Digital Circuits Theory - Laboratory						
Academic year	Laboratory exercises on	Mode of studies	Field of studies	Supervisor	Group	Section
2020/2021	Wednesday 15:30 – 17:00	SSI	Informatics	KP	1	3

Report from Exercise No 4

Performed on: 21.10.2020

Exercise Topic: Asynchronous Sequential Circuits

Performed by:

Dawid Grobert

Purpose of the exercises

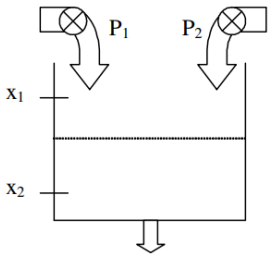
The aim of the exercises was to become familiar with asynchronous sequential circuits. In comparison to the previous task (synchronous sequential circuits), these circuits do not use the clock. When it comes to Flip-Flops, such circuits may contain latches rather than flip-flops, because an asynchronous circuit does not need the precise timing control supported by flip-flops.

In our case, we used the asynchronous Flip-Flops inputs – in my case, the asynchronous inputs of D Flip-Flop to obtain behaviour similar to SR Flip-Flop. In order to do the task, we had to be aware of certain restrictions that exist in these circuits. By consciously following the rules, the aim of the exercises was to obtain correctly functioning circuits.

1 Description of the first task

Task 1

Design a circuit controlling the switching of two pumps. The pumps P_1 and P_2 should be switched on alternately (only one pump can work at a time) when water falls below the level of the sensor x_2 (i.e. when $x_2 = 0$). Working pump should be switched off when the water level exceeds the level of the sensor x_1 (i.e. when $x_1 = 1$). Assume that water level grows when pumps are on, and that it decreases when none of pumps is working.



1.1 Timing Chart

Let us assume that at the beginning the container is empty and there is no water in it. We get the following timing chart. I have signed the states for the easier completion of the subsequent primitive flow map.

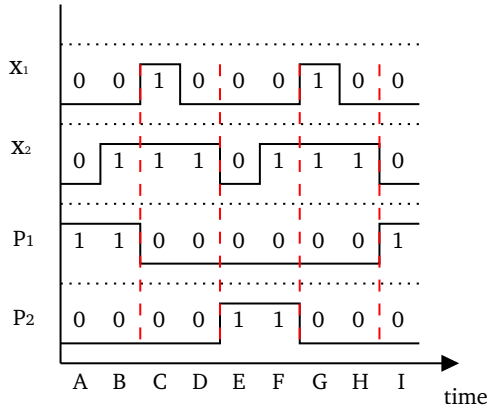


Figure 1: Timing chart made during laboratories

1.2 Implementation

Using huffman method, at first we create the primitive flow map.

Present State	$x_1 x_2$ = 00	01	11	10	P_1	P_2
1	(A)	B			1	0
2		(B)	C		1	0
3		D	(C)		0	0
4	E	(D)			0	0
5	(E)	F			0	1
6		(F)	G		0	1
7		H	(G)		0	0
8	A	(H)			0	0

Next state

Reduction:

A - B

C - D

E - F

G - H

There is nothing here that can be reduced in the columns. Therefore, after the reduction of the lines, we get the given map:

Present State	$x_1 x_2$ = 00	01	11	10	P_1	P_2
1,2	(A)	(B)	C		1	0
3,4	E	(D)	(C)		0	0
5,6	(E)	(F)	G		0	1
7,8	A	(H)	(G)		0	0

Next state

$q_1 q_0$ \ $x_1 x_2$	00	01	11	10
00	00	00	01	
01	11	01	01	
11	11	11	10	
10	00	10	10	

$Q_1 Q_0$

From the given map we get two Karnaugh maps for Q_1 and Q_0.

