COL215 Hardware Assignment 2: 4-Digit-7-segment display

Vanshika (2023CS10746)
Laksh Goel (2023CS10848)

September 8, 2024

1 Task

1. Design a combinational circuit that takes a single 4-bit hexadecimal or decimal digit input from the switches and produces a 7-bit output for the seven-segment display of Basys 3 FPGA board.
2. Extend the design to create a circuit that drives all 4 displays for displaying 4 digits together
   (a) Reduce Frequency of CLK_IN (100Mhz)
   (b) Make a 4-bit input 4X1 Multiplexer
   (c) Determine the number of clock cycles in 100 Mhz for the refresh clock

2 Truth table

Binary-wxyz	Decimal	Hexa
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	b
1100	12	C
1101	13	d
1110	14	E
1111	15	F

Table 1: Logic minimization

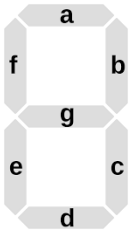

Figure 1: 7-segment display

Binary-wxyz	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	0	0	1	1
1010	1	1	1	0	1	1	1
1011	0	0	1	1	1	1	1
1100	1	0	0	1	1	1	0
1101	0	1	1	1	1	0	1
1110	1	0	0	1	1	1	1
1111	1	0	0	0	1	1	1

Table 2: Logic minimization

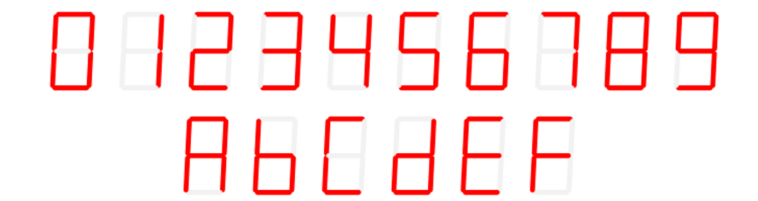

Figure 2: Digits on 7-segment display

3 Logic Minimization

We will do Logic minimization from Table 2, making a K-map for each output column.

Note: In the basys-3 board, the segment conducts if both the anode and cathode are set to 0. In the Logic Minimization of Table 2, we want 0s in place of 1, and 1 in place of 0s, to implement this. So we directly made K-map, considering 1 at those positions, where there are 0 in Table 2.

