Lab 1

Madiba Hudson-Quansah

Question 1

1. Complete the truth table for R and C as a function of Q_1, Q_2, Q_3, and X
2. Complete the state table based on the state transition diagram as above.

Solution:

1.

Q_1	Q_2	Q_3	X	Total Amount	Temp	R	C
0	0	0	0	\$0.00	00	0	0
0	0	0	1	\$0.00	00	0	0
0	0	1	0	\$0.25	01	0	0
0	0	1	1	\$0.25	01	0	0
0	1	0	0	\$0.25	01	0	0
0	1	0	1	\$0.25	01	0	0
0	1	1	0	\$0.50	10	0	0
0	1	1	1	\$0.50	10	1	0
1	0	0	0	\$0.25	01	0	0
1	0	0	1	\$0.25	01	0	0
1	0	1	0	\$0.50	10	0	0
1	0	1	1	\$0.50	10	1	0
1	1	0	0	\$0.50	10	0	0
1	1	0	1	\$0.50	10	1	0
1	1	1	0	\$0.75	11	1	1
1	1	1	1	\$0.75	11	1	0

2.

Present State	Q_1	Q_2	Q_3	X	Next State	R	C
S0	0	0	0	0	S0	0	0
S0	0	0	1	0	S1	0	0
S0	0	1	0	0	S1	0	0
S0	1	0	0	0	S1	0	0
S0	0	0	1	1	S1	0	0
S0	0	1	0	1	S1	0	0
S0	1	0	0	1	S1	0	0
S1	0	0	1	0	S2	0	0
S1	0	1	0	0	S2	0	0
S1	1	0	0	0	S2	0	0
S1	0	0	1	1	S2	1	0
S1	0	1	0	1	S2	1	0
S1	1	0	0	1	S2	1	0
S2	0	0	1	0	S3	1	0
S2	0	1	0	0	S3	1	0
S2	1	0	0	0	S3	1	0
S2	0	0	1	1	S3	1	1
S2	0	1	0	1	S3	1	1
S2	1	0	0	1	S3	1	1
S3	1	0	0	0	S3	1	0
S3	0	0	1	1	S3	1	1
S3	0	1	0	1	S3	1	1
S3	1	0	0	1	S3	1	1

$Q_1 Q_2$ \ $Q_3 X$	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	1	1	1
10	0	0	1	1

Figure 1: D_1

3.

$$D_1 = Q_1Q_2 + Q_2Q_3 + Q_1Q_3$$

Q_1Q_2 \ Q_3X	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

Figure 2: D_0

$$\begin{aligned} D_0 &= \overline{Q_1Q_2}Q_3 + \overline{Q_1}Q_2\overline{Q_3} + Q_1Q_2Q_3 + Q_1\overline{Q_2Q_3} \\ &= Q_1 \oplus Q_2 \oplus Q_3 \end{aligned}$$

Question 2

Xilinx Vivado

Solution:

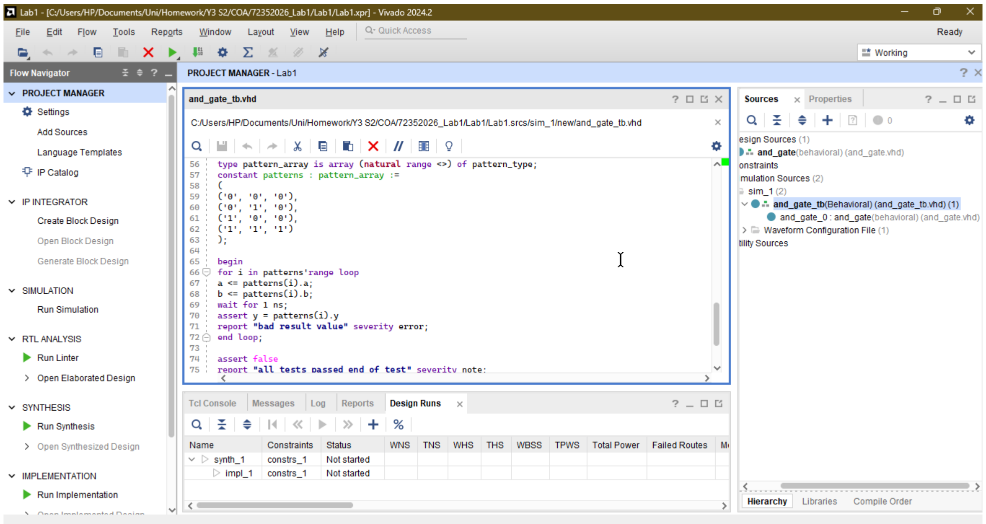

Figure 3: Test Bench

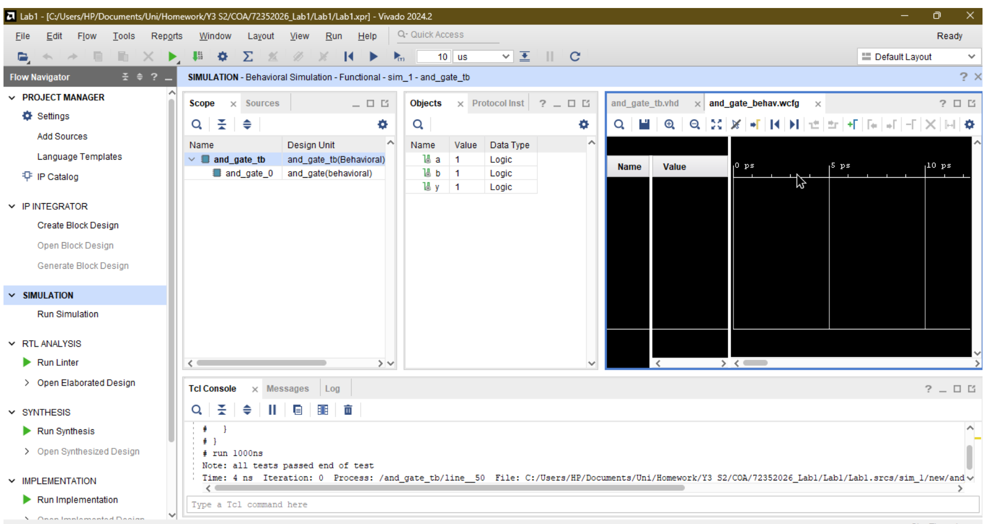

Figure 4: Test Bench Behavioural simulation

Question 3

What assumptions did you make while designing the vending machine coin counter? Were these assumptions realistic and clearly justified?

Solution:

- Coin Slots are independent - Each slot is independent of the other and the machine does not check which specific slot the quarters were placed in only the total amount. This is a realistic assumption as this reflects how real vending machines work.
- Change is limited to 1 quarter - The machine only returns 1 quarter as change. This is an assumption to keep the design simple and easy to implement.
- Slots only accept valid quarters - Only valid U.S. quarters are accepted by the machine. This is a realistic assumption as the machine is designed to accept only quarters.

Question 4

How did you decide the outputs for R (release soda) and C (return change)? Could there be alternative conditions or approaches to achieve the same functionality?

Solution: The outputs for R and C were decided based on the total amount of money given to the machine and the soda chosen by the user. The machine releases a soda if the total amount is greater than or equal to the cost of the soda. An alternative implementation could be to have the machine release a soda if the total amount is greater than the cost of the soda by a certain amount. This would allow the machine to release a soda even if the user gives more money than required for the soda.

Question 5

While creating the state table, how did you ensure all possible transitions were accounted for? Were there any ambiguous or conflicting states that required special handling?

Solution: I ensured all possible transitions were accounted for by creating a state transition diagram and then creating a state table based on the diagram. I made sure to include all possible states and transitions in the state table. There were no ambiguous or conflicting states that required special handling.

Question 6

What were the key challenges you faced during the installation of Xilinx Vivado? Were there any system requirements or errors you needed to address?

Solution: I faced no challenges during the installation of Xilinx Vivado.

Question 7

How would you explain the importance of hardware description languages (like VHDL) in modern digital design to someone unfamiliar with the field?

Solution: Hardware description languages serve as a way to describe the behaviour of digital circuits in a way that is easily understood by both humans and computers. They allow designers to create complex digital circuits without having to worry about the low-level details of the hardware. This makes it easier to design, test, and debug digital circuits and allows for faster development times.