SCHOOL OF INFORMATION, COMPUTER AND COMMUNICATION TECHNOLOGY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY

LAB REPORT

EES 370 DIGITAL CIRCUIT LABORATORY

Lab 10 Digital Circuit Design

By

Mr. Makhapoom Euaumpon	**ID: 6222780833**
Mr. Paphana Yiwsiw	**ID: 6222780379**

Group No. 12 Section 2

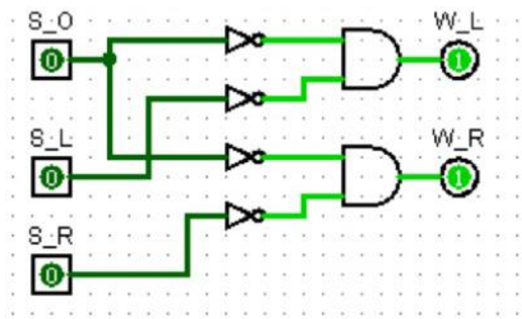
Date: 19 Apr 2021, Time: 13:00-16:00

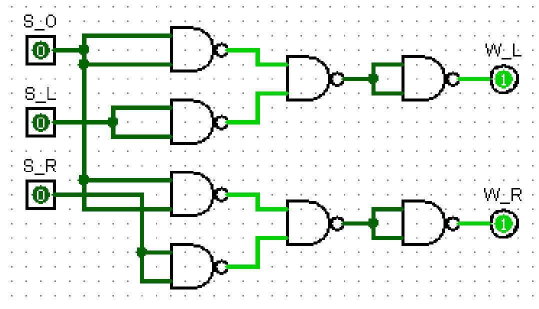
Objectives

1. To learn how to design and create the line-tracking robot logic circuit with given information by using all of previous lab knowledge.
2. To learn how to design and create with smart watering system for paddy field with given information by using all of previous lab knowledge.

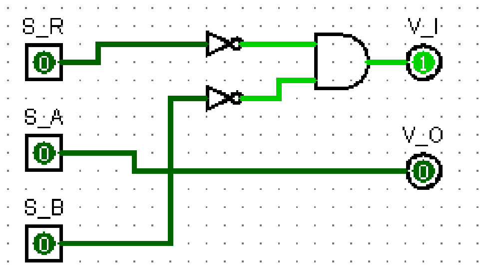
Lab Result

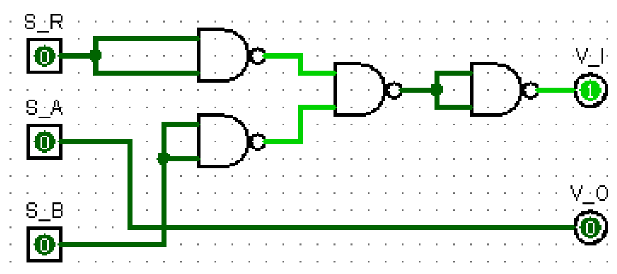
Q1: Line-tracking robot





Q2: Paddy field watering system





Discussion

In Part Q1, we designed the digital circuit for the line-tracking robot. This circuit has its input, infrared sensors and obstacle detection sensor, and its output, motors, in the digital signal that obtained from the robot. The circuit we design for this robot to function has 3 inputs, left infrared sensor (denoted as S_L), right infrared sensor (denoted as S_R) and obstacle detection sensor (denoted as S_O). From the robot, each infrared sensor will send signal as 1 when it detected black line and 0 otherwise. For the obstacle detection sensor, we made assumption as the sensor will send signal as 1 when it detected the object in front of the robot and 0 otherwise. Also, 2 outputs are generated from this circuit which are left motor (denoted as W_L) and right motor (denoted as W_R). If the signal send to the motor is 1, the motor will turn on and make the wheel connected to it will spin, thus make the robot move. If the signal send to the motor is 0, the motor will turn off and make the wheel stop. The robot will stop when both sensor detected a black line. If the obstacle detection sensor detected an obstacle in front, the robot will immediately stop. To have our robot turns left, we must make the wheel on the left to stop spinning and keep the right wheel spins. For turn right, it is the other way around, right wheel stop to spin and left wheel spins.

From above, we have all necessary information to create our truth table and circuit for this robot. Beginning with truth table, the truth table of this line-tracking robot are as follow:

Inputs			Outputs		Description
S_O	S_L	S_R	W_L	W_R	
0	0	0	1	1	Straight line
0	0	1	1	0	Turn right
0	1	0	0	1	Turn left
0	1	1	0	0	Finished
1	0	0	0	0	Obstacle detected
1	0	1	0	0	
1	1	0	0	0	
1	1	1	0	0	

Figure 1: Truth table of the line-tracking robot.

From above, we can derive K-maps and Boolean expressions from the truth table as follows:

W_L

S_L S_R \ S_O	0	1
00	1	0
01	1	0
11	0	0
10	0	0

W_R

S_L S_R \ S_O	0	1
00	1	0
01	0	0
11	0	0
10	1	0

$W\_L = (S\_O)'(S\_L)'$

$W\_R = (S\_O)'(S\_R)'$

Figure 2: K-maps and Boolean expressions of line-tracking robot.

After we obtained Boolean expressions, we can now create a circuit for this robot. We created this circuit in Logisim. We created a normal AND-OR gates circuit and NAND gates only circuit by using previous lab knowledge and apply it here. Thus, both versions of circuit of the line-tracking robot are shown here.