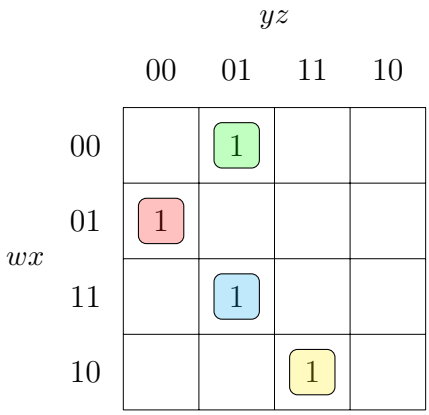

Figure 3: **Segment a = w'x'y'z + w'xy'z' + wxy'z + wx'yz**

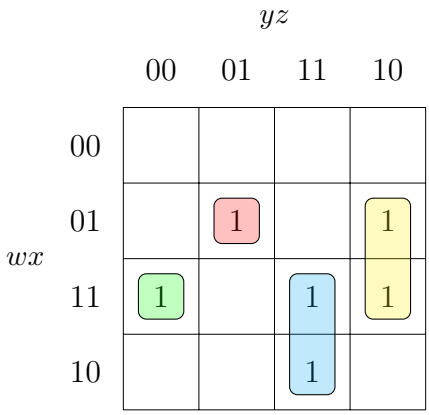

Figure 4: **Segment b = w'xy'z + wxy'z' + xyz' + wyz**

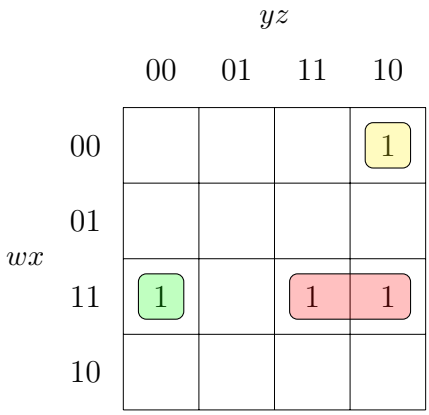

Figure 5: **Segment c= w'x'yz' + wxy'z' + wxy**

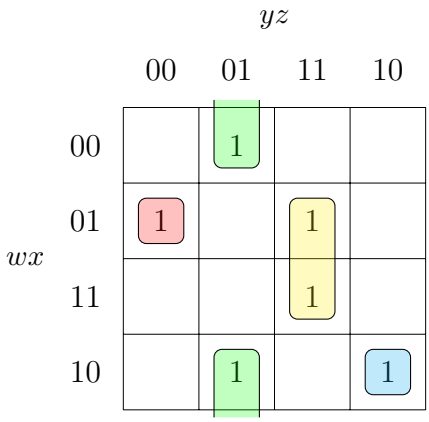

Figure 6: **Segment d = w'x'y'z + w'xy'z' + xyz + wx'yz'**

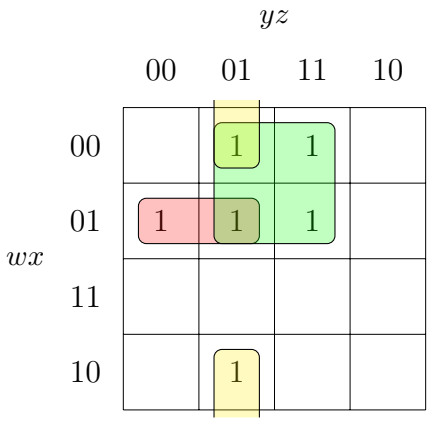

Figure 7: **Segment e = w'z + w'xy' + x'y'z**

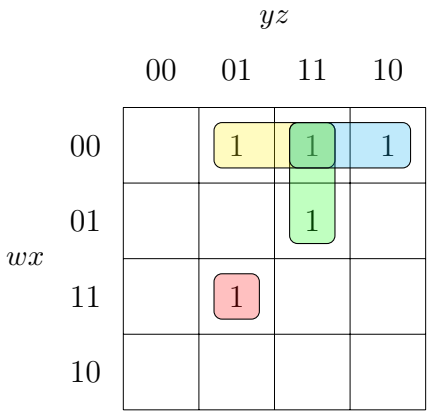

Figure 8: **Segment f= w'x'y + w'x'z + w'yz + wxy'z**

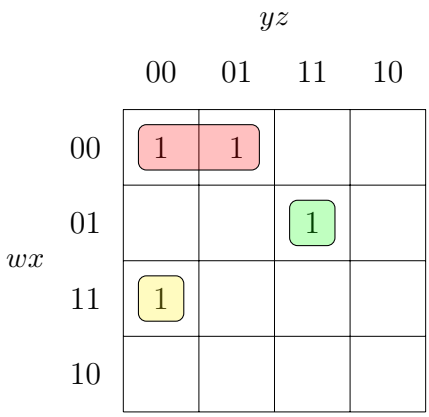

Figure 9: **Segment g = w'x'y' + w'xyz + wxy'z'**

4 Design Decisions

4.1 Part-1: Seven-segment decoder

After logic simplification through K-map, we took 4-bit input from basys-3 board representing a number in binary representation. These 4-bit bits were used to find a, b, c, d, e, f and g, which are then sent to cathodes of the basys-3 board. Output of '0' turns on the segment, and an Output of '1' turns it off.

