COL215

Hardware Assignment 2

Khushi Mishra

2022TT12159

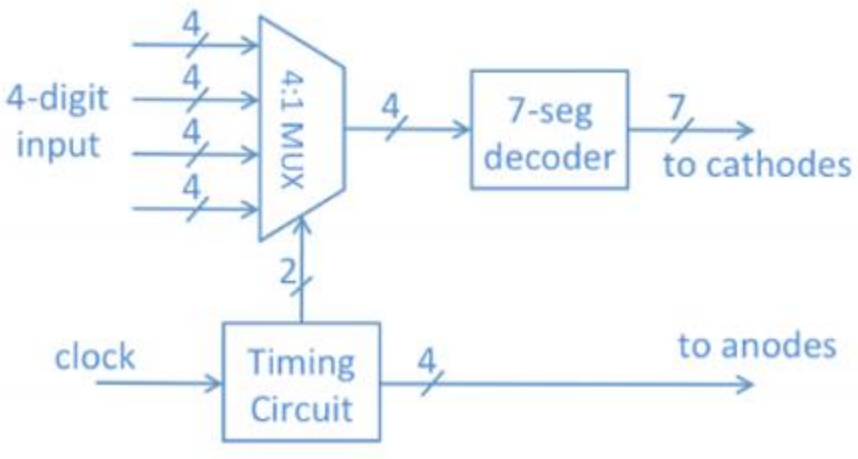
Lakshya Batra

2022TT12163

Introduction

Designing and implementing a circuit that displays a hexadecimal number on the 4 seven-segment display after taking the number as input from four switches on the Basys3 board (a four-bit number). Extending this logic to take a 16-bit input and display all 4 digits on the board, with the help of a timer circuit and a 4X1 MUX gate.

Overall, Logic:



Part 1: Seven Segment Encoder Implementation

The seven-segment encoder consists of 4-bit input, and a seven-bit output, one corresponding to each cathode. Since we wanted an ACTIVE-LOW implementation, a NOT gate is put at start of the logic of each out bit.

The K-map for each output bit is shown below:

IN_BITS: W, X, Y, Z

OUT_BITS: a,b,c,d,e,f,g

Truth Table:

No.	w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	1	1	1	0	1	1	1
B	1	0	1	1	0	0	1	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	0
D	1	1	0	1	0	1	1	1	1	0	1
E	1	1	1	0	1	0	0	1	1	1	1
F	1	1	1	1	1	0	0	0	1	1	1

K-Map:

- a: X'Z' + W'Y + XY + WZ' + W'XZ + WX'Y'

YZ

WX

	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	1	1	1	0
10	1	1	1	1

- b: W'X' + X'Z' + W'Y'Z' + W'YZ + WY'Z

YZ

WX

	00	01	11	10
00	1	1	0	1
01	1	0	1	1
11	1	1	0	0
10	1	0	0	1

- c: W'Z' + W'Z + Y'Z + W'X + WX'

		YZ			
		00	01	11	10
WX	00	1	1	0	1
	01	1	1	1	1
	11	1	1	0	1
	10	0	1	0	1

- d: WY' + W'X'Z' + X'YZ + XY'Z + XYZ'

