HARDWARE ASSIGNMENT 2

Name	Entry no	Group no
Nishant Kumar	2023CS10522	2
Aayushya Sahay	2023CS10543	2

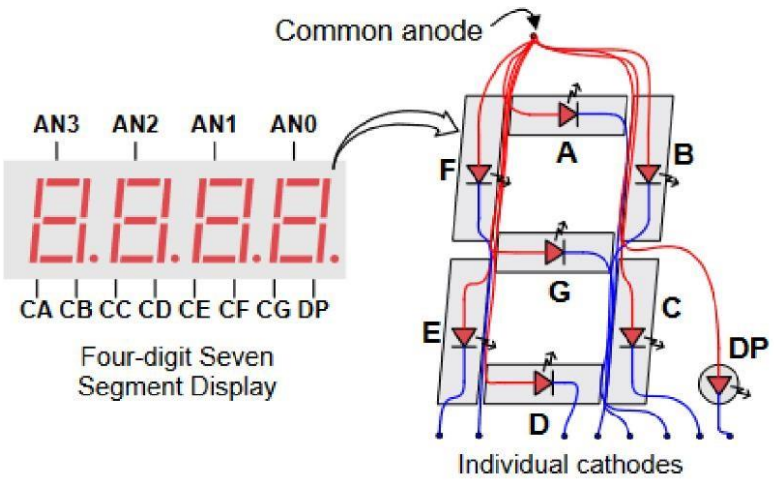
AIM:

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 Basys3 FPGA board.
2. Extend the design to create a circuit that drives all 4 displays for displaying 4 digits together.

DESIGN DECISIONS:

Part-1:

- We took the inputs as 4-bit binary numbers that cover the hexadecimal range by 4 input switches.
- We used lowercase letters to avoid the any ambiguity on the display (B&8; D&0).
- We took care of the ACTIVE-LOW configuration of the basys3 board and programmed the board accordingly.
- A single 7-segment display then works by mapping each of these 7 segments with their derived logic statements (next section).



Part-2:

- The new clock that we made was of 1ms time-period that was enough to not be noticed. We even increased it to 16ms to check the flickering and noticed it very well.
- We built a new 4x1 MUX from the idea that we got in the last Hardware lab to take care of which input to output to the **single-display component** according to a **mux_select** signal (2-bit) which depended on the new clock that we made.
- Lastly, the anodes were left to be programmed which were done using again the new clock and the mux_select signal in the **timing circuit** file. The anodes light up one at a time and go dim when the **reset** button is pressed. The time period of the new clock is 1ms.
- Combining all this, all four displays can be used to show a 4-bit hexadecimal number from four 4-bit inputs using switches.

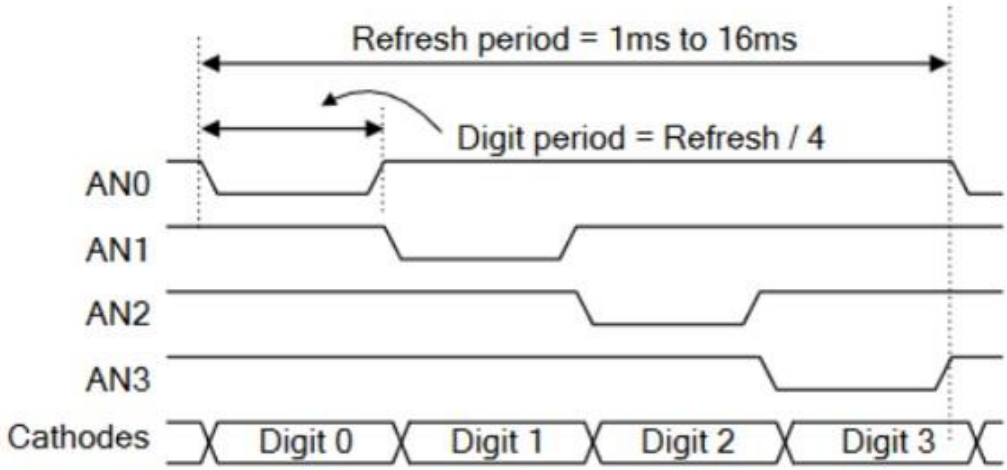


Figure 4: Timing details for 7 seven segment displays

TRUTH TABLE AND LOGIC MINIMIZATION:

The truth table for all the 7 segments according to the 4-bit input is:

Truth Table for 7-Segment Display (Active-Low)

Seg\Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Seg a	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0
Seg b	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	1
Seg c	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1
Seg d	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1
Seg e	0	1	0	1	1	0	0	1	0	1	0	1	0	1	0	0
Seg f	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0
Seg g	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0

The logic was minimized using the k-map technique. Below is the k-map for all segmets.

v0v1\v2v3	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	0	1
10	0	0	0	0

Seg a

v0v1\v2v3	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	1	1	1	1
10	0	0	0	0

Seg b

v0v1\v2v3	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	1	0	1	1
10	0	0	0	0

Seg c

v0v1\v2v3	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	1	0
10	0	0	0	1

Seg d

v0v1\v2v3	00	01	11	10
00	0	1	1	0
01	1	0	1	0
11	0	1	0	0
10	0	1	1	0

Seg e

v0v1\v2v3	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	1	0	0	0
10	0	1	0	0

Seg f

v0v1\v2v3	00	01	11	10
00	1	1	0	0
01	0	0	1	0
11	1	0	0	0
10	0	0	0	0

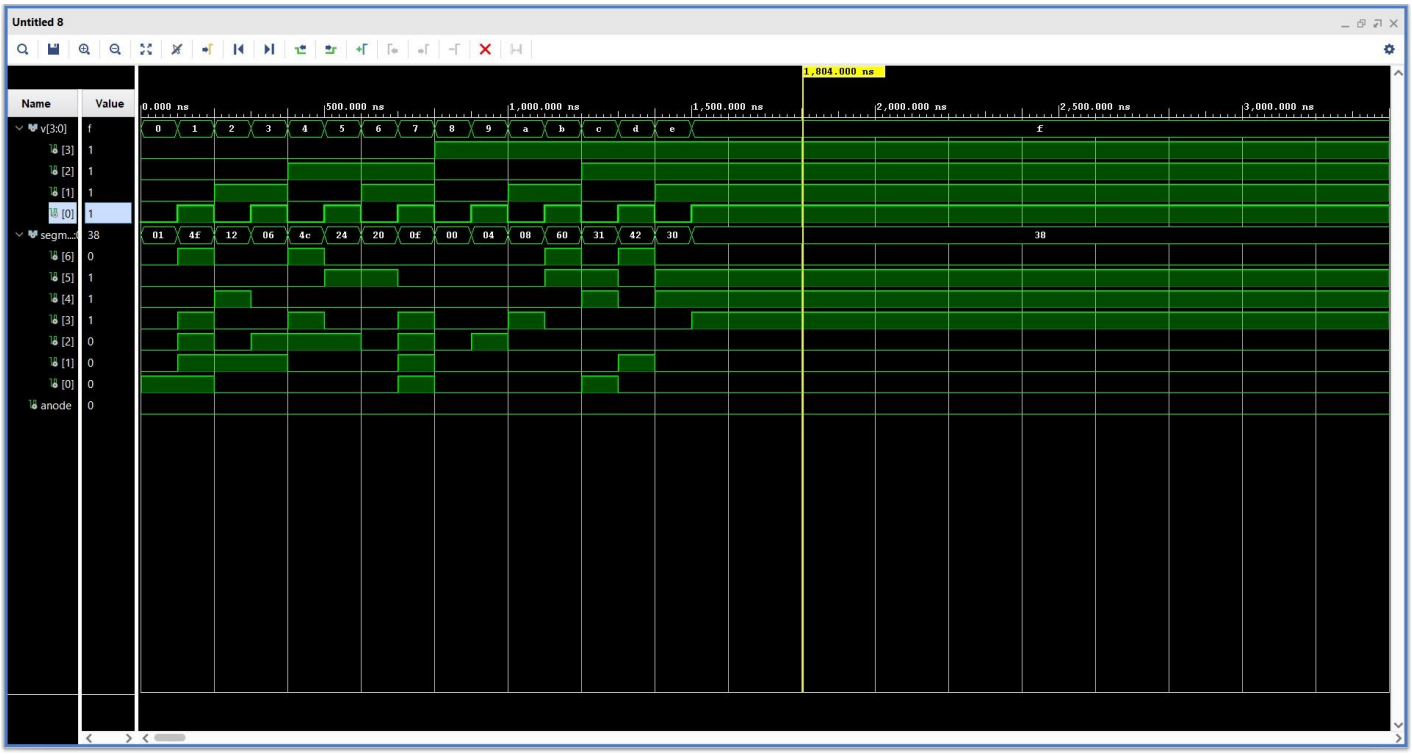
Seg g

Then combining the rectangles (if any) we would minimize the logic for each segment.