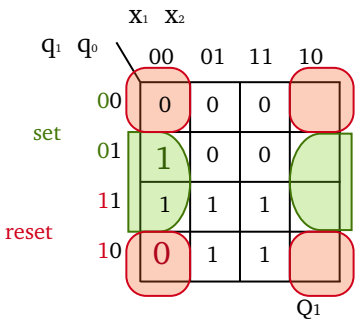
$q_1 q_0$ \ $x_1 x_2$	00	01	11	10
00	0	0	0	
01	1	0	0	
11	1	1	1	
10	0	1	1	

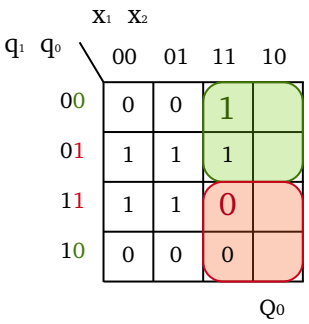
Q_1

$q_1 q_0$ \ $x_1 x_2$	00	01	11	10
00	0	0	1	
01	1	1	1	
11	1	1	0	
10	0	0	0	

Q_0

And from both maps I get an implementation for flip flops SR (of course, I had a D Flip-Flops, although I only used asynchronous inputs.).





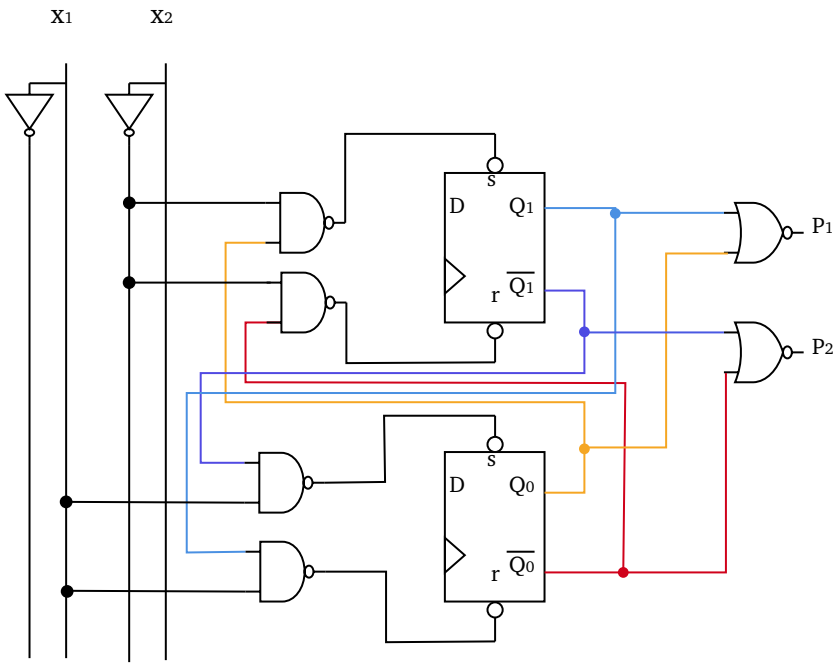
From the maps we can read:

$$\overline{s_1} = \overline{\overline{x_2}\ q_o} \qquad \overline{s_0} = \overline{x_1\ \overline{q_1}}$$
$$\overline{r_1} = \overline{\overline{q_o}\ \overline{x_2}} \qquad \overline{r_0} = \overline{\overline{q_1}\ x_1}$$

And the output is:

$$P_1 = \overline{q_1 + q_0}$$
$$P_2 = q_1 q_0 = \overline{\overline{q_1} + \overline{q_0}}$$

1.3 Circuit diagram



2 Description of the second task

Task 4

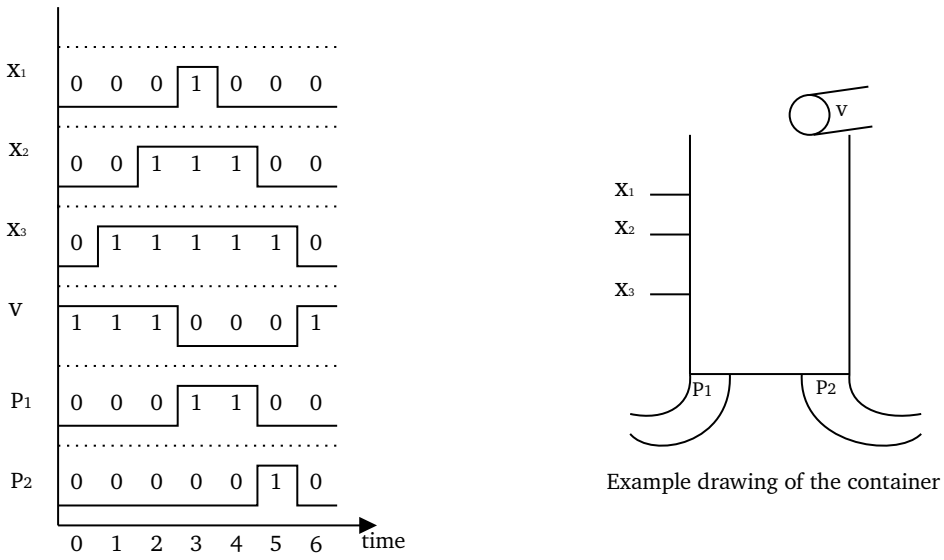
The container for liquid waste from the technological process of paint production is equipped with three level sensors, from x_1 to x_3, x_1 placed the highest, x_3 as the lowest, and x_2 in the middle.