		YZ			
		00	01	11	10
WX	00	1	1	0	1
	01	1	1	1	1
	11	1	1	0	1
	10	0	1	0	1

e: X'Z' + YZ' + WY + WX

WX \ YZ	00	01	11	10
00	1	0	1	1
01	0	0	1	0
11	0	0	1	1
10	1	1	1	1

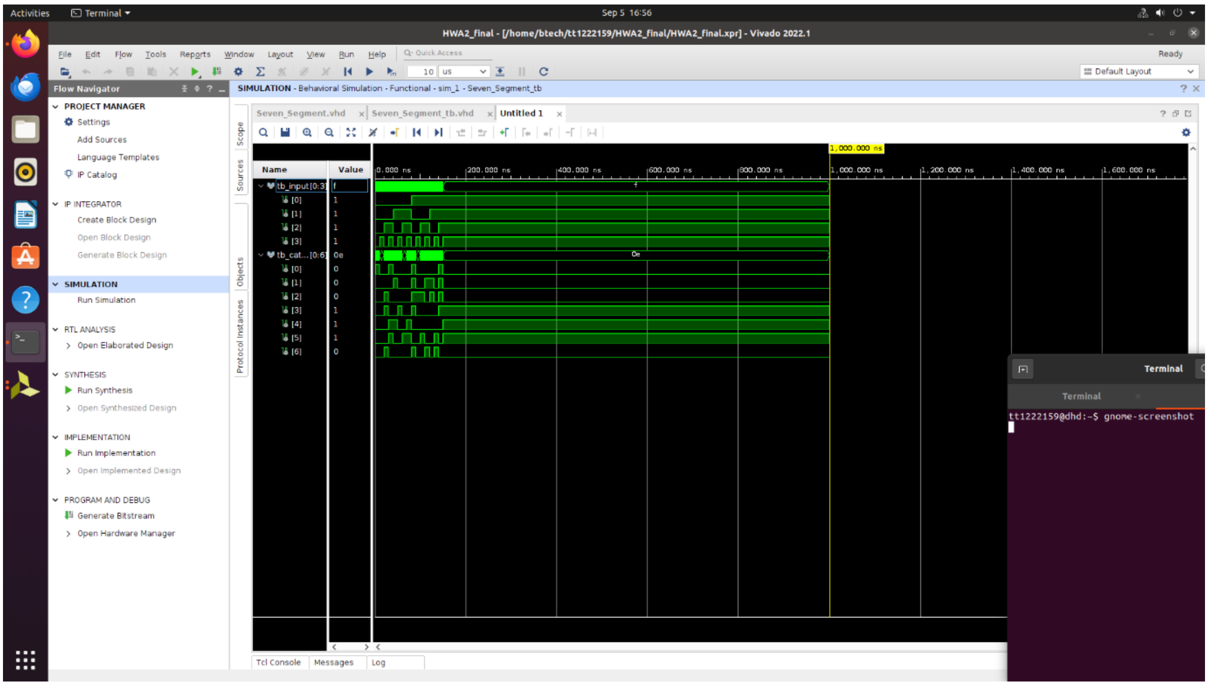
- f: Y'Z' + XZ' + WX' + WY + W'XY'

WX \ YZ	00	01	11	10
00	1	1	1	1
01	0	1	0	1
11	0	0	1	1
10	0	1	1	1

- g: X'Y + YZ' + WX' + WZ + W'XY'

WX \ YZ	00	01	11	10
00	0	1	0	1
01	0	1	1	1
11	1	0	1	1
10	1	1	1	1

Simulation Snapshot (Seven Segment Encoder):



