Report from Lab 3

Detector of the positive edge

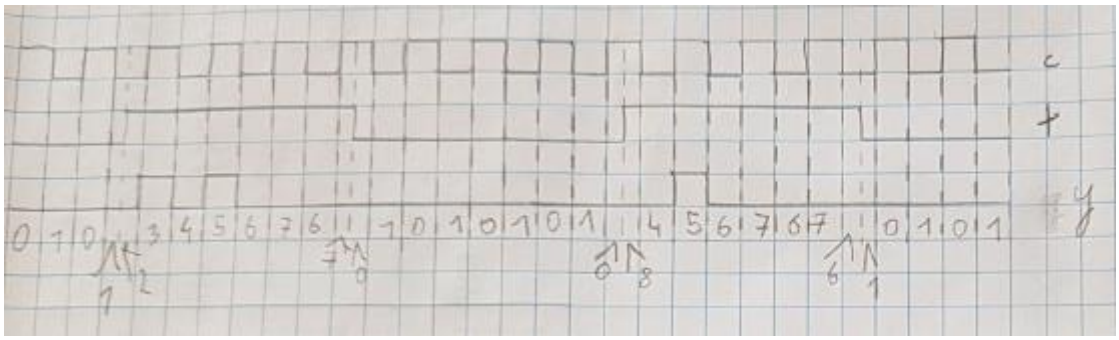
I declare that this piece of work, which is the basis for recognition of achieving learning outcomes in the Digital Circuits course, was completed on my own.

First and last name: Michał Łezka
Student record book number (Student ID number): 303873
Date: 30.04.2021

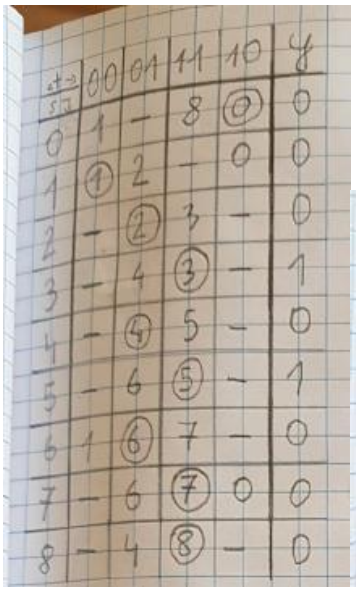
Theoretical design

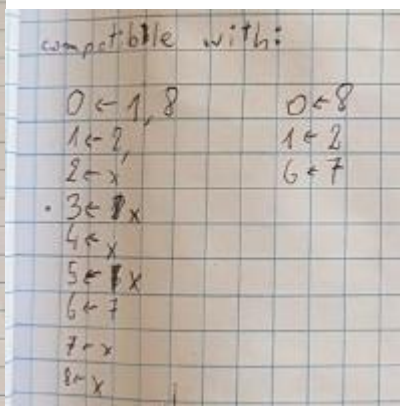
The task assigned to me was task 3. I was supposed to create a detector of the positive edge where device must copy a full, uncut pulse of the clock if the slope of the t is encountered at the high state of the clock and copy two consecutive clock pulses to the output if slope of the t is encountered at the low state of the clock.

First, I have created timing diagram and numbered the states:

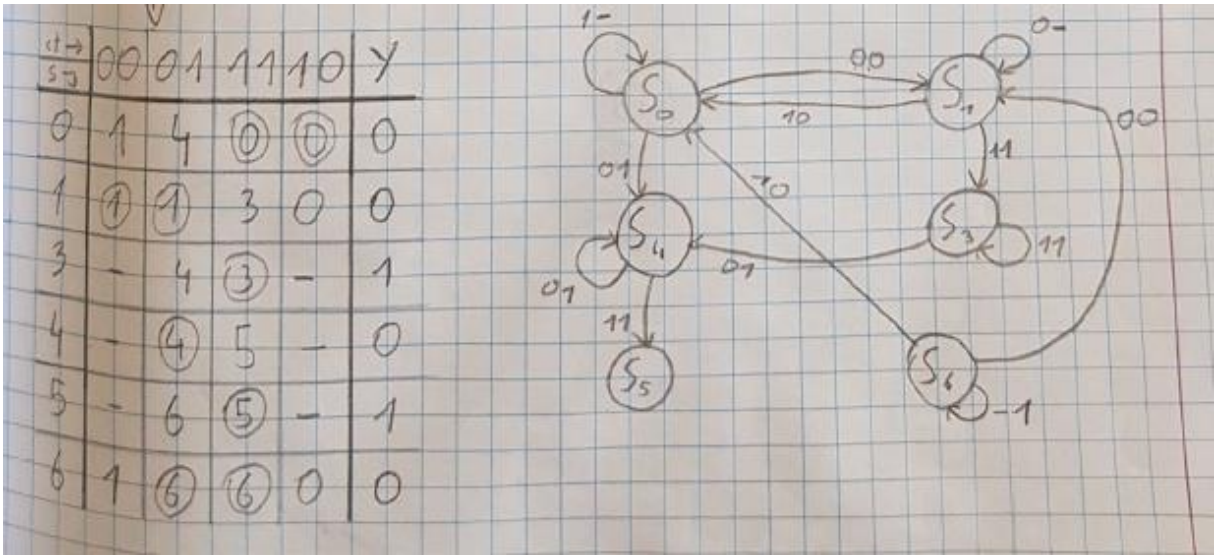


Now, based on that timing diagram I have created a table showing how states behave and searched for compatible states:



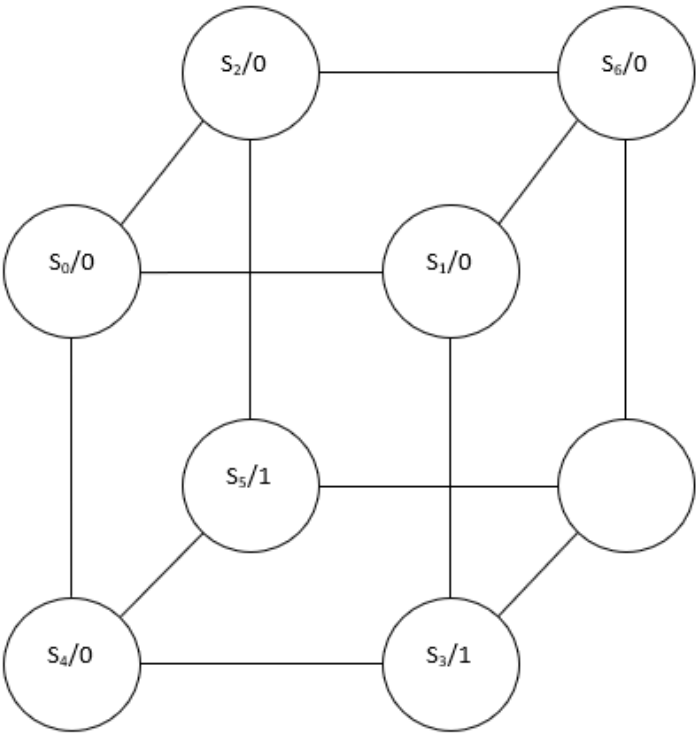


Now, I have merged states 0 with 8, 1 with 2 and 6 with 7 and got the table on the left:

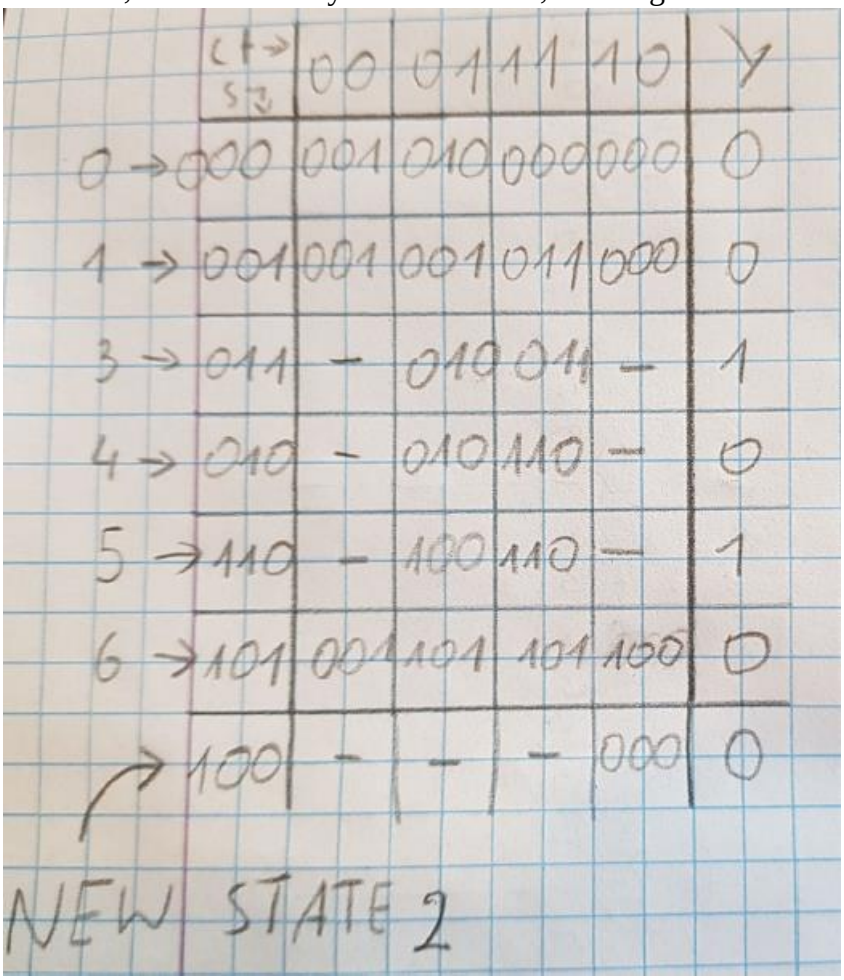


$(t \rightarrow / S_j)$	00	01	11	10	Y
0	1	4	(0)	(0)	0
1	(1)	(1)	3	0	0
3	-	4	(3)	-	1
4	-	(4)	5	-	0
5	-	6	(5)	-	1
6	1	(6)	(6)	0	0

and based on that table, I have drawn state transitions in a graphic way to find critical races. Then, I put it in a hypercube, but I didn't draw all the transitions as it was easy to put states in the cube and found drawing transitions again redundant:



With that, I have created yet another table, showing all the states and their transitions:



Now all that was left was transcoding it into SR flip-flops, but that part was done in excel as it is much quicker to do that there. What I did was rewrite the table above in excel, but with the remaining, not used state 111 and filled it with don't cares. Transcoding was done using the table provided during laboratory introduction and is also visible in the excel on the left:

	00	01	11	10
000	001	010	000	000
001	001	001	011	000
011	x	010	011	x
010	x	010	110	x
110	x	100	110	x
111	x	x	x	x
101	001	101	101	100
100	x	x	x	000

Q	→	Q'	S	R
0	→	0	0	–
0	→	1	1	0
1	→	0	0	1
1	→	1	–	0

S0	00	01	11	10
000	1	0	0	0
001	x	x	x	0
011	x	0	x	x
010	x	0	0	x
110	x	0	0	x
111	x	x	x	x
101	x	x	x	0
100	x	x	x	0

S1	00	01	11	10
000	0	1	0	0
001	0	0	1	0
011	x	x	x	x
010	x	x	x	x
110	x	0	x	x
111	x	x	x	x
101	0	0	0	0
100	x	x	x	0

S2	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	x	0	0	x
010	x	0	1	x
110	x	x	x	x
111	x	x	x	x
101	0	x	x	x
100	x	x	x	0

R0	00	01	11	10
000	0	x	x	x
001	0	0	0	1
011	x	1	0	x
010	x	x	x	x
110	x	x	x	x
111	x	x	x	x
101	0	0	0	1
100	x	x	x	x

