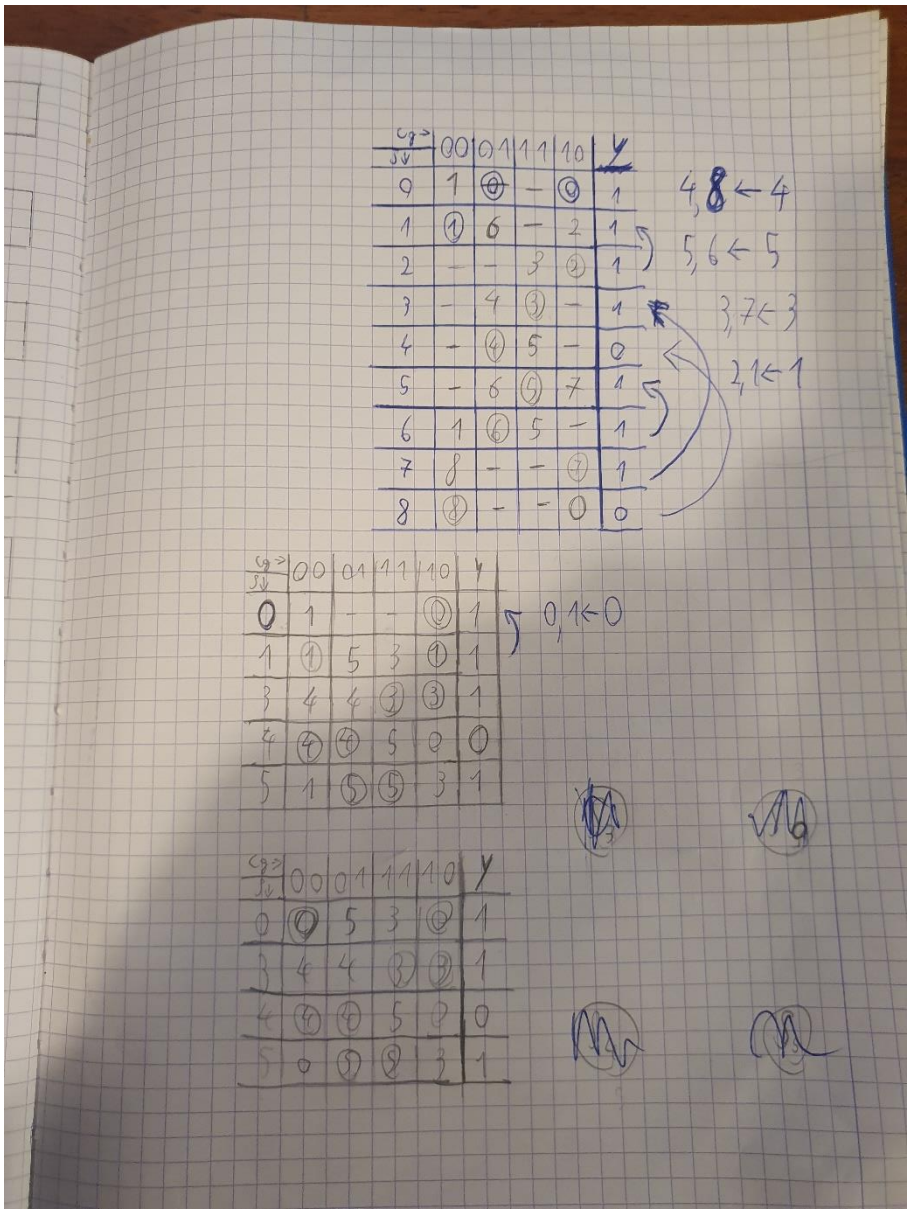
On this laboratory the task was to design a device that would copy a clock pulse in some specified matter. The task given individually was as follows:

Design the sequential asynchronous detector of any change of the gate signal. The device must copy a full, uncut negative (high-low-high) impulse of the clock to the output if the slope of the gate is encountered at a high state of the clock. Otherwise, if the slope of the gate is encountered at a low state of the clock the change of gate is ignored and nothing is outputted. Gate signal is relatively very slowly changing, comparing to the clock. This means, that both low and high states of the gate last significantly longer than the period of the clock.