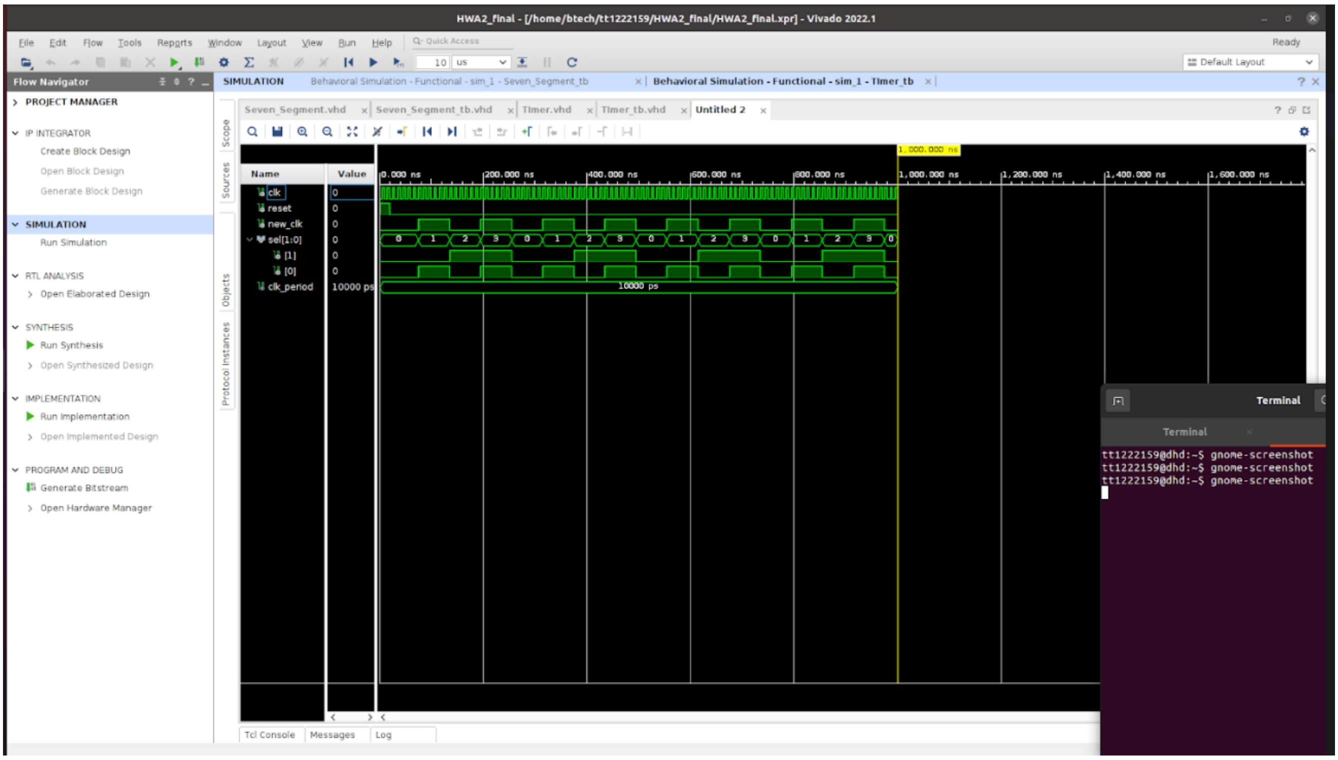
Part B: Timer Circuit Implementation

The Basys3 board has a default frequency of 100Mz; To display a separate digit on each LED display, the corresponding anode signal needs to be activated in a cyclic manner. To avoid flickering, refresh rate should vary between 1kHz - 60Hz (1-16 ms period). The Timer entity is built for this logic.

Inputs: The default clk, and a reset signal.

Outputs: A new_clk of decreased frequency, and a ‘sel’ signal for anode selection (ranges from 0 to 3, in a cyclic manner)

Simulation Snapshot:

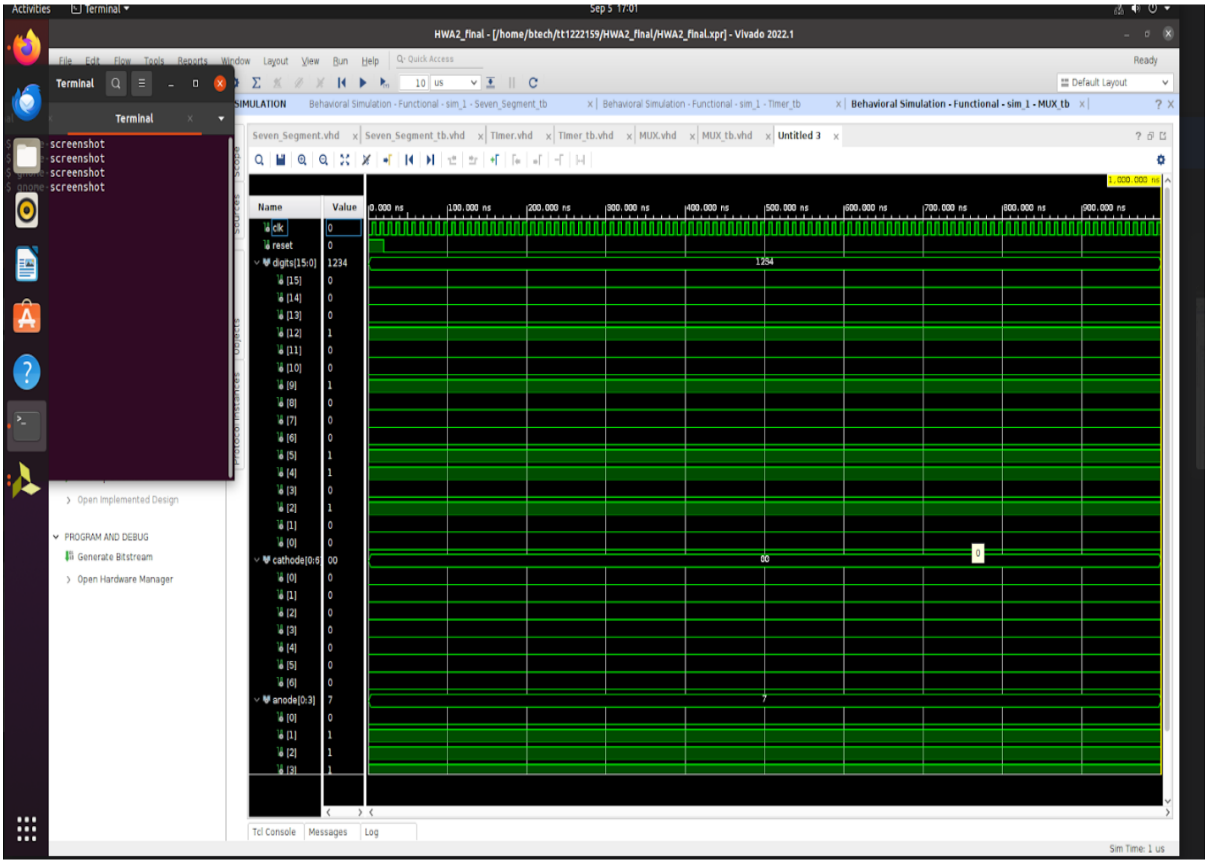


Part 3: MUX gate implementation (Top level Entity)

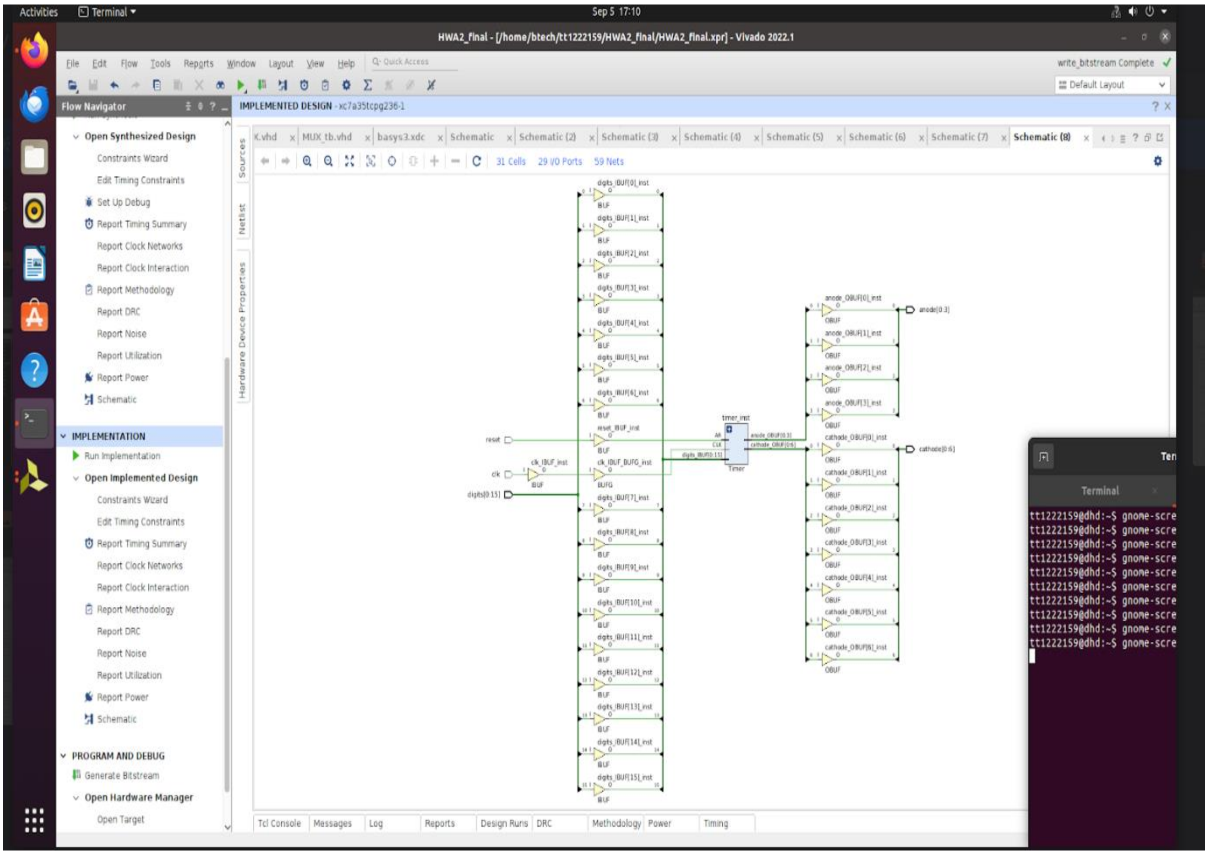
Here, a Multiplexer module is implemented with four 4-bit inputs from slider switches and output going to 7-segment module. This is the top module of our implementation.

