Digital Circuits Theory - Laboratory						
Academic year	Laboratory exercises on	Mode of studies	Field of studies	Supervisor	Group	Section
2024/2025	Wednesday	SSI	Informatics	DP	1	1
	11:45 – 13:15					

Report from Exercise No 4

Performed on: 4.12.2024

Exercise Topic: Asynchronous sequential circuits

Performed by:

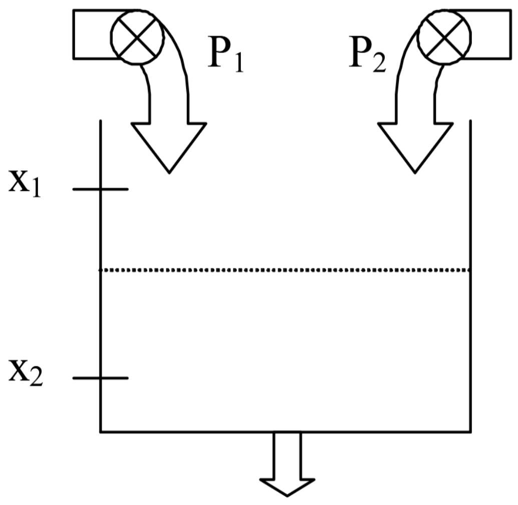
Piotr Copek
Zuzanna Micorek

Introduction

In this report, we present the design and implementation of various digital circuits aimed at solving specific control problems. These tasks involved the use of Karnaugh maps to minimize Boolean expressions and implement the desired functions efficiently. We employed different configurations of logic gates and flip-flops to achieve the required control logic.

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.



Solution

We started by creating a timing chart representing the behavior of the sensors x_1, x_2 and how they affect the pumps.

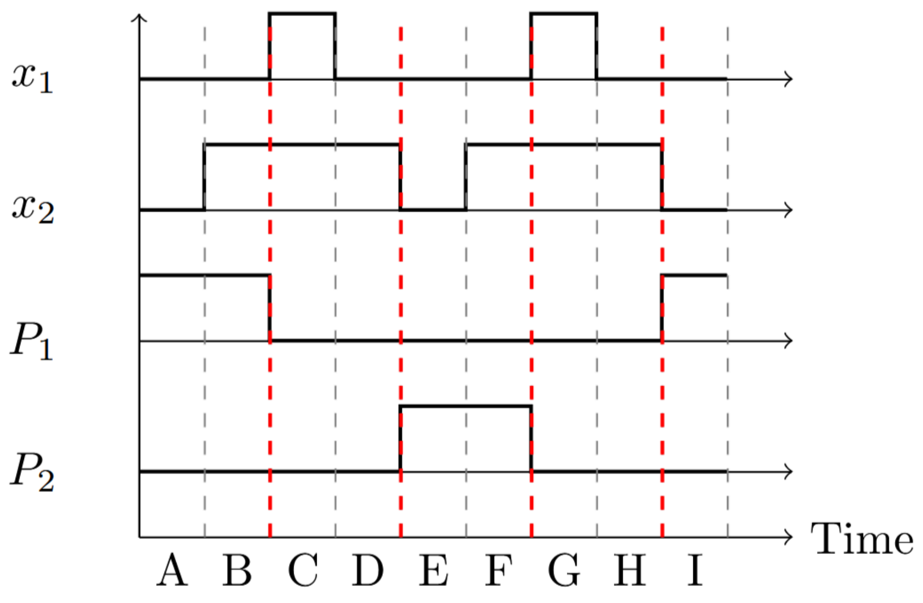


Figure 1 - timing chart for task 1 with all possible combinations of inputs x_1 x_2.

From the timing chart we created a flow chart.

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

Figure 2 - flow chart derived from timing chart.

We created a truth table based on the inputs from sensors x_1 and x_2. This table included all possible states of the system and the corresponding outputs.