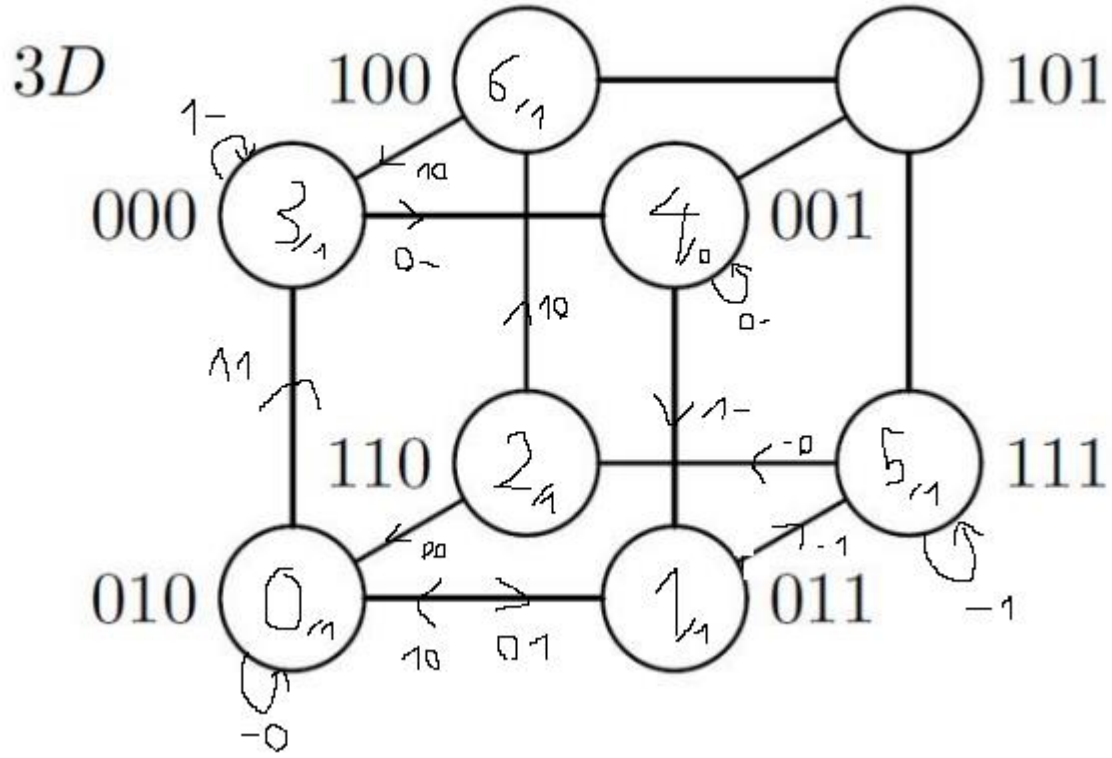
First, I started by drawing a timing diagram (waveform) that would fulfil the task description. Then, I numbered the states present on the waveform. Any compatible states would get later “connected”:



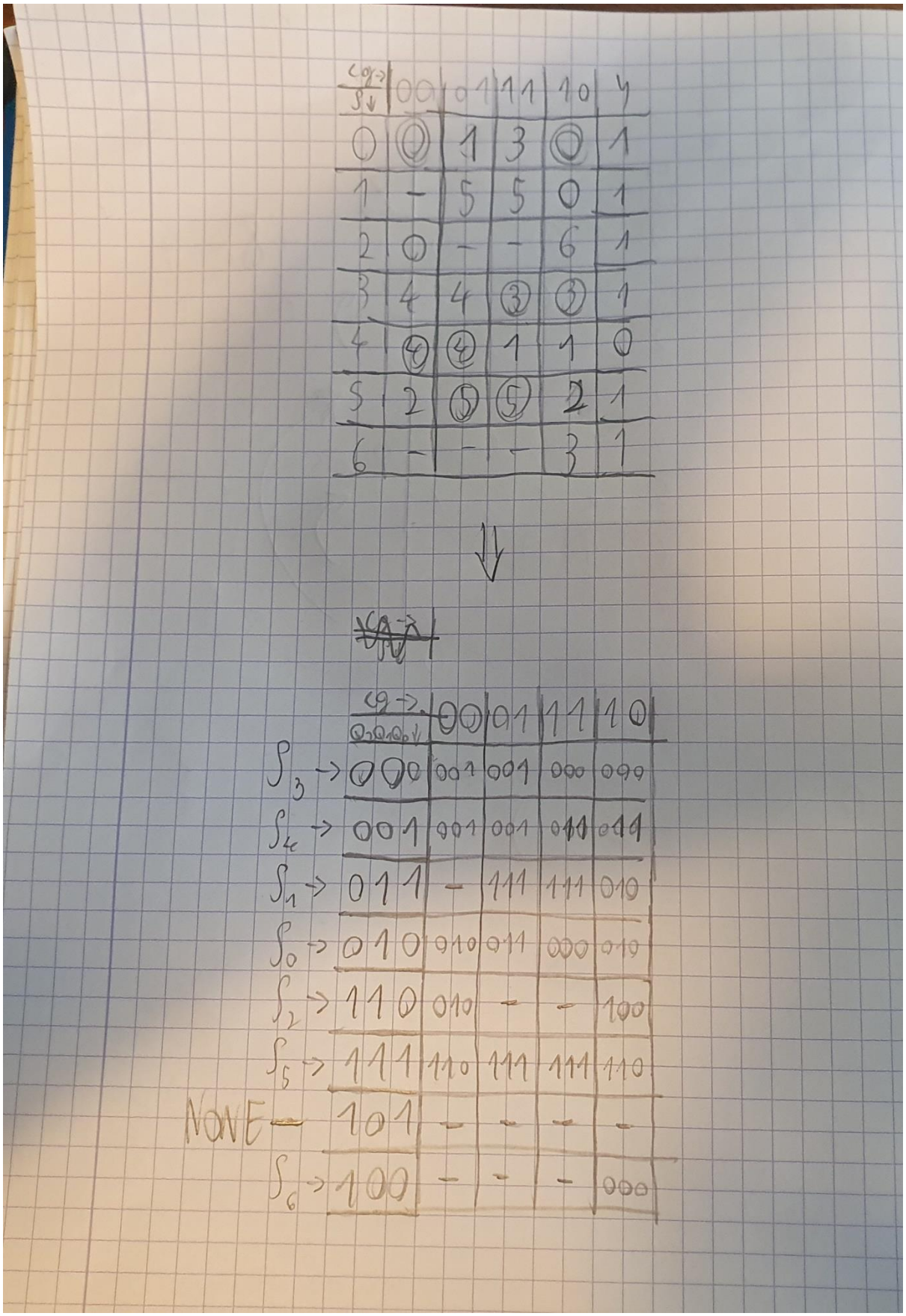
After that on the basis of these waveforms I did a state transition table and reduced the compatible states:



And after that, I needed to fix the races that I saw when tailoring the hypercube. I saw the unallowed transitions and fixed them by adding states and using them as a alternate route to some states:



From that I did another state transition table with the fixed races, and then converted it into binary states:



Q_2Q_1 → / S ↓	00	01	11	10	y
0	(0)	1	3	(0)	1
1	-	5	5	0	1
2	(0)	-	-	6	1
3	4	4	(3)	(3)	1
4	(4)	(4)	1	1	0
5	2	(5)	(5)	2	1
6	-	-	-	3	1

⇓

	Q_2Q_1 → / Q_3Q_0 ↓	00	01	11	10
S_3 →	0 0 0	001	001	000	000
S_4 →	0 0 1	001	001	011	011
S_1 →	0 1 1	-	111	111	010
S_0 →	0 1 0	010	011	000	010
S_2 →	1 1 0	010	-	-	100
S_5 →	1 1 1	110	111	111	110
NONE →	1 0 1	-	-	-	-
S_6 →	1 0 0	-	-	-	000

