YASH SHINDEKAR 2023CS10592 AYUSH PANDEY 2023CS10597

COL215 Hardware Assignment 2 Report

PART – 1 (7 SEGMENT DECODER)

Introduction

This assignment involves designing and implementing a combinational circuit that takes a 4-digit decimal/hexadecimal number from switches on the Basys3 board and displays it on the four 7-segment displays on the board.

Problem Description

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.

Definitions and Theory

A seven-segment display is a form of electronic display device used for displaying decimal numerals that is an alternative to the more complex dot-matrix displays. The 7 segments consist of seven LEDs (hence 'seven-segment') arranged in a rectangular fashion. Each segment can be lit independently of the others to display the digits 0-9 and some alphabets.

Key Components:

1. **7-Segment Display**:
    - Seven individual segments (labeled a, b, c, d, e, f, g) that can be lit in various combinations to form numerical digits or specific characters.
    - Each segment is controlled individually by applying a voltage.
2. **Binary Input**:
    - Typically a 4-bit binary input (ranging from 0000 to 1001 for digits 0 to 9).
    - The decoder circuit takes this binary input and controls the segments accordingly.
3. **Decoder Logic**:

- The 7-segment decoder contains combinational logic circuits that map the binary input to the correct combination of segments.
- For instance, to display the digit "3," the segments a, b, c, d, and g would be lit.

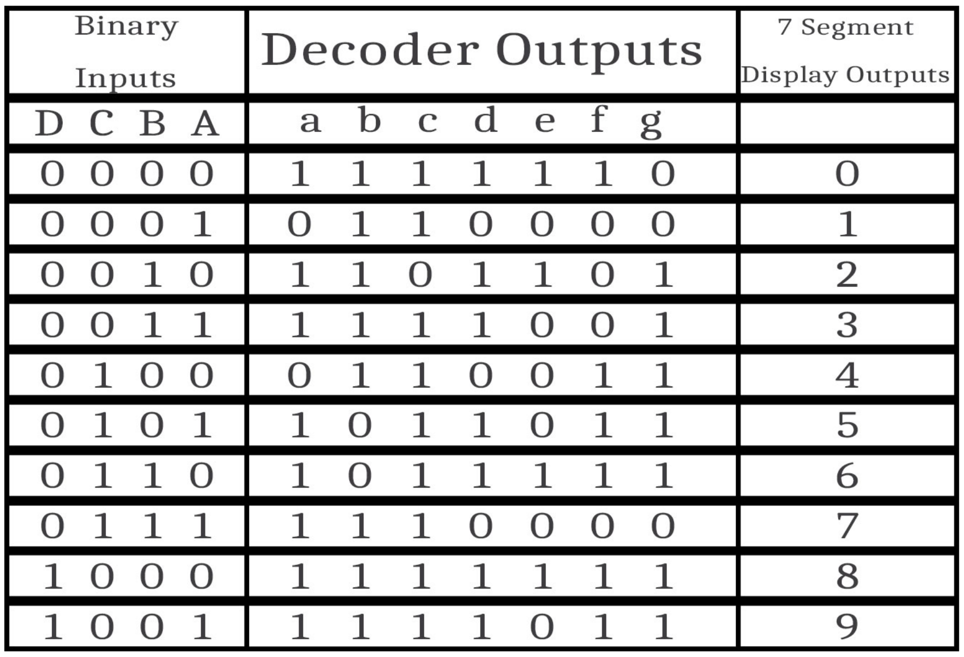
Functionality:

- **BCD to 7-Segment Decoder**: A common type of 7-segment decoder is the **Binary Coded Decimal (BCD) to 7-segment decoder**, which takes a 4-bit BCD input (0000 to 1001) and drives the 7-segment display accordingly.
- **Logic Implementation**: Each segment's state (on/off) is determined by a set of logic expressions based on the input binary code. For example, to display the digit "1," only segments b and c should be lit.