The logics statements for each segment are:

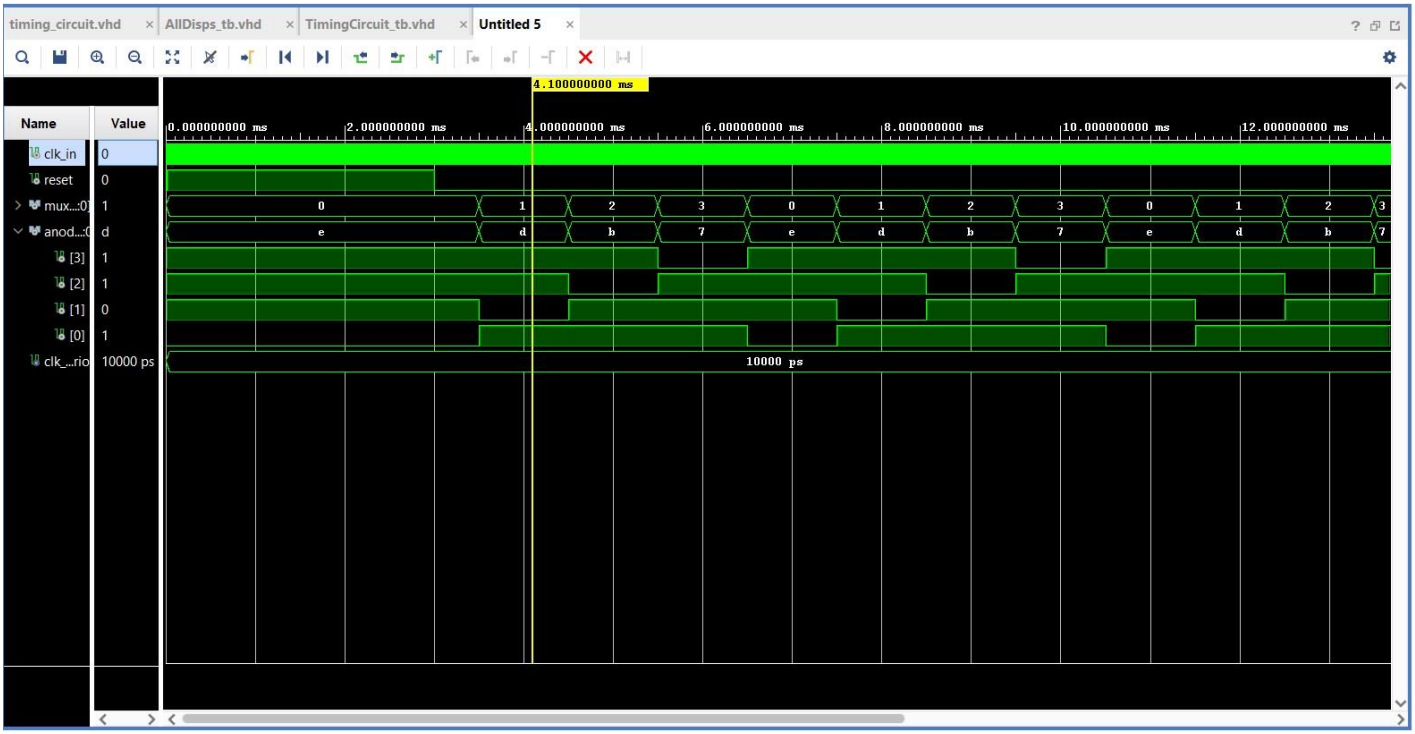
Segment	Logic
Seg a	(not v3 and not v2 and not v1 and v0) or (not v3 and v2 and not v1 and not v0) or (v3 and not v2 and v1 and v0) or (v3 and v2 and not v1 and v0)
Seg b	(not v3and v2 and not v1 and v0) or (v2 and v1and not v0) or (v3 and v2 and not v0) or (v3 and v1 and v0)
Seg c	(not v3 and not v2 and v1 and not v0) or (v3 and v2 and not v0) or (v3 and v2 and v1)
Seg d	(not v3 and not v2 and not v1 and v0) or (not v3 and v2 and not v1 and not v0) or (v3 and not v2 and v1 and not v0) or (v2 and v1 and v0)
Seg e	(not v2 and not v1 and v0) or (not v3 and v2 and not v1) or (not v3 and v0)
Seg f	(v3 and v2 and not v1 and v0) or (not v3 and not v2 and v0) or (not v3 and not v2 and v1) or (not v3 and v1 and v0)
Seg g	(not v3 and v2 and v1 and v0) or (v3 and v2 and not v1 and not v0) or (not v3 and not v2 and not v1)