Present State \ x_1x_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

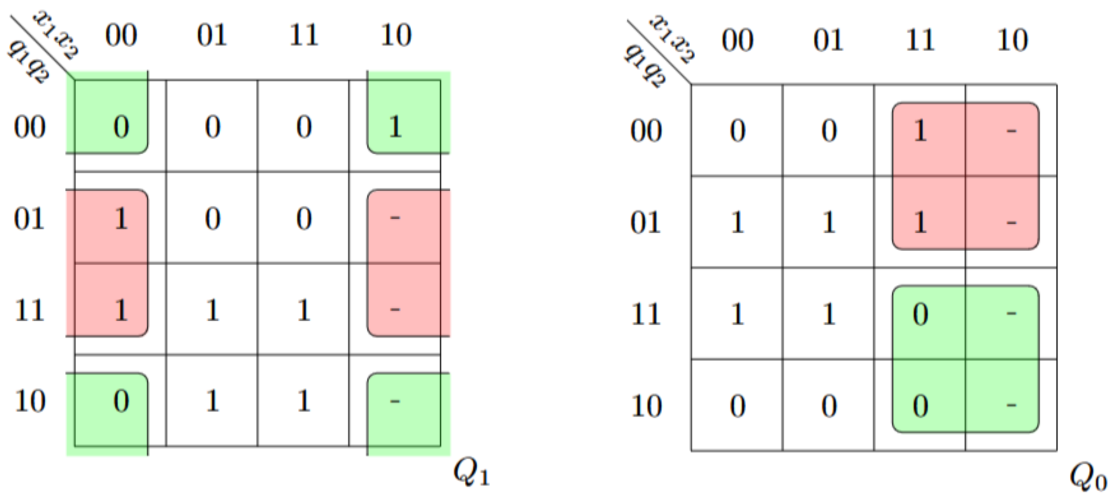
Figure 3 - karnaugh map derived from flow chart.

$$\begin{gathered} State \rightarrow q_1q_2 \\ 1,2 \rightarrow 00 \\ 3,4 \rightarrow 01 \\ 5,6 \rightarrow 11 \\ 7,8 \rightarrow 10 \end{gathered}$$

q_1q_0 \ x_1x_2	00	01	11	10
00	00	00	01	
01	11	01	01	
11	11	11	10	
10	00	10	10	

Q_1Q_0

Figure 4 - encoded karnaugh map.



$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	0	0	1
01	1	0	0	-
11	1	1	1	-
10	0	1	1	-

Q_1

$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	0	1	-
01	1	1	1	-
11	1	1	0	-
10	0	0	0	-

Q_0

Figure 5 - groups for s-r flipflops.

From the Karnaugh maps we obtained functions for s-r flip-flops:

$$\overline{s_1} = \overline{\overline{x_2} q_0}$$
$$\overline{r_1} = \overline{\overline{x_2}\,\overline{q_0}}$$

$$\overline{s_0} = \overline{x_1 \overline{q_0}}$$
$$\overline{r_0} = \overline{q_1 x_1}$$

And for outputs:

$$P_1 = \overline{q_1 + Q_0}$$
$$P_2 = q_1 q_0 = \overline{\overline{q_1} + \overline{q_0}}$$

We implemented the circuit using the flip-flop and output functions. The circuit used NAND, NOR, and NOT gates to achieve the required logic.