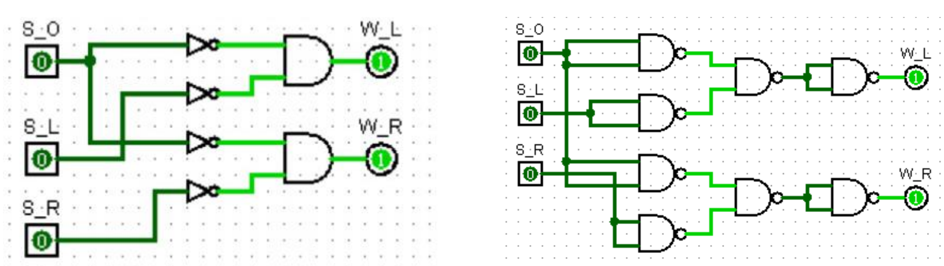


Figure 3: Line-tracking robot circuit using AND-OR gates (left) and NAND gates only (right).

After this, we can construct the circuit using breadboards and ICs and connect the circuit to a real robot.

In Part Q2, we designed a digital circuit for the smart watering circuit in paddy field. This smart watering system has three-level water sensor and rain gauge which we are going to connect to our circuit as inputs. The three-level water sensor consist of 2 sensors which we will denoted as S_A and S_B. S_A as sensor A which is located at the upper end and S_B as sensor B which located at the lower end of three-level water sensor. The sensor will send signal as 1 when water is above the sensor and 0 otherwise. We want water to be between sensor A and B. Thus, we want to get the water in paddy field to have a signal from S_A and S_B as 01. If the water is too low, we will get the signal as 00. If the water is too high, the signal will be 11. Also, there is no such case that the water can be above sensor A and below sensor B (frankly speaking, there is no way that water can float in the air), thus if the S_A and S_B have signal as 10, we will use don't care (X) for this case. Next component in this watering system is rain gauge which will denoted as S_R which it will send signal as 1 if it is raining and 0 otherwise. When it is raining in this paddy field, we want to control the flow of water only when the water is too much in the field and we want the excess water out of the field. To control the water in this paddy field, we have 2 water valves: inlet valve, denoted as V_I, to let the water flow into the field if the signal is 1 and outlet valve, denoted as V_O, to drain the water from the field if the signal is 1. These valves will be controlled and connected by outputs of the circuit.

From the information we have about this watering system, we can design the truth table for this circuit shown below.

Inputs			**Outputs**		**Description**
S_R	S_A	S_B	V_I	V_O	
0	0	0	1	0	Not raining, water is too low, open only the inlet valves
0	0	1	0	0	Not raining, water is optimal, close both valves.
0	1	0	X	X	Impossible case (discussed above).
0	1	1	0	1	Not raining, water is too high, open only the outlet valves
1	0	0	0	0	Raining, water is too low, valves not open (let rain fill the field).
1	0	1	0	0	Raining, water is optimal, close both valves.
1	1	0	X	X	Impossible case (discussed above).
1	1	1	0	1	Raining, water is too high, open only the outlet valves.

Figure 4: Truth table of watering system circuit.

After that, we can use K-maps and obtain Boolean expressions of this watering system circuit as shown in figure 5.

Then, we can construct this circuit into Logisim program for normal AND-OR gates circuit and NAND gates only circuit. Therefore, we have the final circuit of the watering system shown in figure 6 and we can implement this circuit using breadboard and ICs in real life.

V_I

S_A S_B \ S_R	0	1
00	1	0
01	0	0
11	0	0
10	X	X

V_O

S_A S_B \ S_R	0	1
00	0	0
01	0	0
11	1	1
10	X	X

V_I = (S_R)'(S_B)'

V_O = S_A

Figure 5: K-maps and Boolean expressions of watering system circuit.

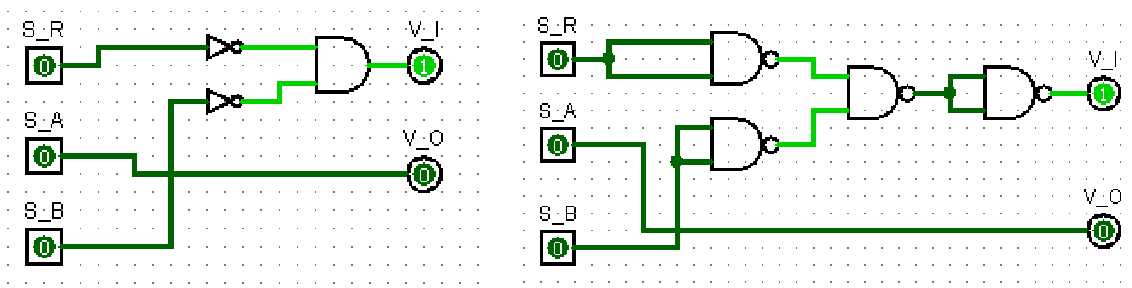


Figure 6: Smart watering system circuit using AND-OR gates (left) and NAND gates only (right).

Conclusion

In this lab, we all can say that this lab is using all previous lab knowledge that we learn so far from the beginning of the semester to design and create truth table both line-tracking robot logic circuit and smart watering system for paddy field logic circuit. And using Logisim program to create K-maps and obtain Boolean expressions. Then, after we acquired Boolean expressions, we can create logic circuits to make the system that we want to create to function correctly.