Common Applications:

- **Digital Clocks**
- **Calculators**
- **Counters**
- **Embedded Systems** for displaying numerical outputs

Truth Table:



Binary Inputs				Decoder Outputs							7 Segment Display Outputs
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

7 Segment Display Boolean expression of each output functions can be written as:-

a = F1 (A, B, C, D) = ∑m (0, 2, 3, 5, 7, 8, 9)

b = F2 (A, B, C, D) = ∑m (0, 1, 2, 3, 4, 7, 8, 9)

c = F3 (A, B, C, D) = ∑m (0, 1, 3, 4, 5, 6, 7, 8, 9)

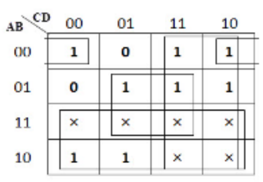
d = F4 (A, B, C, D) = ∑m (0, 2, 3, 5, 6, 8)

e = F5 (A, B, C, D) = ∑m (0, 2, 6, 8)

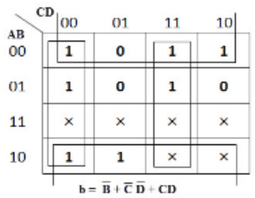
f = F6 (A, B, C, D) = ∑m (0, 4, 5, 6, 8, 9)

g = F7 (A, B, C, D) = ∑m (2, 3, 4, 5, 6, 8, 9)

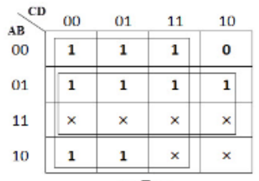
The below figures shows the 7 Segment K map simplification for the common cathode seven-segment decoder in order to design the combinational circuit.



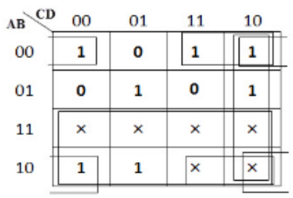
$a = A + C + BD + \overline{BD}$



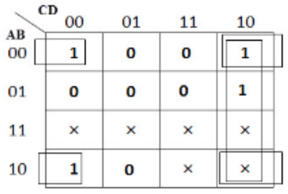
$b = \overline{B} + \overline{C}\,\overline{D} + CD$



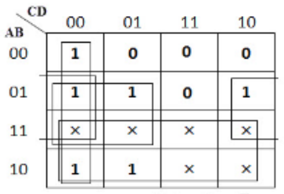
$c = B + \overline{C} + D$



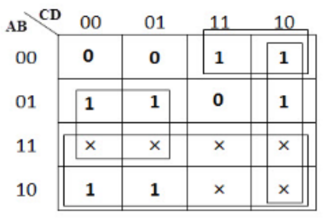
$d = \overline{B}\,\overline{D} + C\overline{D} + B\overline{C}D + \overline{B}C + A$