R1	00	01	11	10
000	x	0	x	x
001	x	x	0	x
011	x	0	0	x
010	x	0	0	x
110	x	1	0	x
111	x	x	x	x
101	x	x	x	x
100	x	x	x	x

R2	00	01	11	10
000	x	x	x	x
001	x	x	x	x
011	x	x	x	x
010	x	x	0	x
110	x	0	0	x
111	x	x	x	x
101	1	0	0	0
100	x	x	x	1

S0 = ~c~t

R0 = c~t + ~ctQ1

S1 = ~ct~Q1~Q0 + ct~Q2Q0

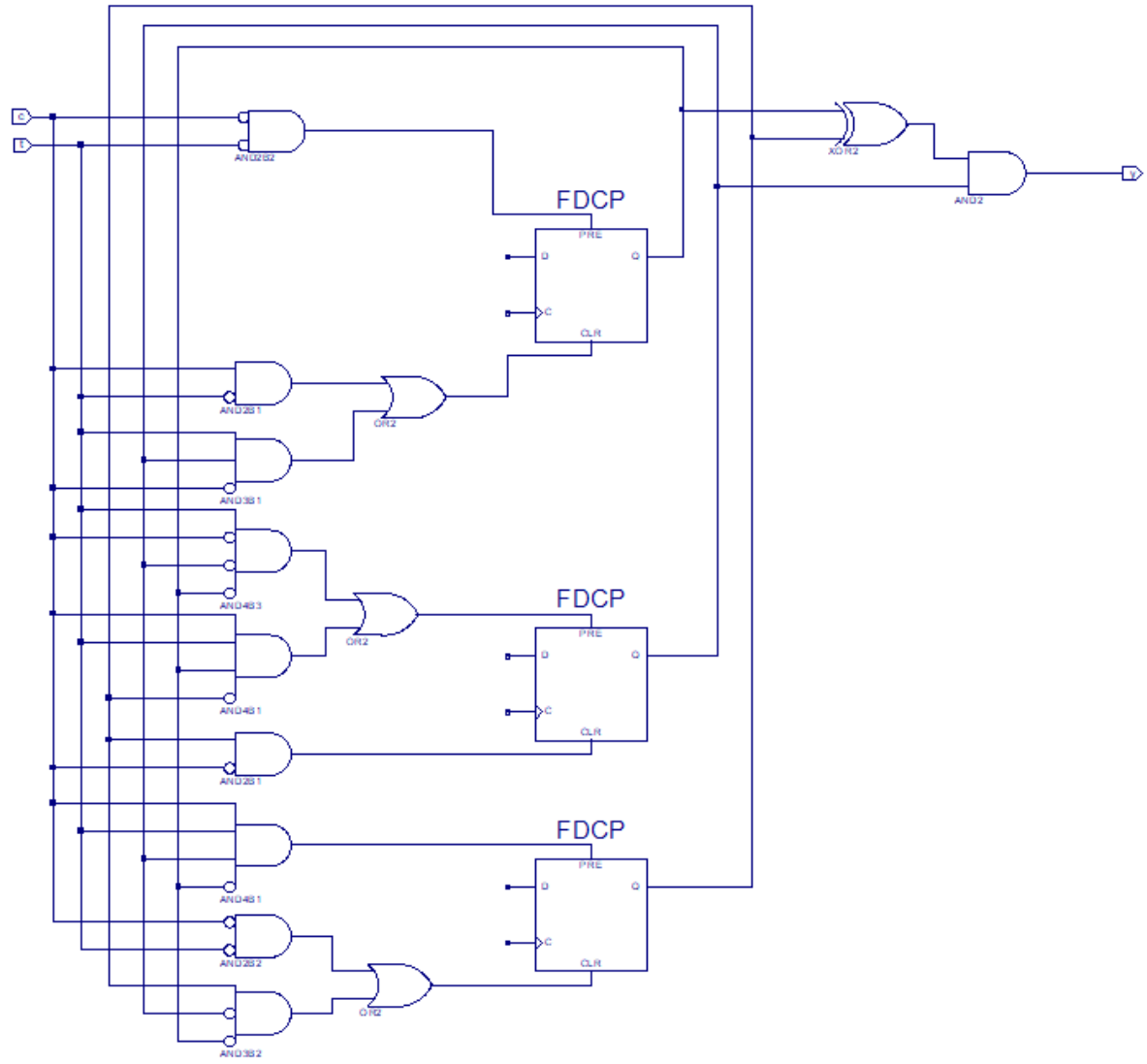
R1 = ~cQ2

S2 = ctQ1~Q0

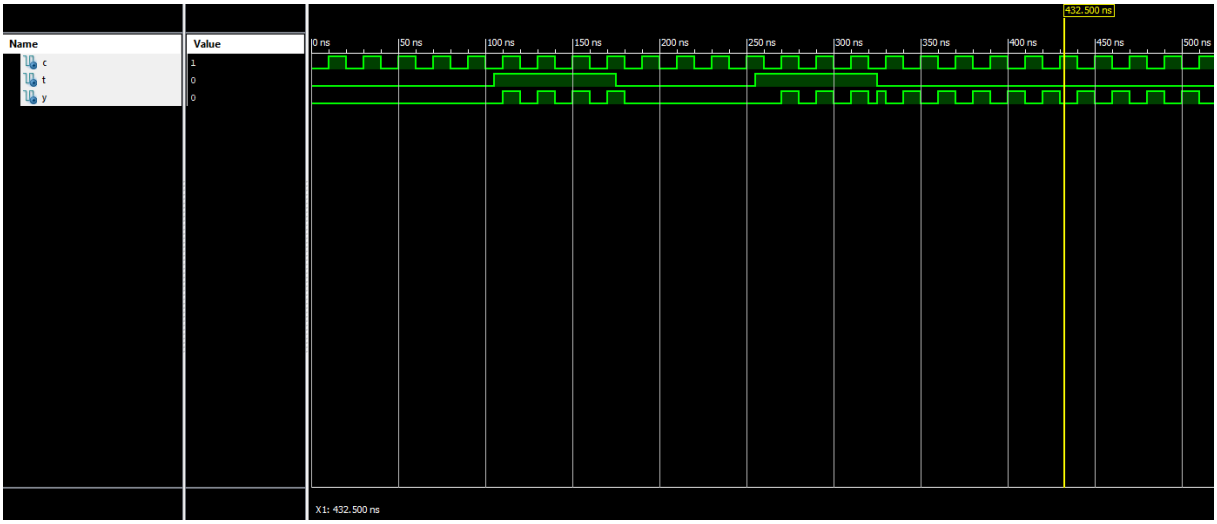
R2 = ~c~t + Q2~Q1~Q0

Schematics and test results

Having the Karnaugh tables solved, it was simple to implement it in the Xilinix as a few “and” and “or” gates. Because you can’t add output as an input, if I needed Q0, Q1 or Q2 as input, I drew a wire from the proper flip-flop and connected it to the input where needed. This is the schematic:



And this is how the testbench looked like:



As we can see, it doesn't work properly, it triggers on the positive edge but then it doesn't do what it was supposed to.

Troubleshooting

After seeing that the detector is not working properly, I started looking through everything to find some kind of mistake. I found 4 mistakes in my transcoding:

S0	00	01	11	10
000	1	0	0	0
001	x	x	x	0
011	x	0	1	x
010	x	0	0	x
110	x	0	0	x
111	x	x	x	x
101	1	1	1	0
100	x	x	x	0

→

S0	00	01	11	10
000	1	0	0	0
001	x	x	x	0
011	x	0	x	x
010	x	0	0	x
110	x	0	0	x
111	x	x	x	x
101	x	x	x	0
100	x	x	x	0

but to my surprise, it didn't change the result at all. Sadly, after another hour of looking through everything I didn't find any other mistakes, I found things that could've been a bit better optimised, but what I initially did was, from what I saw, not incorrect.

Conclusions

Overall, the experiment succeeded only partially. Some kind of mistake creeped in and I couldn't find said mistake. Even though the results are wrong, I believe that it is caused by a very well-hidden mistake and not lack of knowledge on how to perform the task.