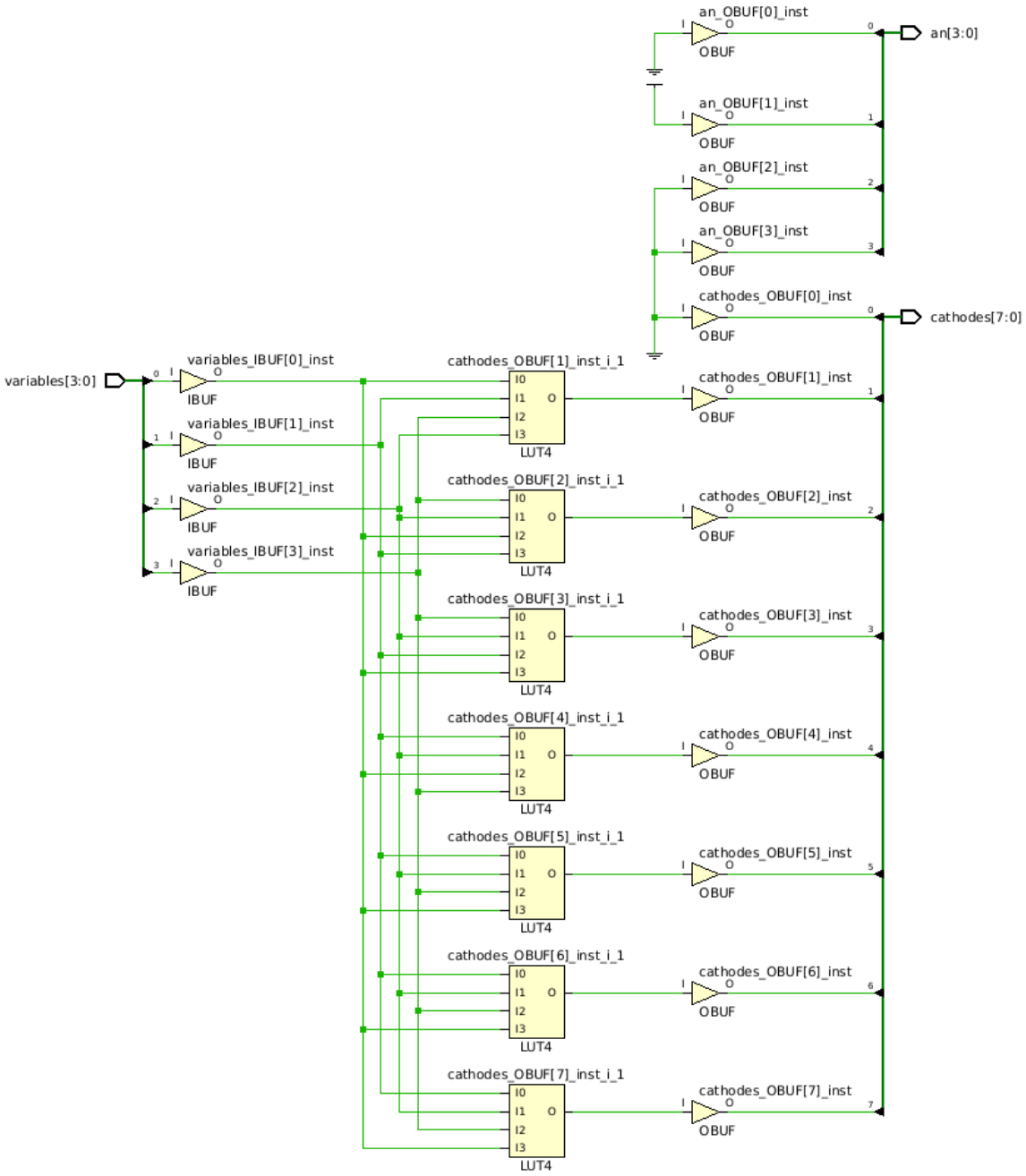

Figure 10: seven-segment-decoder

4.2 Part-2: 4-Display-Seven-segment-decoder

4.2.1 Setting up NEW_CLK

The Basys-3 board has an onboard 100 MHz clock(CLK_IN). We need to reduce this frequency between 4*60Hz to 4*1KHz so that we can see all 4 displays at the same time.
This is because, for 4 anodes, we want a frequency of 60Hz to 1KHz, so each anode will be

on for $\frac{1}{4} * \frac{1}{1000}$s to $\frac{1}{4} * \frac{1}{60}$s. So

$$N = \frac{\frac{1}{4} * \frac{1}{1000}}{\frac{1}{100{,}000{,}000}} = 25,000$$

to

$$N = \frac{\frac{1}{4} * \frac{1}{60}}{\frac{1}{100{,}000{,}000}} = 416,666.67$$

So possible range of N=25,000 to 416,666 We chose N to be equal to 50,000 In this way, by adjusting the speed, we were able to see all the 4 displays at the same time.

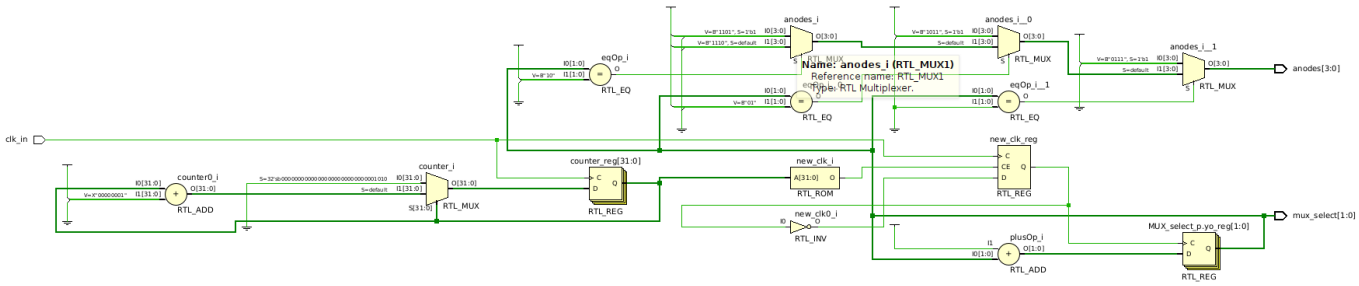

Figure 11: Timing Circuit

4.2.2 Logic behind the Multiplexer

We are taking four 4-bit numbers from a basys-3 board for 4 digits. Since all 4 displays have common cathodes, we need to decide which 4-bit number is to be decoded and sent to cathodes according to the timing block. If the timing block turns on display 1, we will send the first 4-bit number to the decoder; if it turns on display 2, we will send the second 4-bit number to the decoder, and so on. So we need to select a number according to the timing block.
This will be done by 4X1 Multiplexer. It takes an input of 2-bit number from timing-block. These 2 bits will act as select lines for the multiplexer.

Select line	Number selected
00	1
01	2
10	3
11	4

Table 3: Logic minimization

We implemented 4X1 multiplexer, using two 2X1 multiplexer. Each 2X1 multiplexer was implemented using AND, OR and NOT gates.