Simulations



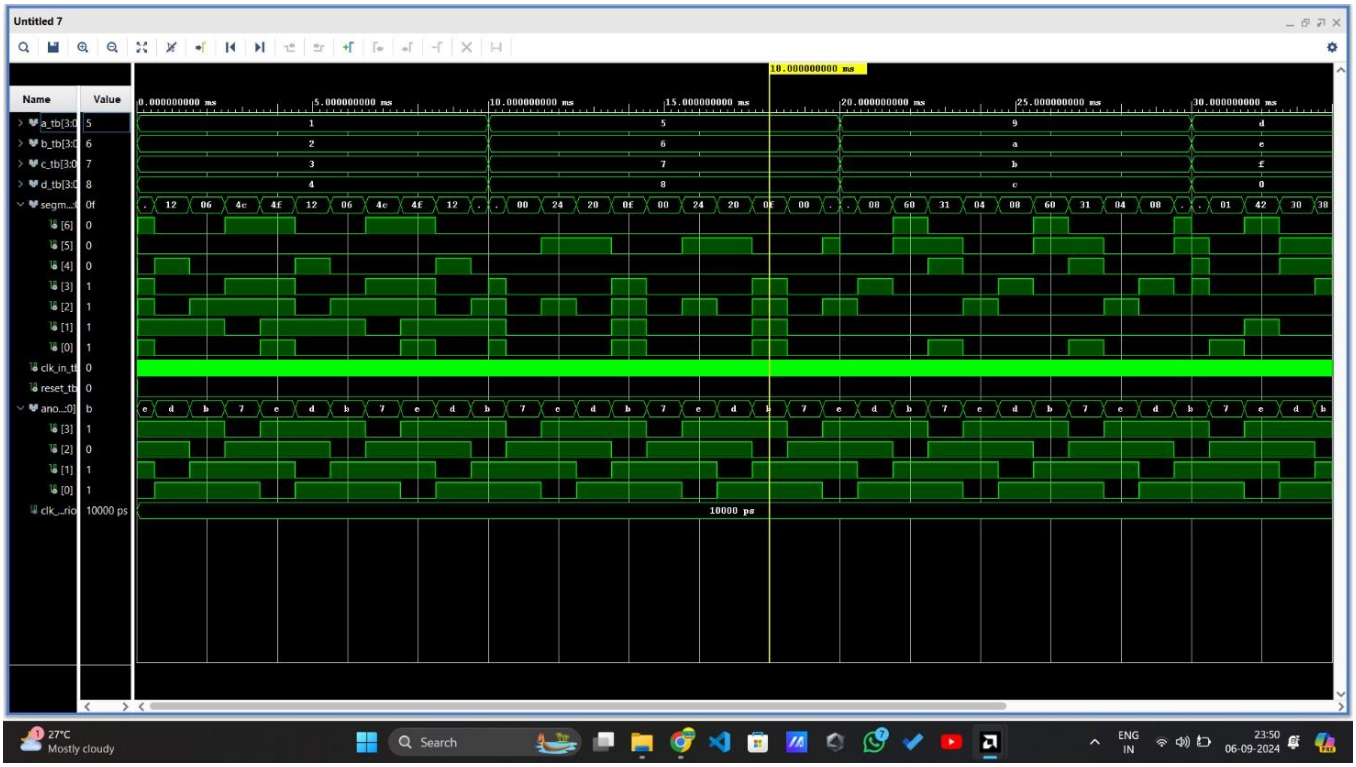
The above simulation is for the single 7-segment display vhd file.