From that all that was left from theoretical design was to do transcoding for SR flip flops. This screenshot shows that step:

cg / Q2Q1Q0	00	01	11	10
000	1	1		
001				
011				
010		1		
110				
111				
101				
100				

S_0

S0 = c'g'Q1' + c'g

cg / Q2Q1Q0	00	01	11	10
000				
001				
011				
010				
110				
111				
101				
100				

R_0

R0 = g'Q1

Q	$\rightarrow$	Q'	S	R
0	$\rightarrow$	0	0	–
0	$\rightarrow$	1	1	0
1	$\rightarrow$	0	0	1
1	$\rightarrow$	1	–	0

cg / Q2Q1Q0	00	01	11	10
000				
001				
011	-			
010				
110				
111				
101				
100				

S_1

S1 = cq0

cg / Q2Q1Q0	00	01	11	10
000				
001				
011				
010				
110				
111				
101				
100				

R_1

R1 = cQ2Q1Q0' + cgQ1Q0'

$$y = (Q_2' Q_1' Q_0)'$$

cg / Q2Q1Q0	00	01	11	10
000				
001				
011				
010				
110				
111				
101				
100	-		-	

S_2

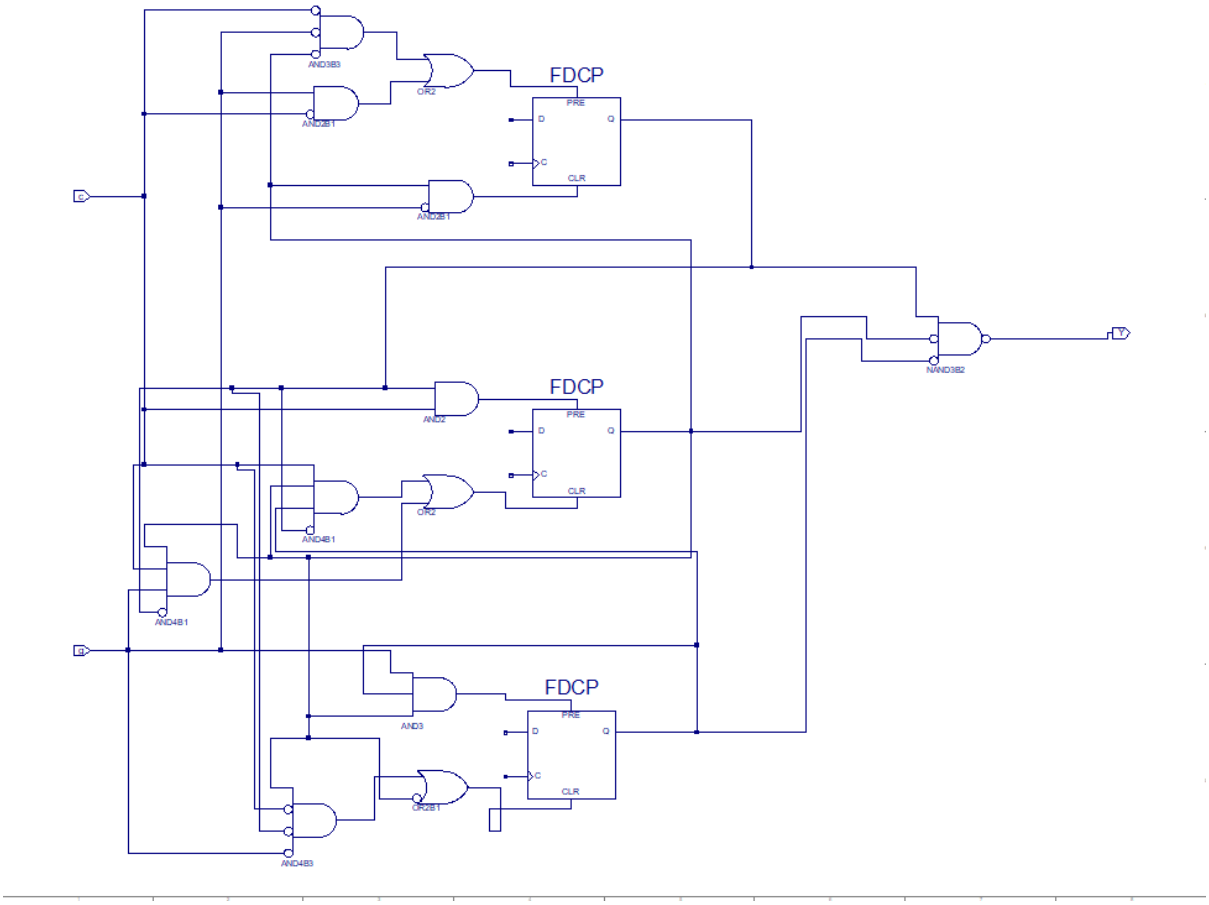
S2 = gQ1Q2

cg / Q2Q1Q0	00	01	11	10
000				
001				
011				
010				
110		-		
111				
101				
100		-		

R_2

R2 = Q1' + c'g'Q1Q0'

From that all I needed to do was do a schematic in xilinix:



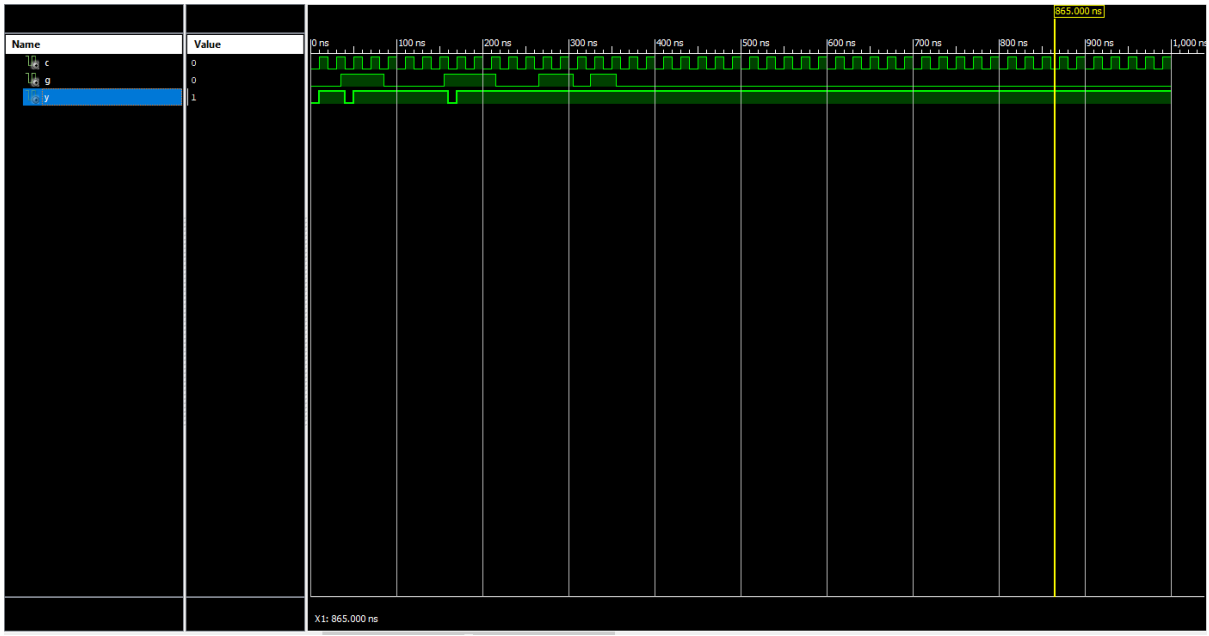
And then do a VHDL testbench that would be the same as the waveform sketch I did at the beginning. Here is a code snippet of the vhdl test bench I did write:

```
SIGNAL Y      :     STD_LOGIC;
SIGNAL c      :     STD_LOGIC:= '0';
SIGNAL g      :     STD_LOGIC:= '0';
```