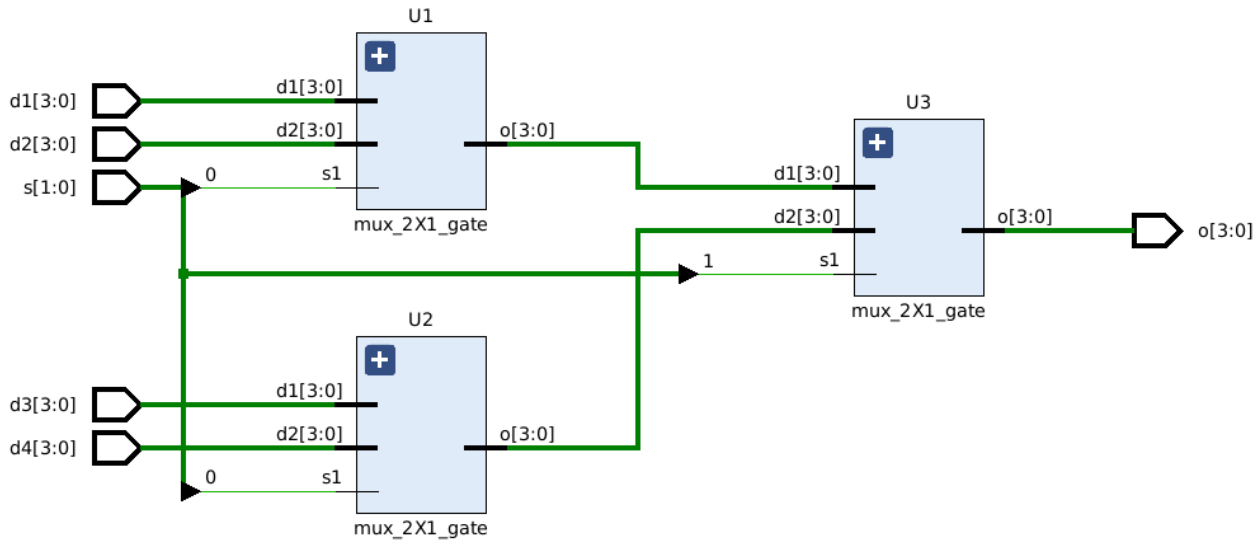

Figure 12: 4X1 Multiplexer

4.2.3 Putting everything together

Finally we combined all three circuits.

One output of the timing circuit was sent to anodes (based on new_clock), and the other output to the multiplexer to select which 4-bit input is to be selected.

Multiplexer takes four 4-bit numbers and a select line(from timing circuit). Based on the select line, it selects one of the numbers and sends it to a seven-segment decoder, which decides on outputs to be sent to the cathode.

Seven-segment decoder uses these 4 bits to decide cathode outputs using the logic provided. This output is sent to basys board to be displayed according to which anode is activated(by timing circuit).