The sensors work in a binary way — when the liquid in the container is below a sensor it shows 0, and when the liquid exceeds the level of the sensor it shows 1. The waste flows to the container through the valve V, and the container is emptied by two pumps, P_1 and P_2. Design a circuit controlling the operation of the waste container, with the following assumptions:

- when the container is empty (level of the liquid below the lowest sensor), the valve V becomes open ($V = 1$),
- when the waste reaches the level of the highest sensor, the valve V should be closed, and the pump P_1 starts working,
- when the level of the liquid falls below the middle sensor, pump P_1 should be stopped, and pump P_2 turned on.
- when the waste container becomes empty, the working pump should be switched off. The whole system should work in a cyclic manner.

2.1 Timing Chart

Once again, I will assume that at the beginning the container is empty and there is no water in it. We get the following time chart:



Example drawing of the container

Figure 2: Timing chart made during laboratories

2.2 Implementation

Using huffman method, at first we create the primitive flow map.

Present State	000	001	011	010	110	111	101	100	V	P_1	P_2
0	(0)	1							1	0	0
1		(1)	2						1	0	0
2			(2)			3			1	0	0
3			4			(3)			0	1	0
4		5	(4)						0	1	0
5	0	(5)							0	0	1

X_1 X_2 X_3 — Next State

Reduction

0 - 1 - 2

3 - 4

After the reduction of the lines, we get the given map:

Present State	000	001	011	010	110	111	101	100	V	P_1	P_2
0,1,2	(0)	(1)	(2)			3			1	0	0
3,4		5	(4)			(3)			0	1	0
5	0	(5)							0	0	1
									-	-	-

X_1 X_2 X_3 — Next State

Race correction

q_1 q_0	000	001	011	010	110	111	101	100	V	P_1	P_2
00	00	00	00			01			1	0	0
01		11	01			01			0	1	0
11	10	11							0	0	1
10	00								-	-	-

X_1 X_2 X_3 — Q_1 Q_0

From the given map we get two Karnaugh maps for Q_1 and Q_0.

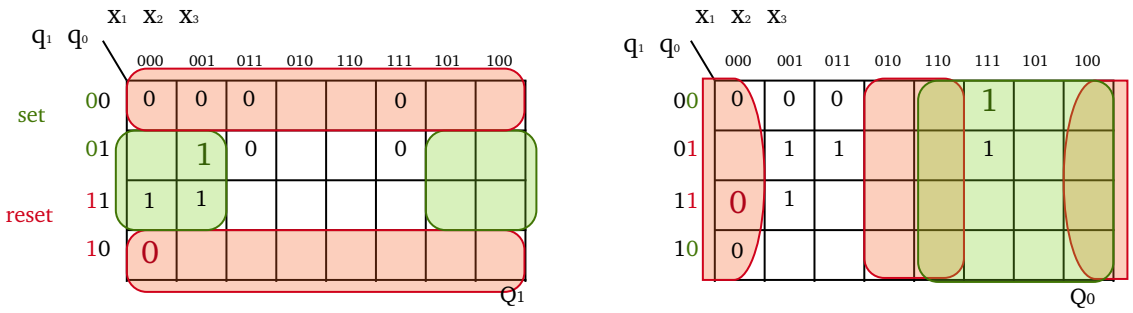
q_1 q_0	000	001	011	010	110	111	101	100
00	0	0	0			0		
01		1	0			0		
11	1	1						
10	0							

X_1 X_2 X_3 — Q_1

q_1 q_0	000	001	011	010	110	111	101	100
00	0	0	0			1		
01		1	1			1		
11	0	1						
10	0							

X_1 X_2 X_3 — Q_0

And from both nets I get an implementation for flip flops SR (of course, I had a D Flip-Flops, although I only used asynchronous inputs.).