[…]

```
-- *** Test Bench - User Defined Section ***
      c <= not c after 10 ns;
   tb : PROCESS
   BEGIN
           wait for 35 ns;
           g <= '1';
           wait for 50 ns;
           g <= '0';
           wait for 70 ns;
           g <= '1';
           wait for 60 ns;
           g<= '0';
           wait for 50 ns;
           g <= '1';
           wait for 40 ns;
           g <= '0';
           wait for 20 ns;
           g <= '1';
           wait for 30 ns;
           g <= '0';
      WAIT; -- will wait forever
   END PROCESS;
```

This VHDL testbench resulted in this simulation output:

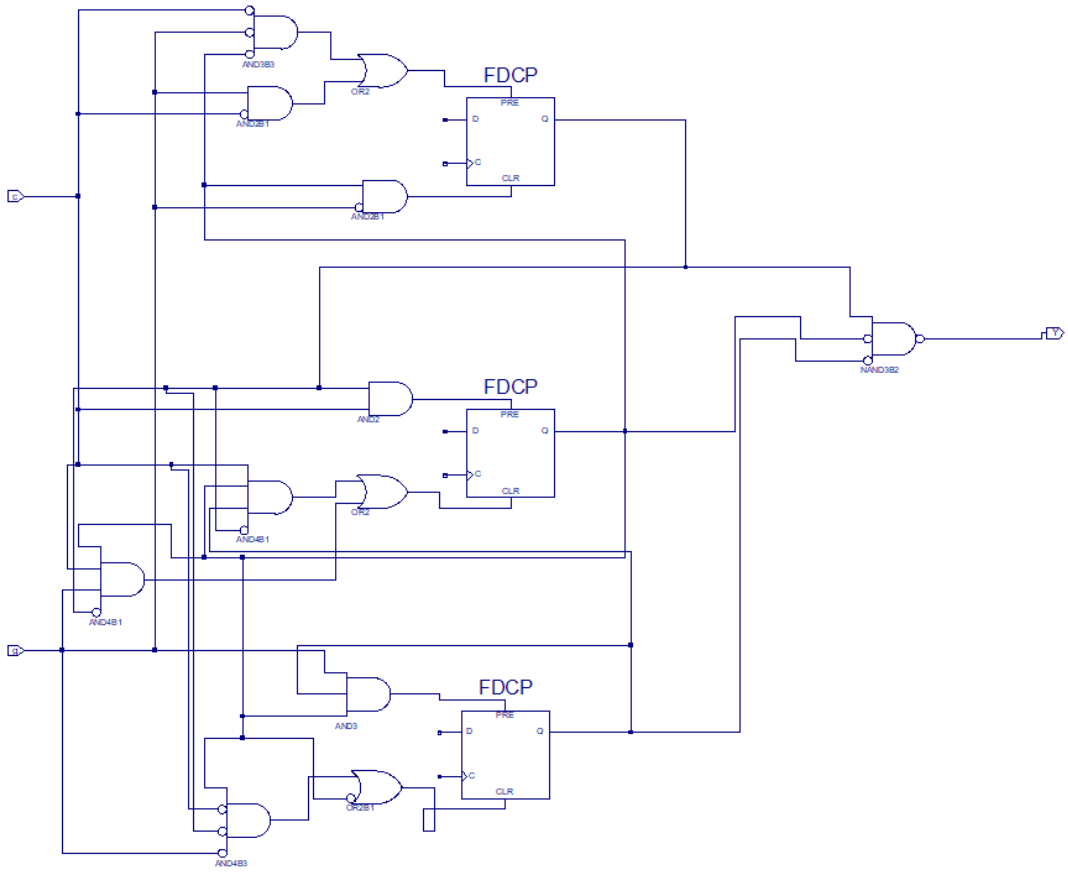


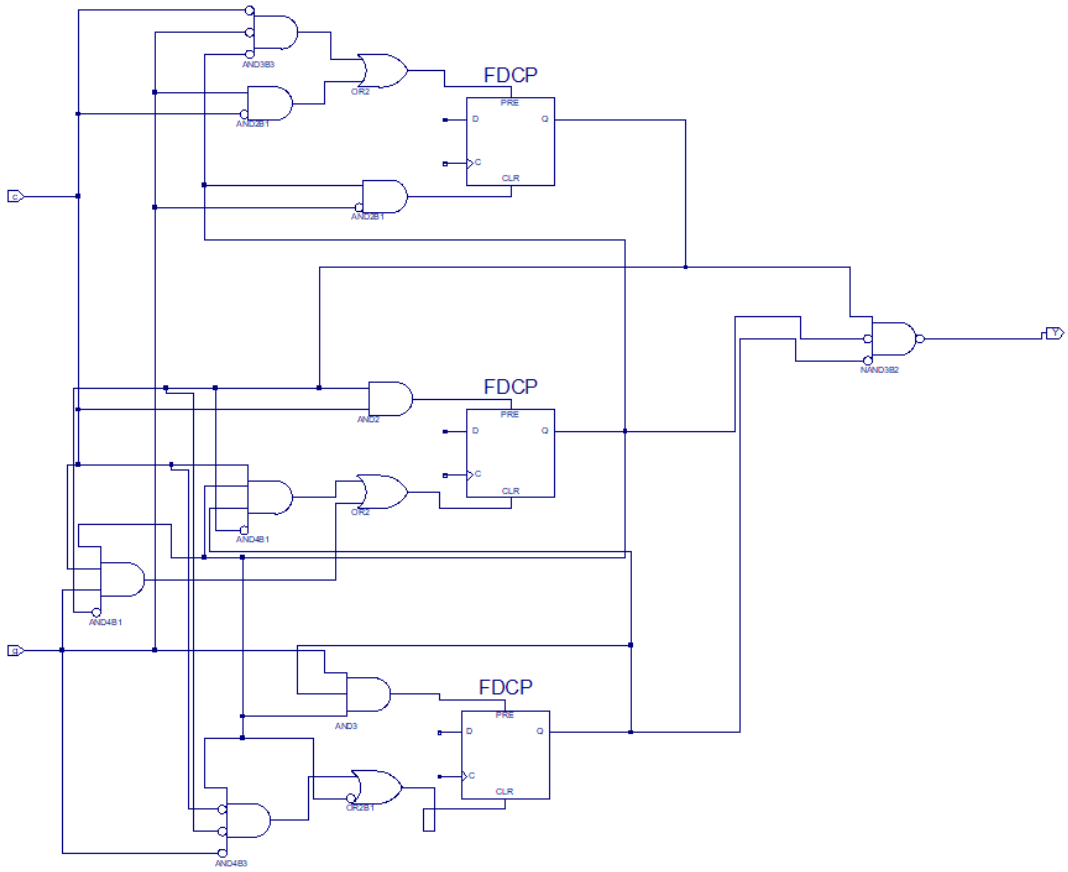
Conclusions:

Unfortunately, it can be seen from the simulation that I did not complete the task. The issues are:

- When the g is changed from high to low and the clock is high, the negative clock pulse should be copied, but isn't.
- There is "synchronisation problem". Q of all SR flip-flops is not reset. This causes that the output Y is 0 until the first positive pulse of the clock appears (Y should be 1 from the start and stay 1 unless negative pulse is copied)

I did not manage to fix the abovementioned issues. I did not find issues with transitions related to the copying of the negative pulse after g pulse ending at positive clock pulse, hence I wasn't able to find the problem causing the issue from the first point. For 2nd point, I didn't know how to fix these flip-flops in Xilinix.


FDCP
FDCP
FDCP
AND3B3
OR2
AND2B1
AND2B1
AND2
OR2
AND4B1
AND4B1
AND3
OR2B1
AND4B3
NAND3B2
C
D
Y
PRE
CLR
D
Q
C


FDCP
FDCP
FDCP
AND3B3
OR2
AND2B1
AND2B1
AND2
OR2
AND4B1
AND4B1
AND3
OR2B1
AND4B3
NAND3B2
C
D
Y
PRE
CLR
D
Q
C