1. The input ‘v[3:0]’ is the input and ‘segment[6:0]’ is the observed output.
2. v[3:0] is a std_logic_vector that is used to take the 4-bit input.
3. segment[6:0] represents the 7 segments i.e. a,b,c,d,e,f, and g, where segement[0] represents ‘g’ and segment[6] represents ‘a’ and others are represented similarly.



The above figure is the simulation of timing circuit.

1. The signal ‘clk_in’ simulates the on-board clock that has frequency 100MHz.
2. The mux_out is a 2 bit vector that changes according to the new clock that has frequency of 1kHz as implemented in the timing circuit.
3. The ‘reset’ button is also passed to this timing circuit to turn off all the anodes and reset the new clock.
4. The anodes is a 4 bit vector that represents the state of each of the 4 anodes (ACTIVE-LOW).
5. The outputs are the mux_out and the anodes vectors.



The above simulation is of the complete implementation of all 4 7-segment displays.

1. The signals ‘a’, ‘b’, ‘c’, and ‘d’ are the inputs to the allDisplays entity.
2. The signal ‘clk_in’ simulates the on-board clock on the basys3 board, and ‘reset’ resets the new clock and turns off all the segments.
3. The ‘segments’ and ‘anodes’ are the outputs which follow from the above two simulations.
4. The active anode keeps changing periodically as displayed in the timing circuit simulation and the segments change accordingly.
5. The output ‘*output*’ is observed.

Resource Utilization Tables:

Single 7-segment display

```
Report Cell Usage:
+------+-----+------+
|      |Cell |Count |
+------+-----+------+
|1     |LUT4 |     7|
|2     |IBUF |     4|
|3     |OBUF |     8|
+------+-----+------+

1. Slice Logic
--------------

+-------------------------+------+-------+-----------+-------+
|        Site Type        | Used | Fixed | Available | Util% |
+-------------------------+------+-------+-----------+-------+
| Slice LUTs              |    4 |     0 |     20800 |  0.02 |
|   LUT as Logic          |    4 |     0 |     20800 |  0.02 |
|   LUT as Memory         |    0 |     0 |      9600 |  0.00 |
| Slice Registers         |    0 |     0 |     41600 |  0.00 |
|   Register as Flip Flop |    0 |     0 |     41600 |  0.00 |
|   Register as Latch     |    0 |     0 |     41600 |  0.00 |
| F7 Muxes                |    0 |     0 |     16300 |  0.00 |
| F8 Muxes                |    0 |     0 |      8150 |  0.00 |
+-------------------------+------+-------+-----------+-------+

2. Memory
---------

+----------------+------+-------+-----------+-------+
|    Site Type   | Used | Fixed | Available | Util% |
+----------------+------+-------+-----------+-------+
| Block RAM Tile |    0 |     0 |        50 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |        50 |  0.00 |
|   RAMB18       |    0 |     0 |       100 |  0.00 |
+----------------+------+-------+-----------+-------+

4. DSP
------

+-----------+------+-------+-----------+-------+
| Site Type | Used | Fixed | Available | Util% |
+-----------+------+-------+-----------+-------+
| DSPs      |    0 |     0 |        90 |  0.00 |
```

```
+-----------+------+-------+-----------+-------+
```