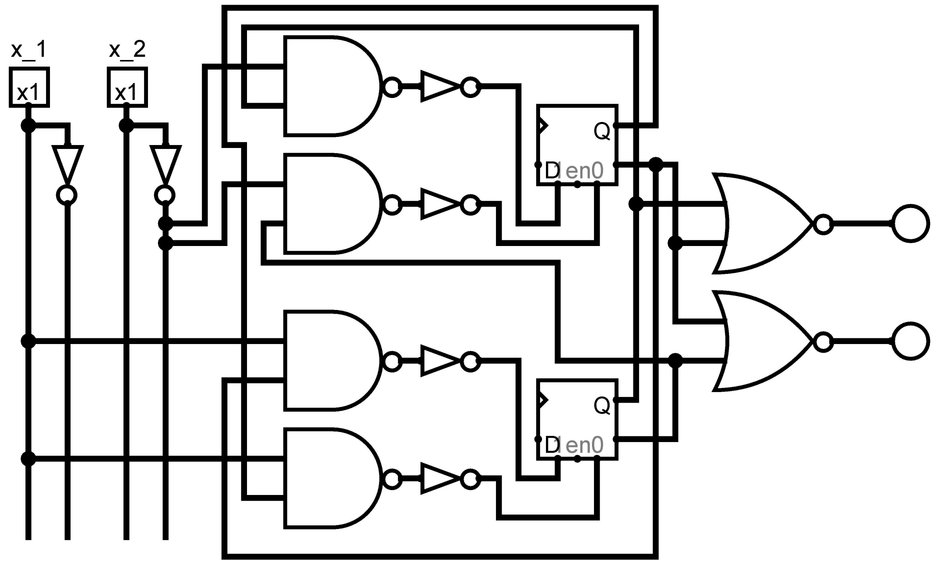


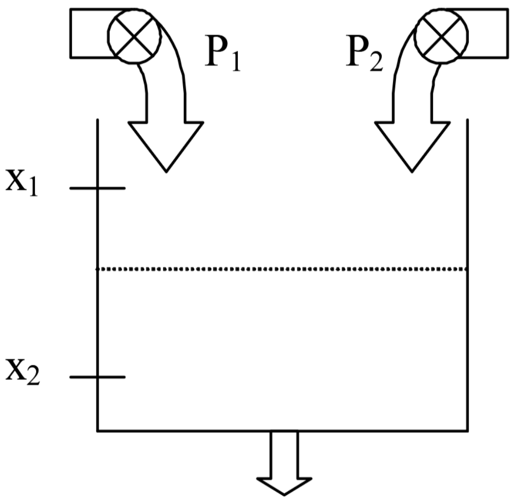
Figure 6 - implementation of task 1.

Summary

This task involved designing a pump control circuit using s-r flip-flops. By analyzing the sensor inputs and deriving the necessary logic, the circuit alternately switches the pumps based on water level conditions. The use of Karnaugh maps ensured the logic was optimized.

Task 2

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.



Additional task - can not use any kind of flip-flops.

Solution

We started by creating a timing chart representing the behavior of the sensors x_1, x_2 and how they affect the pumps.

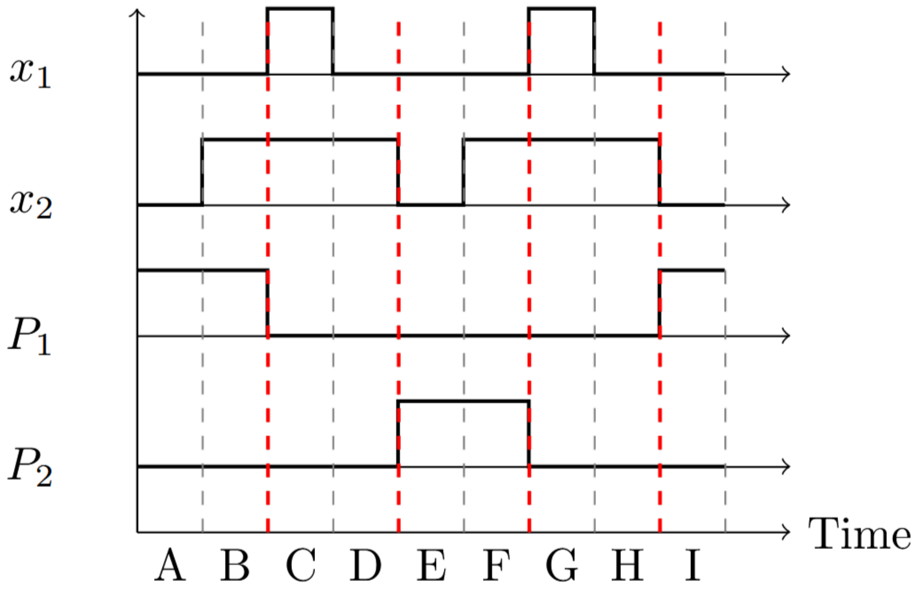


Figure 7 - timing chart for task 1 with all possible combinations of inputs x_1 x_2.