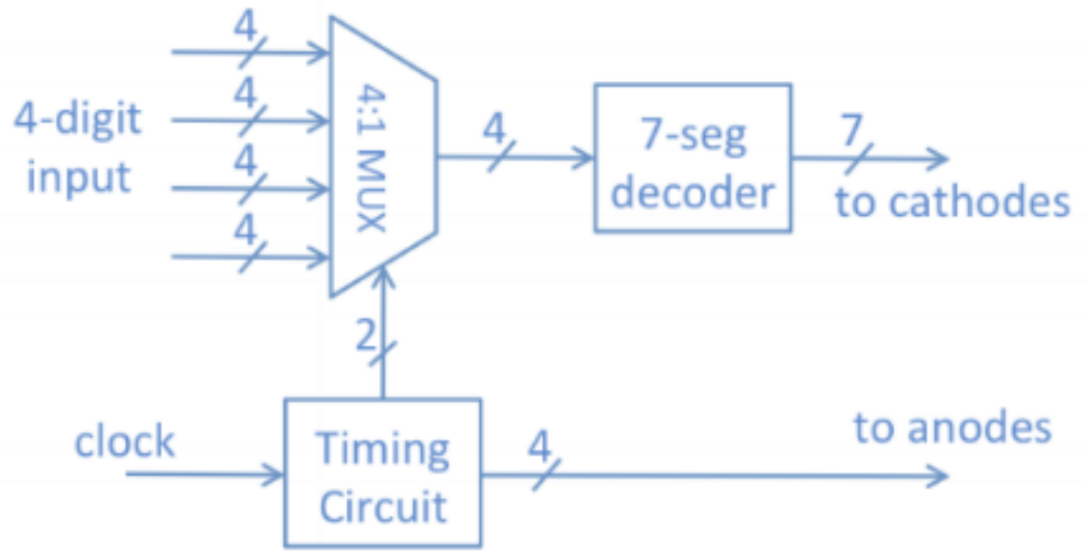

Figure 13: Logic for the final circuit

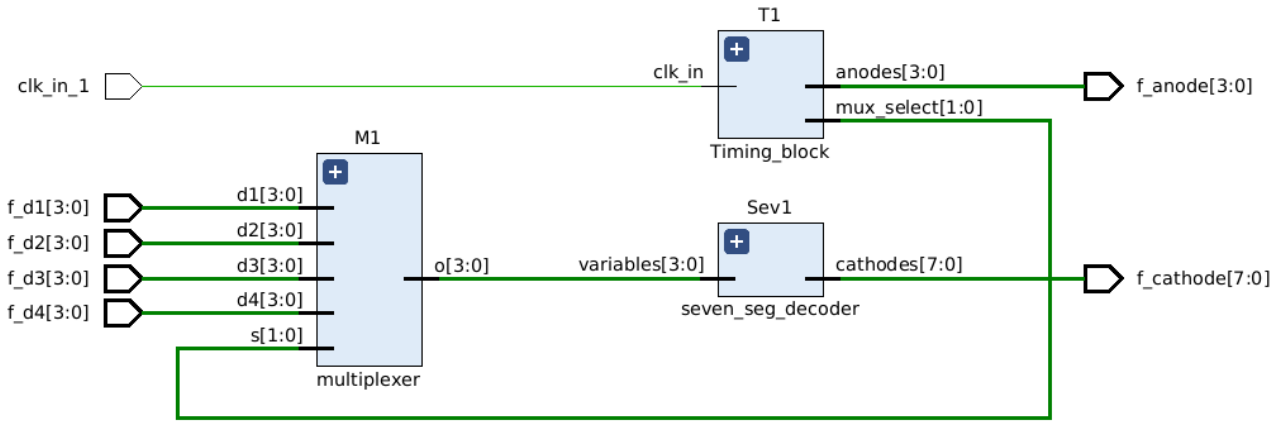

Figure 14: Design Schematic

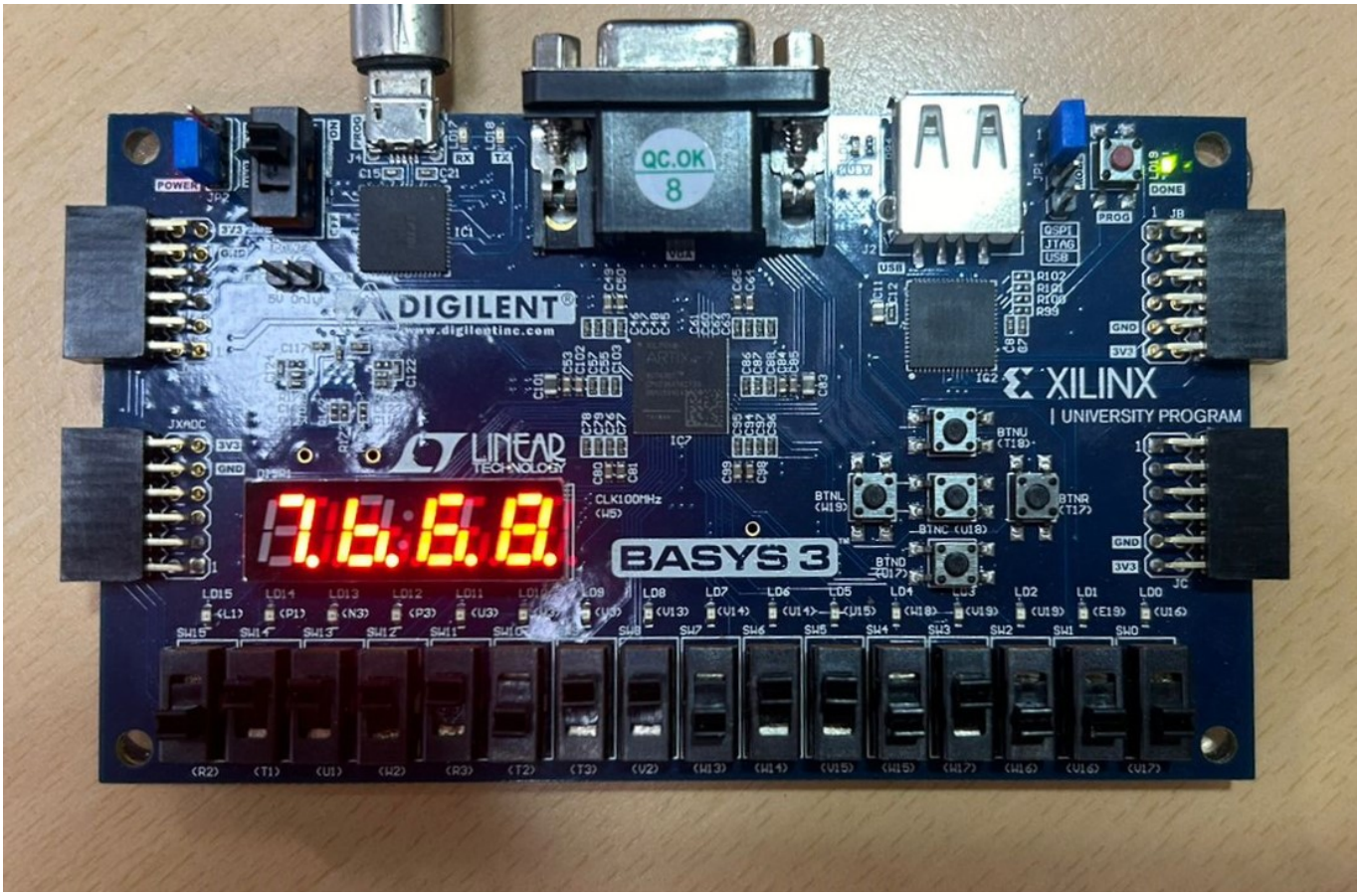

Figure 15: Implementation on Hardware

5 Simulation

5.1 Seven-segment decoder