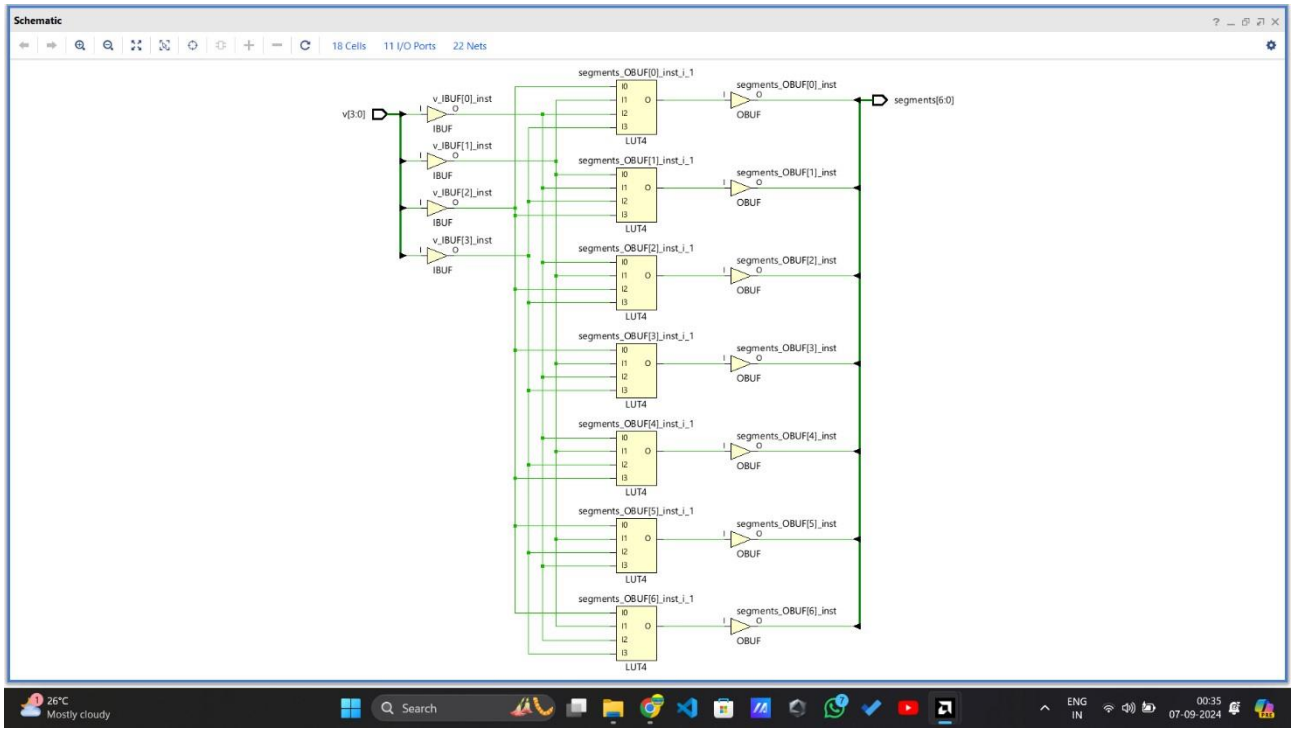
All four 7-segment displays

```
45.   Report Cell Usage:
46.   +------+-------+------+ 47. |      |Cell
|Count | 48. +------+-------+------+ 49. |1     |BUFG
|     1| 50. |2     |CARRY4 |     8| 51. |3     |LUT1
|     2| 52. |4     |LUT2   |     1| 53. |5     |LUT3
|     4| 54. |6     |LUT4   |    11| 55. |7     |LUT5
|    36| 56. |8     |LUT6   |     4| 57. |9     |FDCE
|    33| 58. |10    |FDRE   |     2| 59. |11    |IBUF
|    18| 60. |12    |OBUF   |    11|
61. +------+-------+------+
62.
63.1. Slice Logic
64.--------------
65.
66. +-------------------------+------+-------+-----------+-----------+-------
+ 67. |        Site Type        | Used | Fixed | Prohibited | Available | Util%
|  68. +-------------------------+------+-------+-----------+-----------+-----
--+ 69. | Slice LUTs*             |   52 |     0 |          0 |     20800 |
0.25 | 70. |   LUT as Logic          |   52 |     0 |          0 |     20800 |
0.25 | 71. |   LUT as Memory         |    0 |     0 |          0 |      9600 |
0.00 | 72. | Slice Registers         |   35 |     0 |          0 |     41600 |
0.08 | 73. |   Register as Flip Flop |   35 |     0 |          0 |     41600 |
0.08 | 74. |   Register as Latch     |    0 |     0 |          0 |     41600 |
0.00 | 75. | F7 Muxes                |    0 |     0 |          0 |     16300 |
0.00 | 76. | F8 Muxes                |    0 |     0 |          0 |      8150 |
0.00 |
77. +-------------------------+------+-------+-----------+-----------+-------+
78.
79.2. Memory
80.---------
81.
82. +----------------+------+-------+-----------+-----------+-------+
83. |    Site Type   | Used | Fixed | Prohibited | Available | Util% |
84. +----------------+------+-------+-----------+-----------+-------+
85. | Block RAM Tile |    0 |     0 |          0 |        50 |  0.00 |
86. |   RAMB36/FIFO* |    0 |     0 |          0 |        50 |  0.00 |
87. |   RAMB18       |    0 |     0 |          0 |       100 |  0.00 |
88.   +----------------+------+-------+---
---------+-----------+-------+
89.   90.
91.
```