From the timing chart we created a flow chart.

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

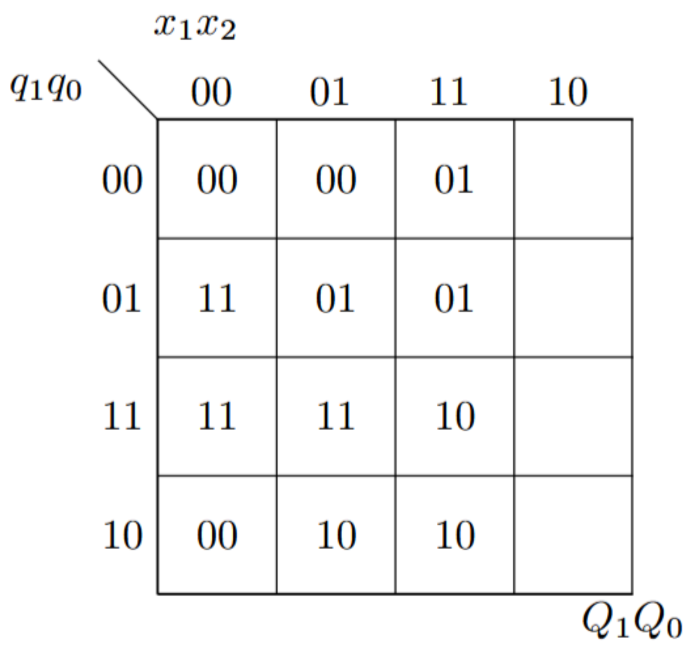
Figure 8 - flow chart derived from timing chart.

We created a truth table based on the inputs from sensors x_1 and x_2. This table included all possible states of the system and the corresponding outputs.

Present State \ x_1x_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

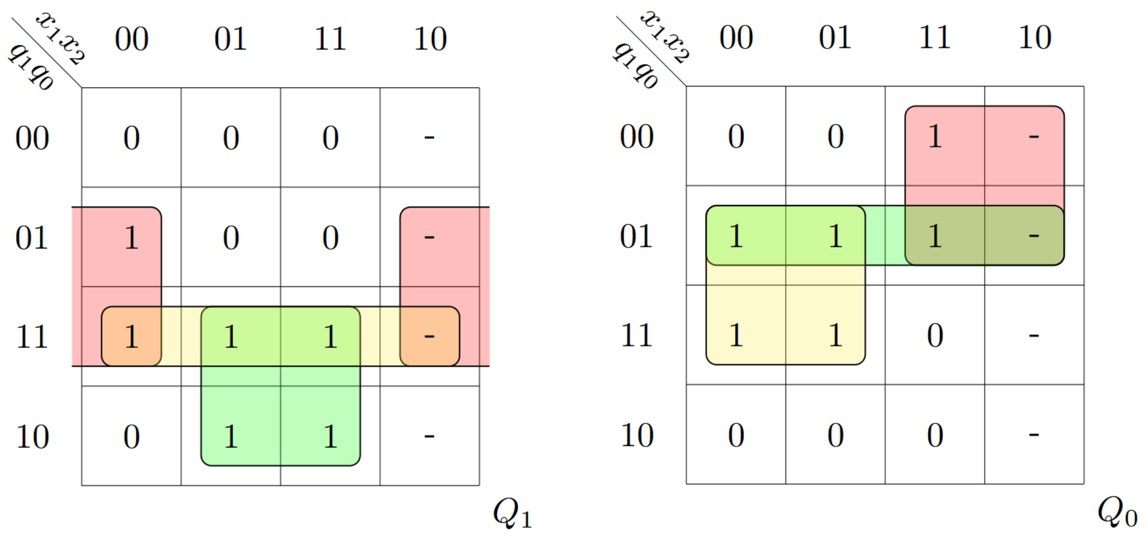
Figure 9 - karnaugh map derived from flow chart.