From the maps we can read:

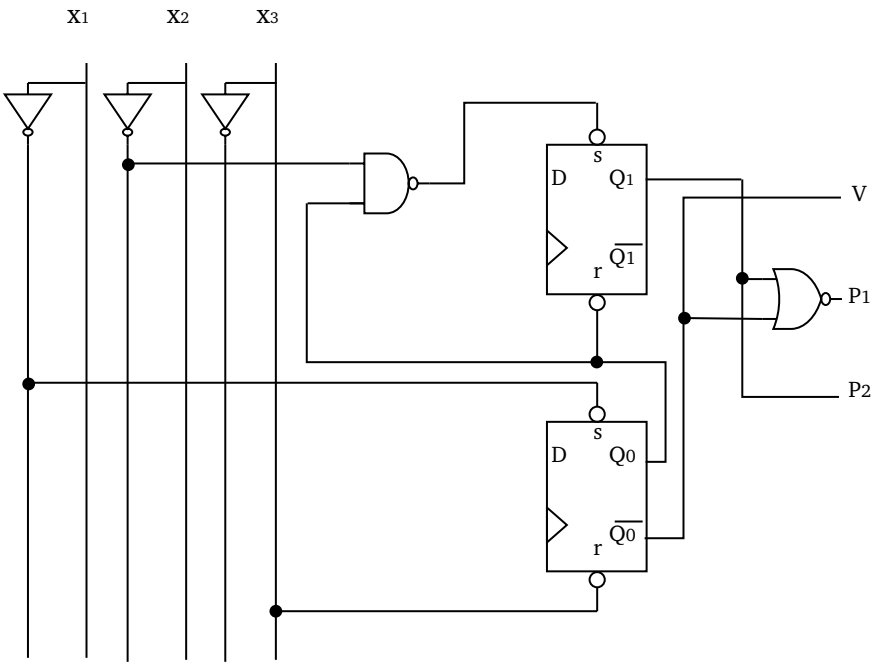
$$\overline{s_1} = \overline{q_0\,\overline{x_2}} \qquad \overline{r_1} = q_0$$
$$\overline{s_0} = \overline{x_1} \qquad \overline{r_0} = x_3$$

And the output is:

Q_1	Q_0	V	P_1	P_2
0	0	1	0	0
0	1	0	1	0
1	1	0	0	1
1	0	-	-	-

$$V = \overline{Q_0}$$
$$P_1 = \overline{Q_1}\,Q_0 = \overline{Q_1 + \overline{Q_0}}$$
$$P_2 = Q_1$$

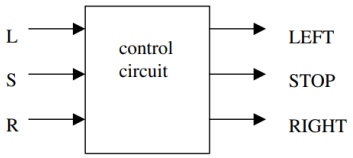
2.3 Circuit diagram



3 Description of the third task

Task 3

Design a circuit controlling the operation of the inertial two-directional engine. The engine can start to rotate only if it is stopped (*RIGHT = 0*, *LEFT = 0*, *STOP = 1*). The engine should start to rotate in right direction (*RIGHT = 1*) when button *R* is pushed and it should keep rotating until button *S* is pushed. Pushing the *R* or *L* button when engine rotates right should be ignored. The engine should start to rotate in left direction (*LEFT = 1*) when button *L* is pushed and it should keep rotating until button *S* is pushed. Pushing the *L* or *R* button when engine rotates left should be ignored. Similarly pushing *S* button when engine is stopped should not change its state. Pushing it when engine rotates in any direction should stop it by assigning outputs: *RIGHT = 0, LEFT = 0, STOP = 1*. Since all input buttons are monostable radio ones, it is assumed that only one of buttons *S, L, R* can be equal to one at a time.



We start the task by signing the states:

S_1 – engine moves left (L=1, S=0, R=0)
S_2 – engine stops (L=0, S=1, R=0)
S_3 – engine moves right (L=0, S=0, R=1)

3.1 Implementation

In this task, we will go straight to implementation. This is the type of task that allows you to imagine the operation of a digital circuit and transfer it directly to a primitive flow map. This method will be faster and not more difficult here.