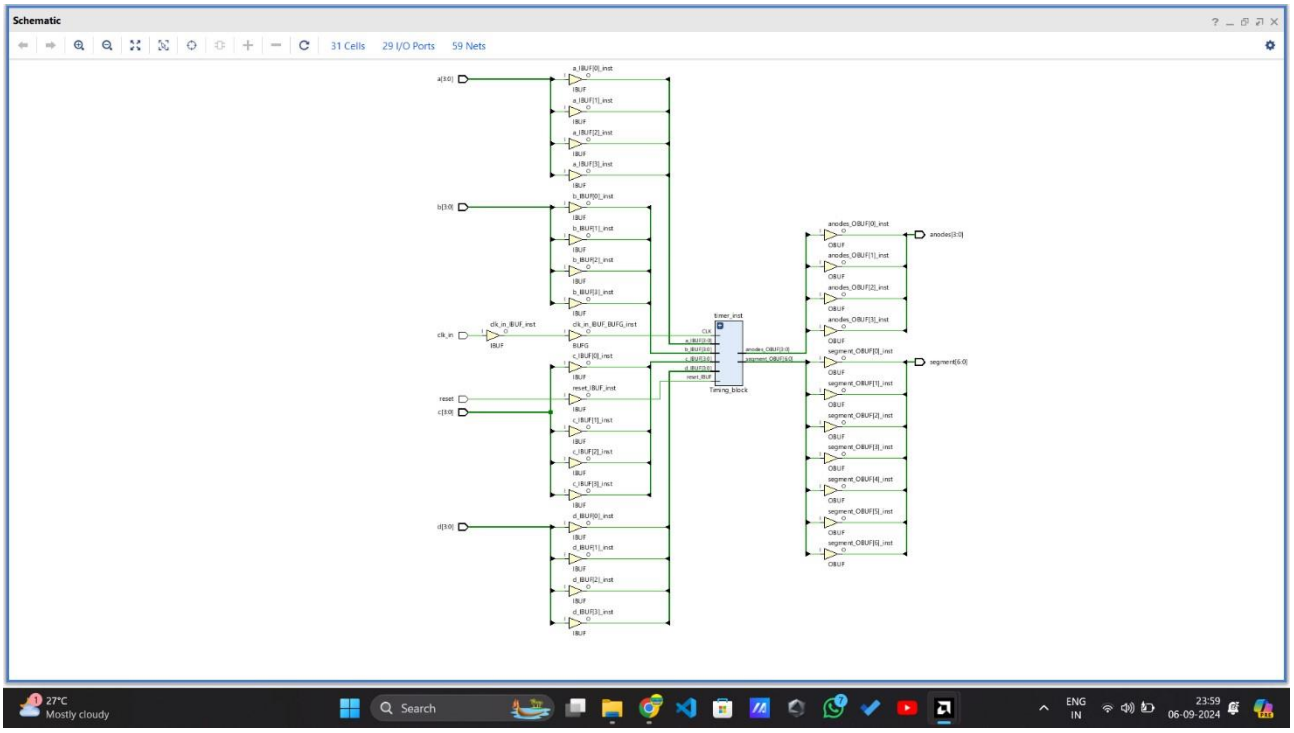
```
3. DSP
------

+------------+------+-------+------------+-----------+-------+
| Site Type | Used | Fixed | Prohibited | Available | Util% |
+------------+------+-------+------------+-----------+-------+
| DSPs      |    0 |     0 |          0 |        90 |  0.00 |
+------------+------+-------+------------+-----------+-------+
```

Schematics:



Schematic for single 7-segment display



Schematic for all four 7-segment displays