$q_1q_0 \backslash x_1x_2$	00	01	11	10
00	00	00	01	
01	11	01	01	
11	11	11	10	
10	00	10	10	

Q_1Q_0

Figure 10 - encoded karnaugh map.



$q_1q_0 \backslash x_1x_2$	00	01	11	10
00	0	0	0	-
01	1	0	0	-
11	1	1	1	-
10	0	1	1	-

Q_1

$q_1q_0 \backslash x_1x_2$	00	01	11	10
00	0	0	1	-
01	1	1	1	-
11	1	1	0	-
10	0	0	0	-

Q_0

Figure 11 - groups for feedback loops with taking hazard into consideration.

From the Karnaugh map we obtained input functions on which we applied De Morgan’s laws to transform Boolean expression into a form that could be implemented with NAND gates. We also obtained output functions.

$$Q_1 = \overline{\overline{\overline{\overline{q_0\,\overline{x_2}}\;\overline{q_1q_0}}}\;\overline{q_1q_0}}$$

$$Q_0 = \overline{\overline{\overline{\overline{q_0\,\overline{x_2}}\;\overline{q_1q_0}}}\;\overline{\overline{q_1}\,q_0}}$$

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

We implemented the circuit using the derived logic functions. The circuit was constructed using NAND gates and NOR gates.