Present state \ $X_L X_S X_R$	000	001	011	010	110	111	101	100
S1	S1	S1		S2				S1
S2	S2	S3		S2				S1
S3	S3	S3		S2				S3
-								

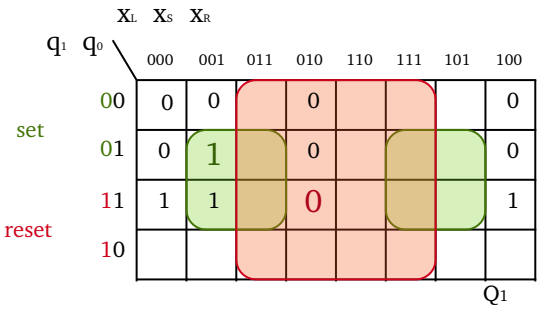
Next state

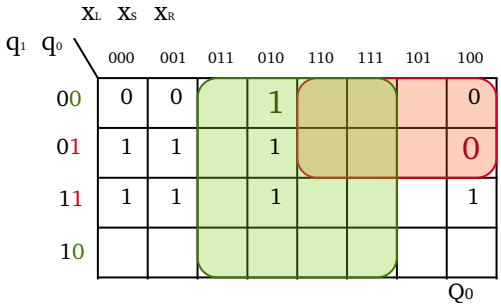
S1 = 00
S2 = 01
S3 = 11

$q_1 q_0$ \ $X_L X_S X_R$	000	001	011	010	110	111	101	100
00	00	00		01				00
01	01	11		01				00
11	11	11		01				11
10								

Q_1 Q_0

So let's bring out the groups for SR Flip-Flop.





We can read from these groups:

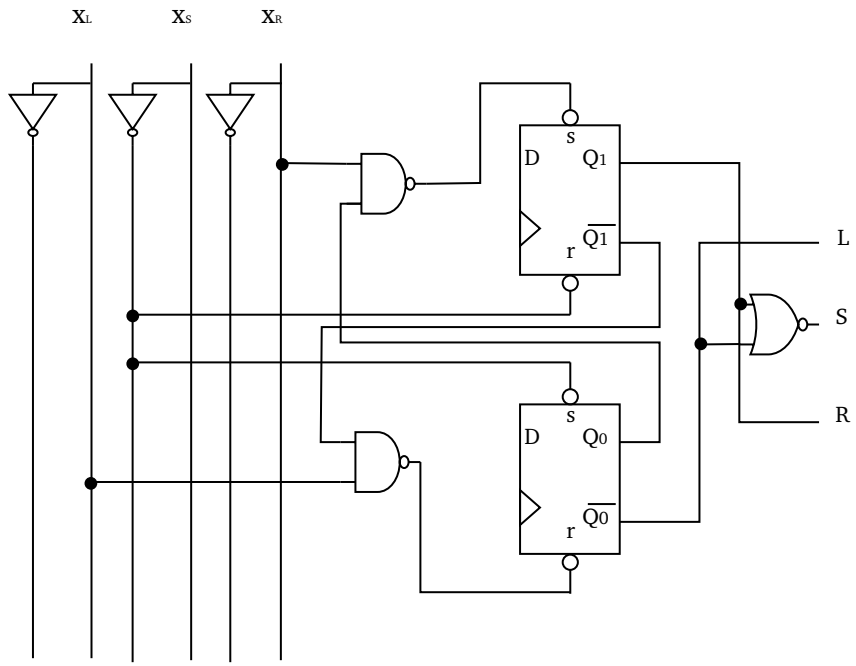
$$\overline{s_1} = \overline{q_0\ x_R} \qquad \overline{r_1} = \overline{x_s}$$
$$\overline{s_0} = \overline{x_s} \qquad \overline{r_0} = \overline{\overline{q_1}\ x_L}$$

Finally the output we can easily read from:

Q_1	Q_0	L	S	R
0	0	1	0	0
0	1	0	1	0
1	1	0	0	1
1	0	-	-	-

$$L = \overline{Q_0}$$
$$S = \overline{Q_1}\ Q_0 = \overline{Q_1 + \overline{Q_0}}$$
$$R = Q_1$$

3.2 Circuit diagram



4 Conclusions

I have built correctly functioning systems on the laboratory. The only mistake I made was in the last task (see page 8) where initially I did not include "don't care" states in the groups, so they were smaller and led to an incorrect result. Once this mistake has been corrected, everything worked out correctly.

The use of asynchronous Flip-Flops made it possible to get sufficiently small systems that did not require much haste.