Figure 16: Simulation of part 1 of assignment

5.2 4-Display-Seven-segment-decoder

5.2.1 Timing_block

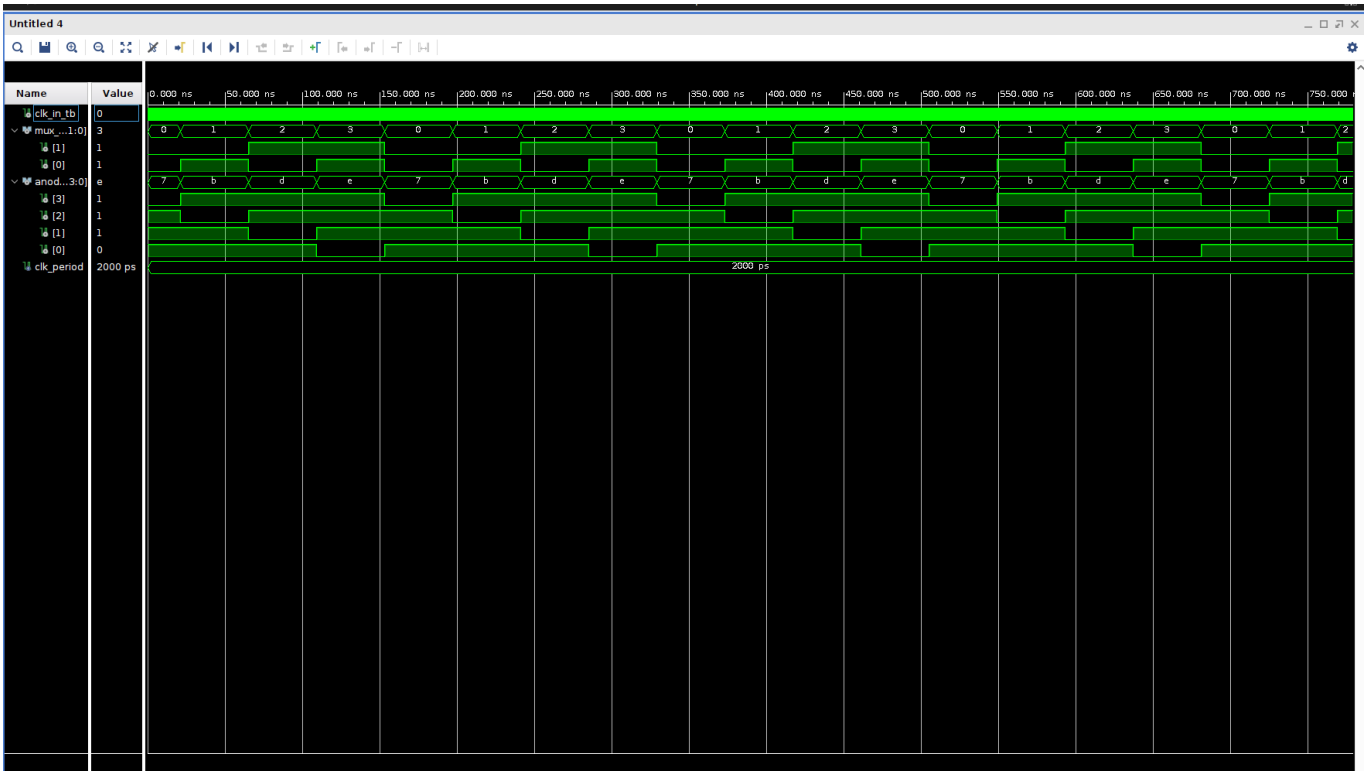

Figure 17: Timing Circuit

5.2.2 Multiplexer

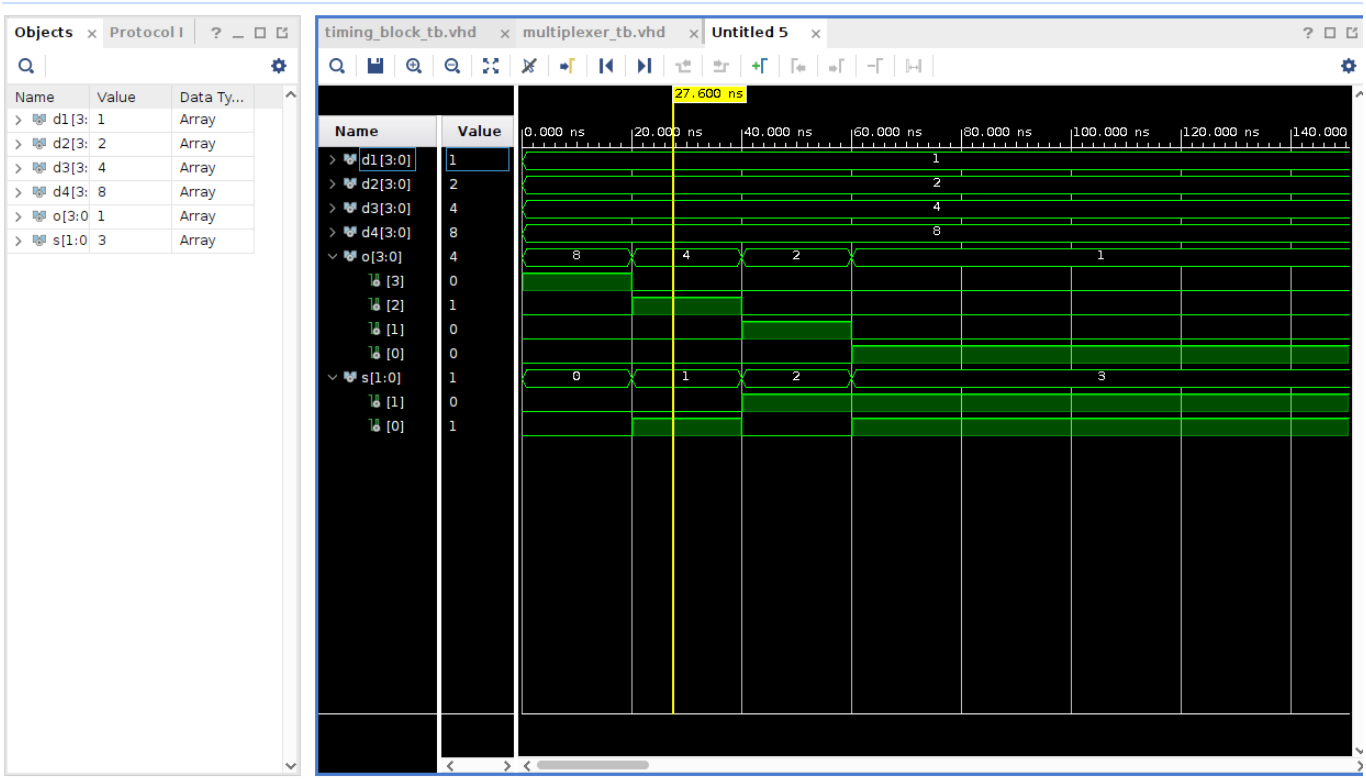

Figure 18: 4X1 Multiplexer