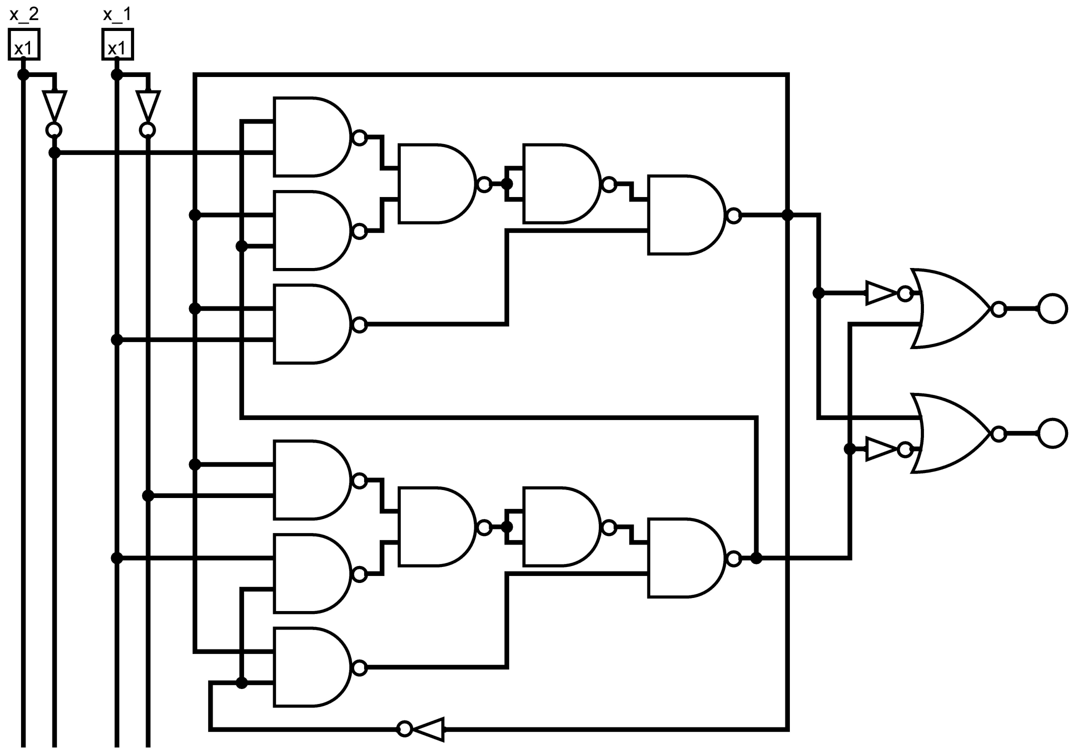


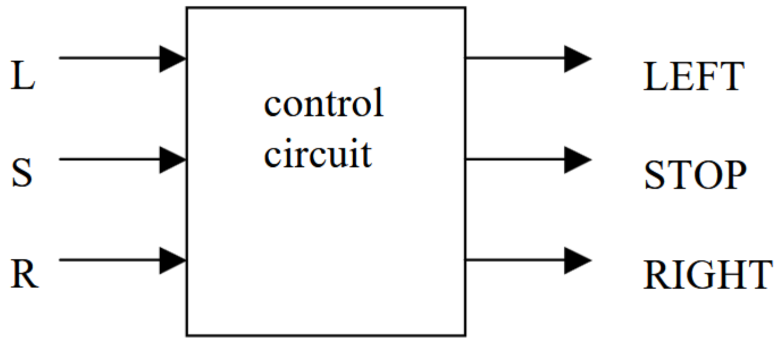
Figure 12 - implementation of task 1 with additional requirement.

Summary

In this task, we designed a circuit to control the switching of two pumps without using any flip-flops. This was achieved by implementing the logic functions from the truth table and Karnaugh maps, utilizing NAND and NOR gates to accomplish the required logic.

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.



Solution

We started by creating a flow chart, which includes transitions between states.

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

Figure 13 - flow chart used to solve task 3.

We encoded the states:

$$\begin{gathered} State \to q_1q_0 \\ S1 \to 00 \\ S2 \to 01 \\ S3 \to 11 \end{gathered}$$

Then, we created a truth table based on the inputs (L, S, R) and current states.

q_1q_0	$x_Lx_Sx_R$							
	000	001	011	010	110	111	101	100
00	00	00	–	01	–	–	–	00
01	01	11	–	01	–	–	–	00
11	11	11	–	01	–	–	–	11
–	–	–	–	–	–	–	–	–

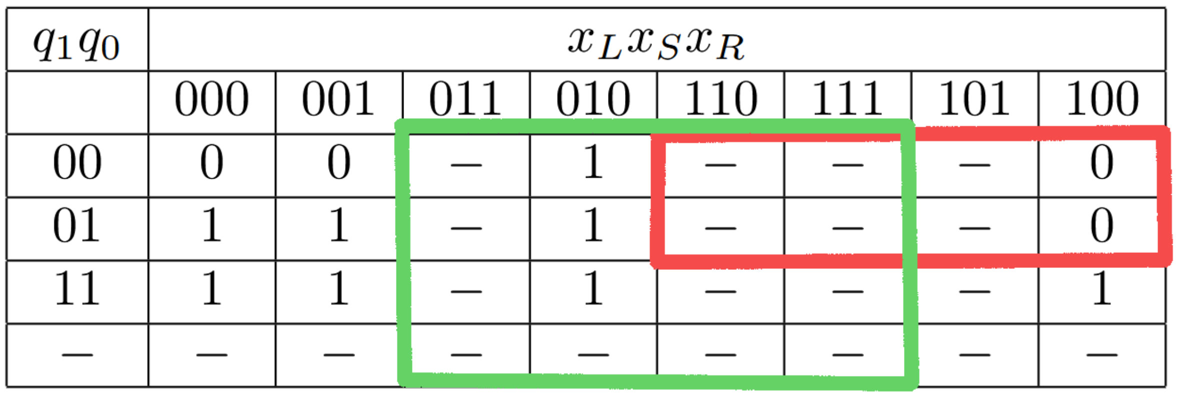
Q_1 Q_0

Figure 14 - encoded karnaugh map for both next states (Q_1Q_2).

q_1q_0	$x_Lx_Sx_R$							
	000	001	011	010	110	111	101	100
00	0	0	–	0	–	–	–	0
01	0	1	–	0	–	–	–	0
11	1	1	–	0	–	–	–	1
–	–	–	–	–	–	–	–	–

Q_1

Figure 15 - karnaugh map for only Q_1 with marked groups.



q_1q_0	$x_Lx_Sx_R$							
	000	001	011	010	110	111	101	100
00	0	0	–	1	–	–	–	0
01	1	1	–	1	–	–	–	0
11	1	1	–	1	–	–	–	1
–	–	–	–	–	–	–	–	–

Q_0

Figure 16 - karnaugh map for only Q_0 with marked groups.