Inputs: It takes the new_clk, sel signal (from timer circuit) and a 16-bit input (from the slider switches), and maps each output digit correspondingly according to the MUX logic.

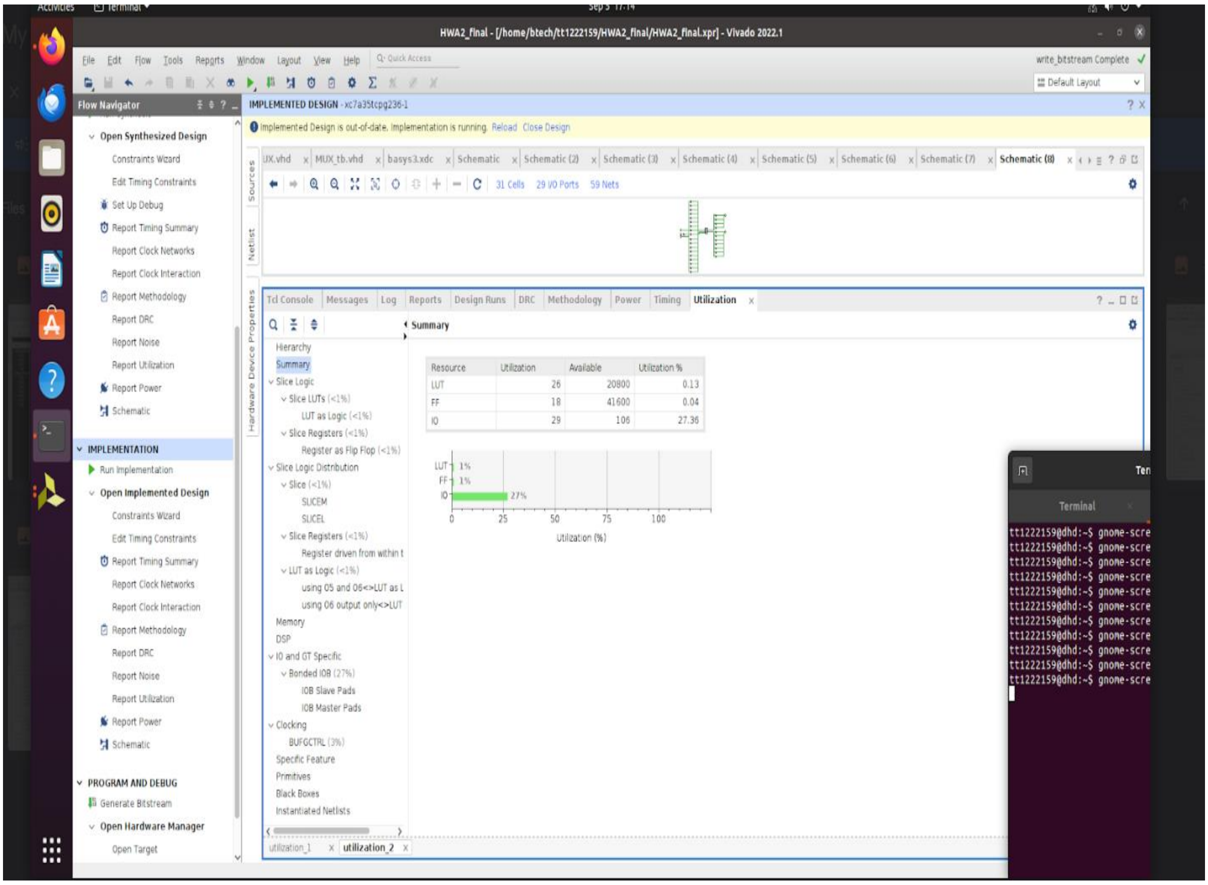
Simulation Snapshot:



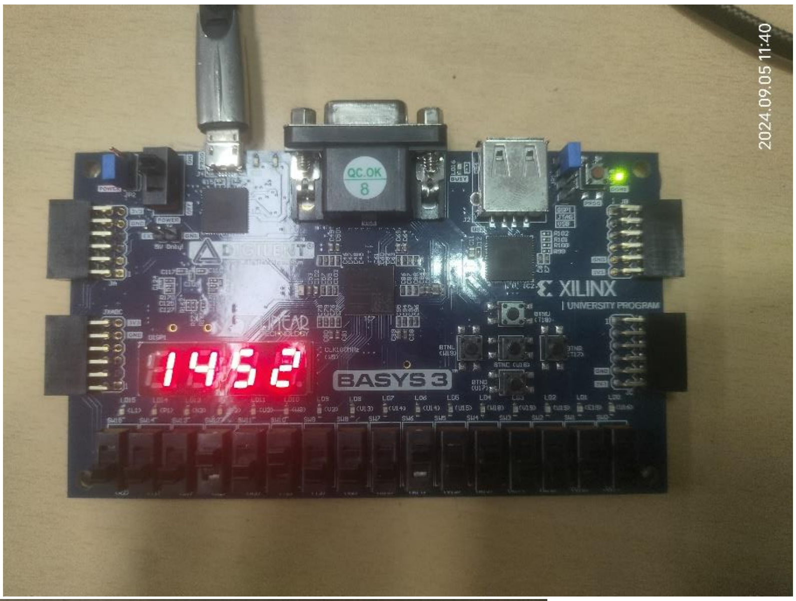
Schematic Snapshot:

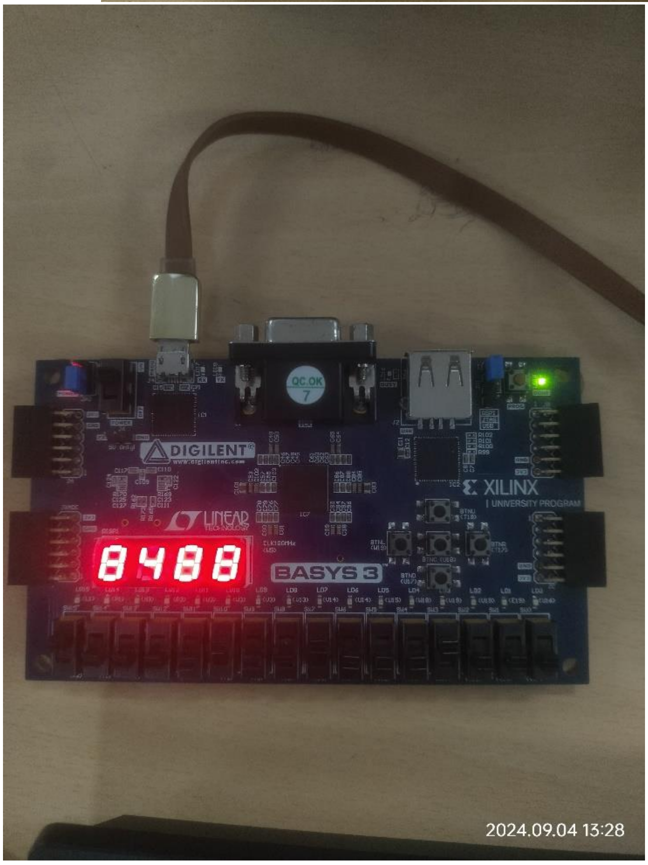


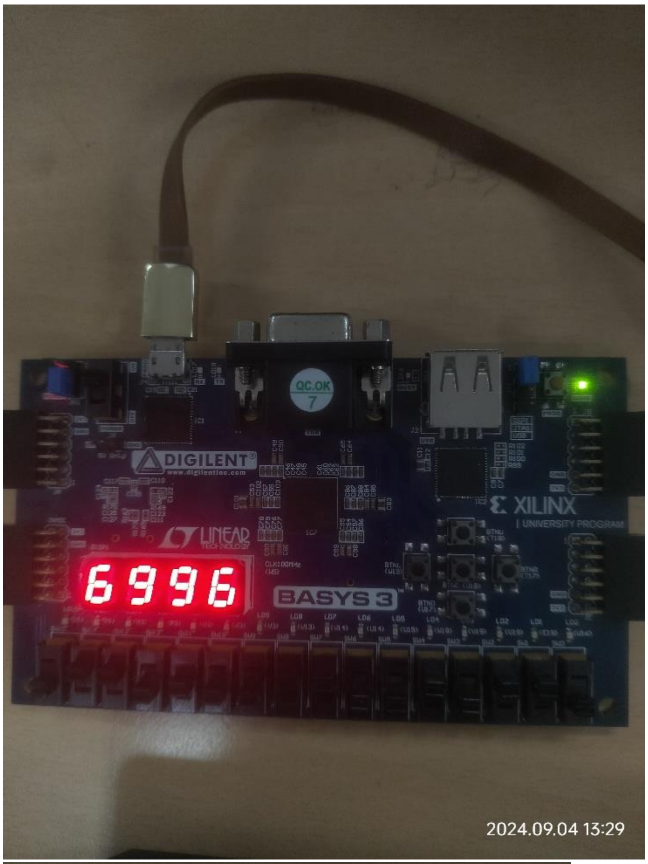
Resource Utilisation Table

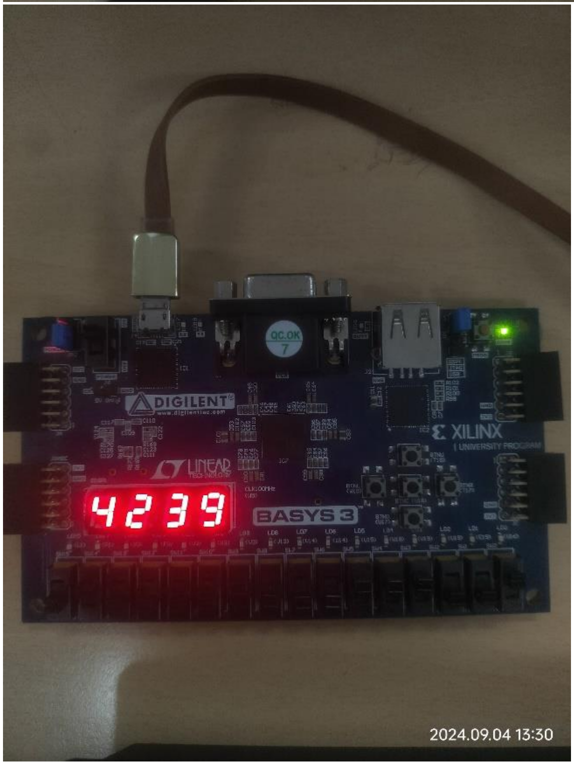


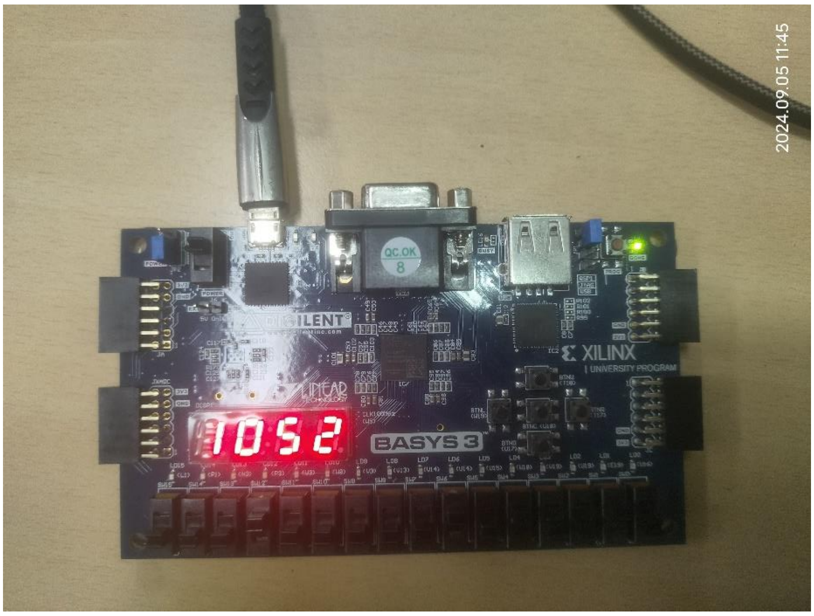
Hardware Snapshots:


2024.09.05 11:40
QC.OK
8
XILINX
UNIVERSITY PROGRAM
1452
BASYS 3


QC.OK
7
DIGILENT
XILINX
UNIVERSITY PROGRAM
8488
BASYS 3
2024.09.04 13:28


DIGILENT
6996
BASYS 3
XILINX
2024.09.04 13:29


DIGILENT
4239
BASYS 3
XILINX
2024.09.04 13:30



Conclusion:

We were able to implement the logic successfully, taking a 16 –bit input from the basys3 board, and displaying the corresponding numbers on the LEDs.