5.2.3 Overall Design

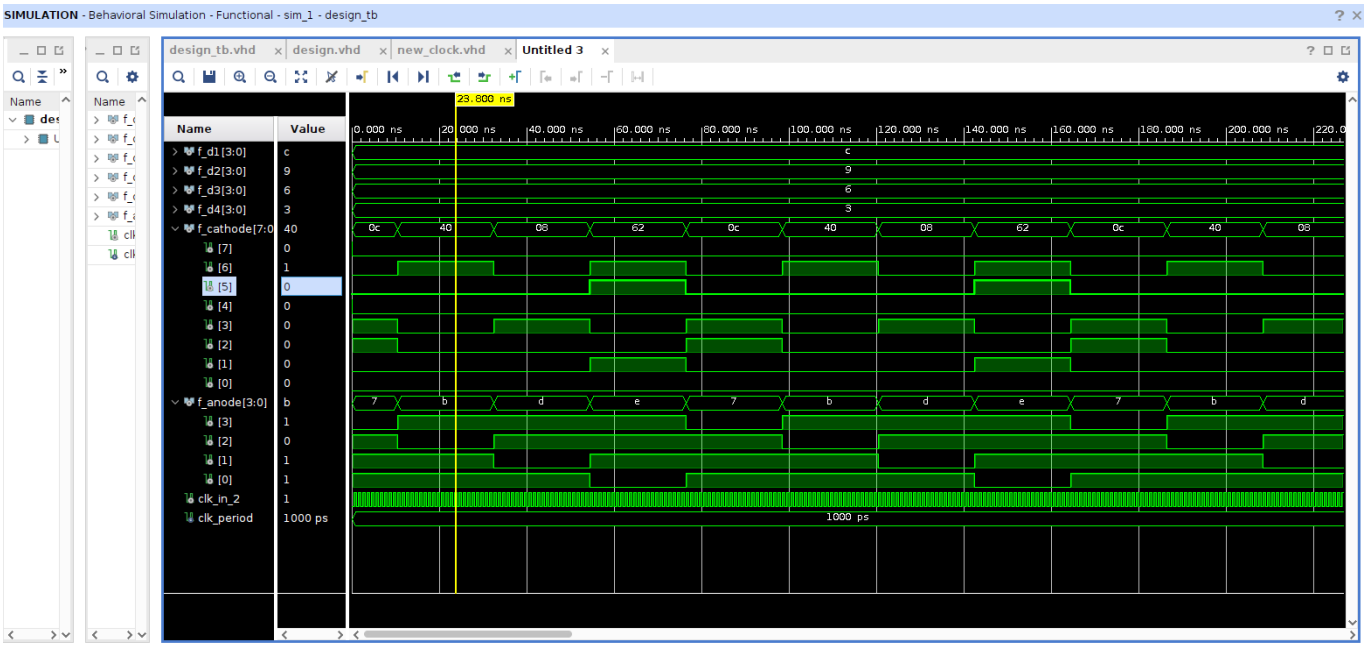

Figure 19: Overall 4-display seven segment

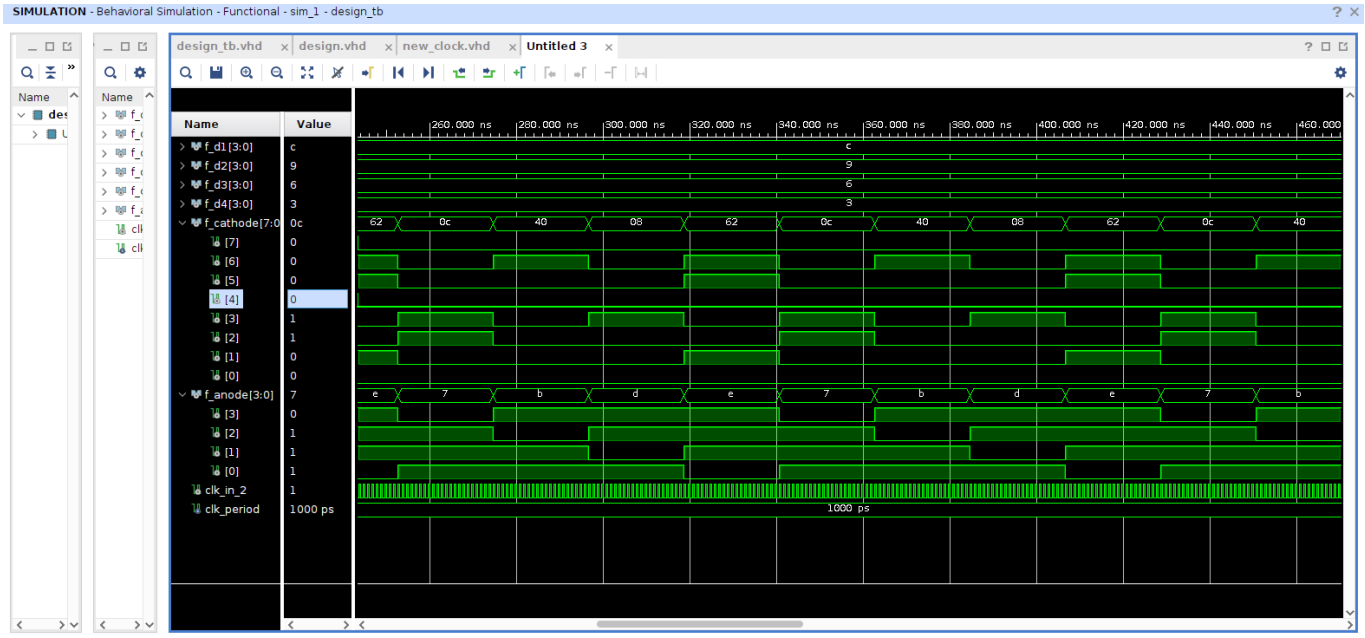

Figure 20: Overall 4-display seven segment

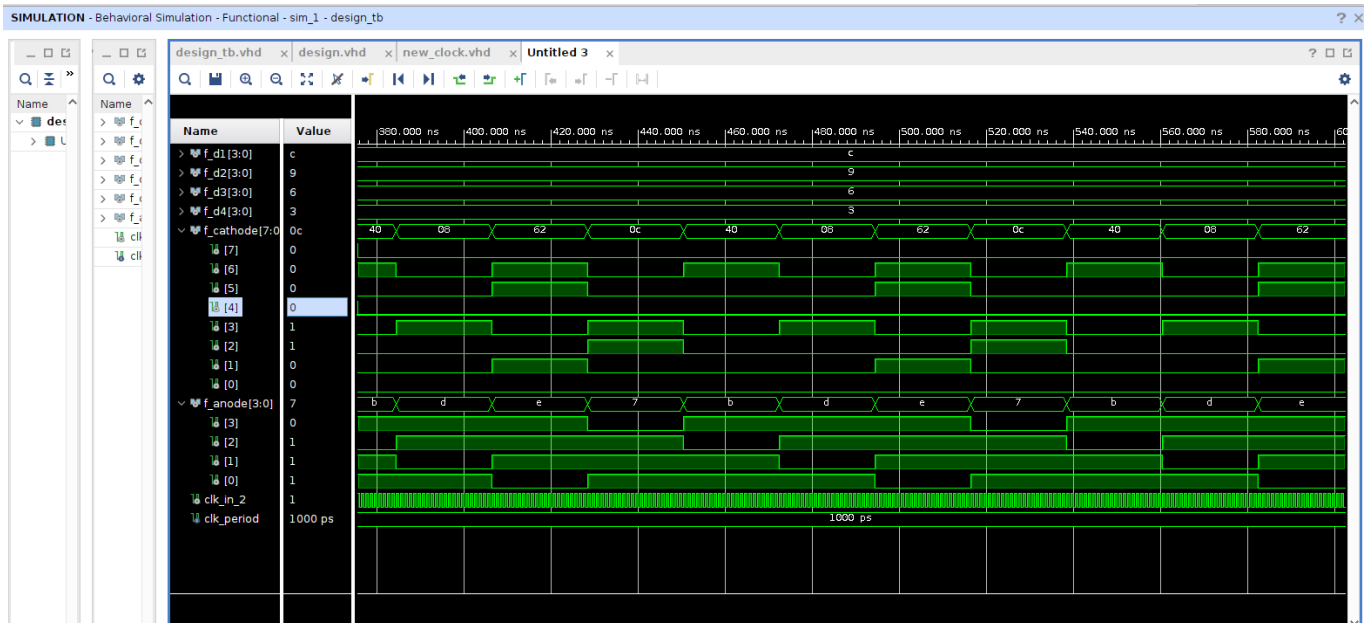