We derived functions for s-r flip-flops using Karnaugh maps:

$$\overline{s_1} = \overline{q_0 x_R}$$
$$\overline{r_1} = \overline{x_S}$$

$$\overline{s_0} = \overline{x_S}$$
$$\overline{r_0} = \overline{\overline{q_1}\ x_L}$$

From table below we obtained output functions:

Q_1	Q_0	L	S	R
0	0	1	0	0
0	1	0	1	0
1	1	0	0	1
1	0	–	–	–

Figure 17 - output table used to derive output functions.

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

We implemented the circuit using the derived functions. We used a combination of AND, OR, and NOT gates, along with NAND gates.

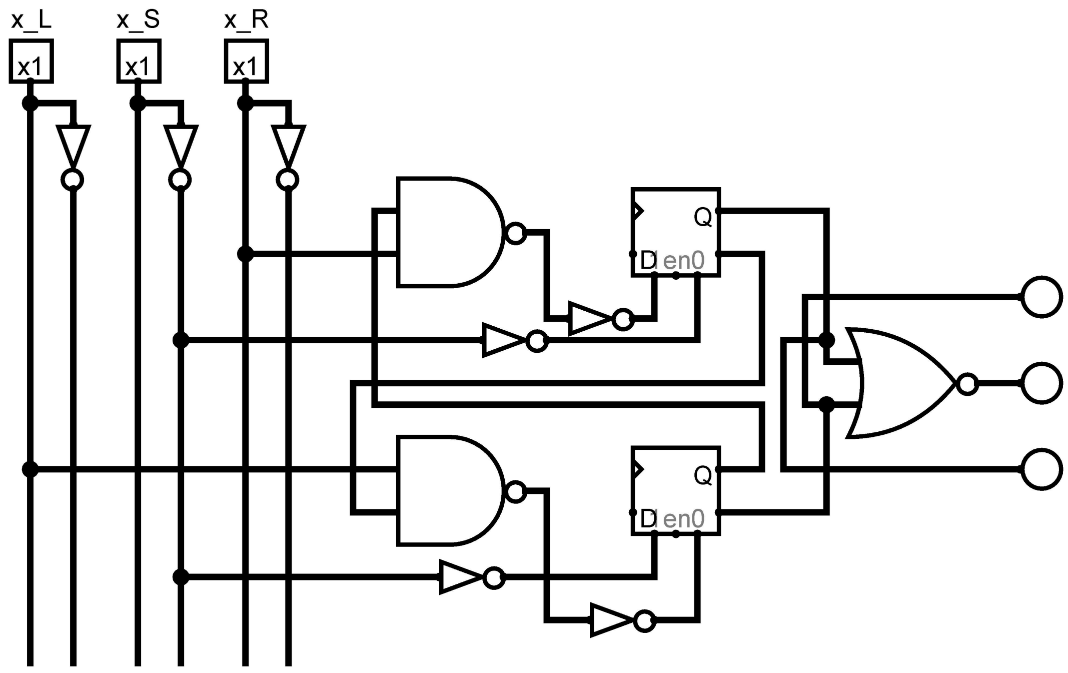


Figure 18 - implementation of task 3.

Summary

This task involved designing a control circuit for an inertial two-directional engine. The circuit ensures the engine starts and stops in the desired direction based on input, maintaining specific states for rotation and stop conditions. The implementation used a combination of basic logic gates and NAND gates.

Final Conclusions and Observations

In these tasks, we successfully designed various circuits for pump control and engine operation. The designs included using flip-flops, avoiding flip-flops, and ensuring proper state transitions based on sensor inputs and button presses. In each task we were creating truth tables, deriving logic functions, and implementing the circuits. The implementation utilized a combination of AND, OR, NOT, and NAND gates to achieve the desired logic, ensuring efficient and practical solutions for the given problems.