Figure 21: Overall 4-display seven segment

6 Resource Utilization

6.1 Seven-segment decoder

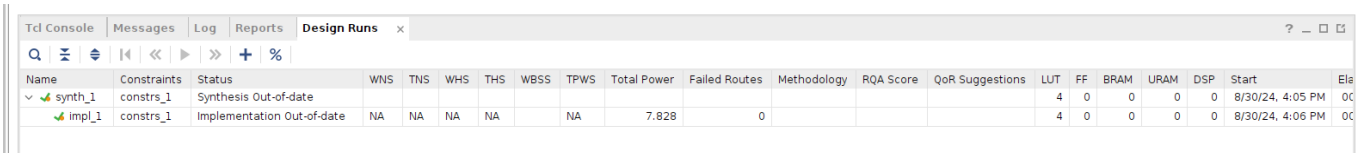

Figure 22: Resource Utilization

6.2 4-Display-Seven-segment-decoder

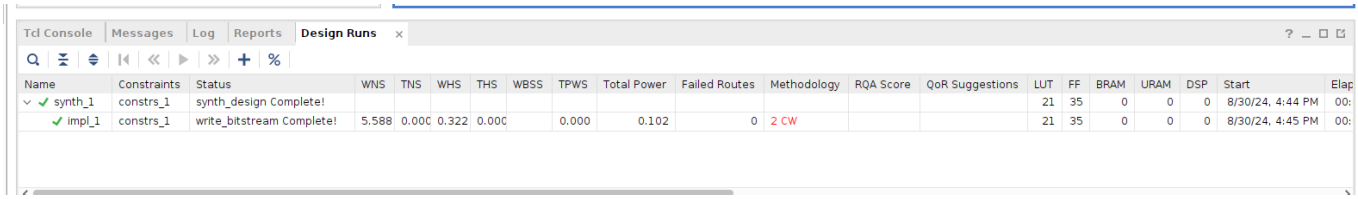

Figure 23: Resource Utilization