$e = \overline{B}\,\overline{D} + C\overline{D}$



$f = A + \overline{C}\,\overline{D} + B\overline{C} + B\overline{D}$



$g = \overline{B}C + C\overline{D} + B\overline{C} + B\overline{C} + A$

LOGIC GATE DIAGRAM :-

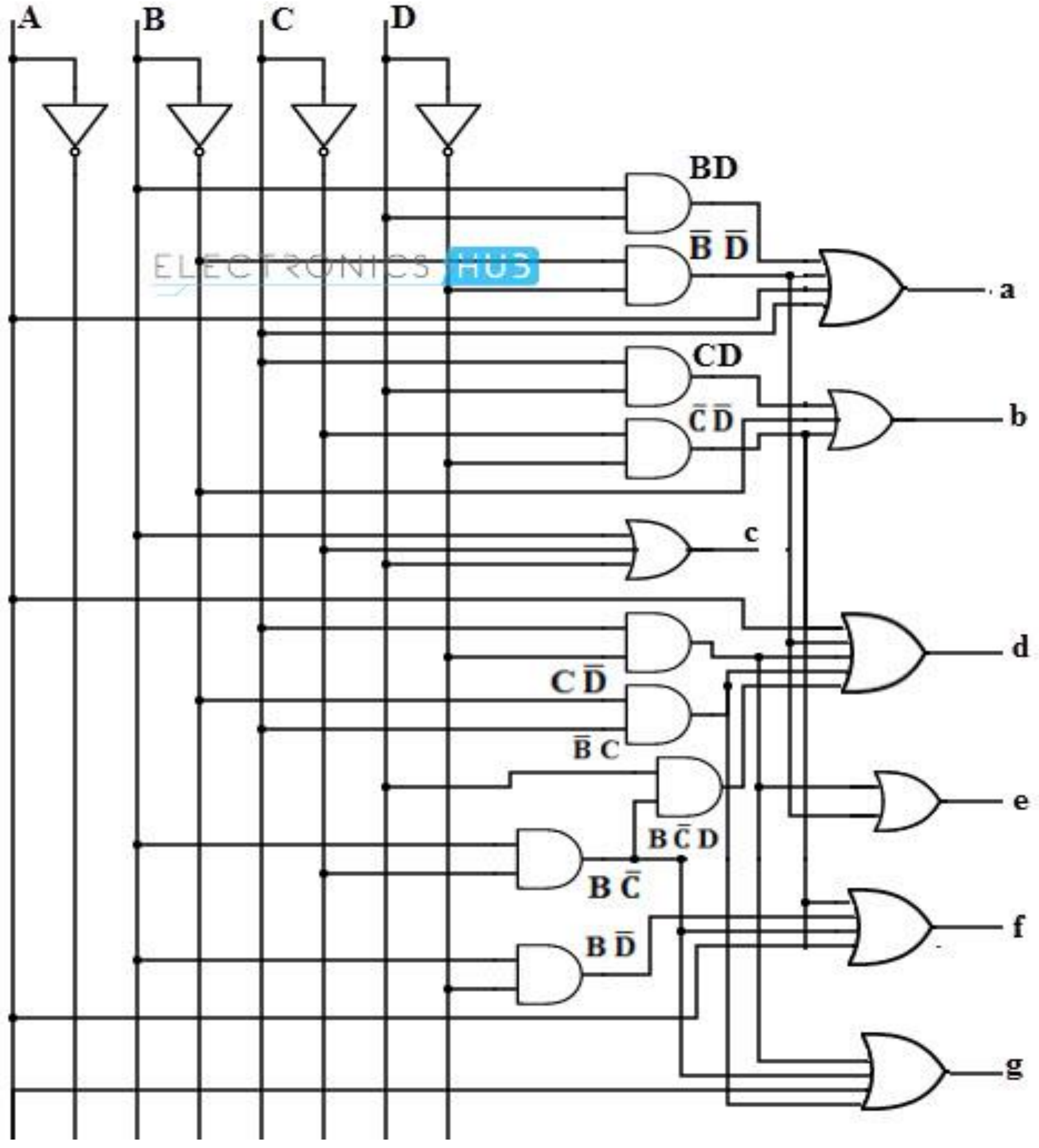
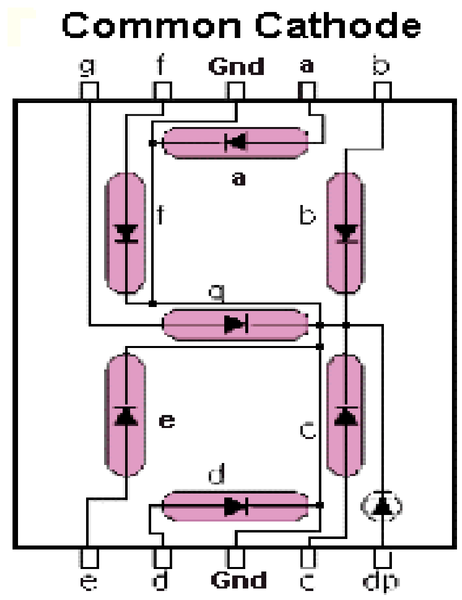
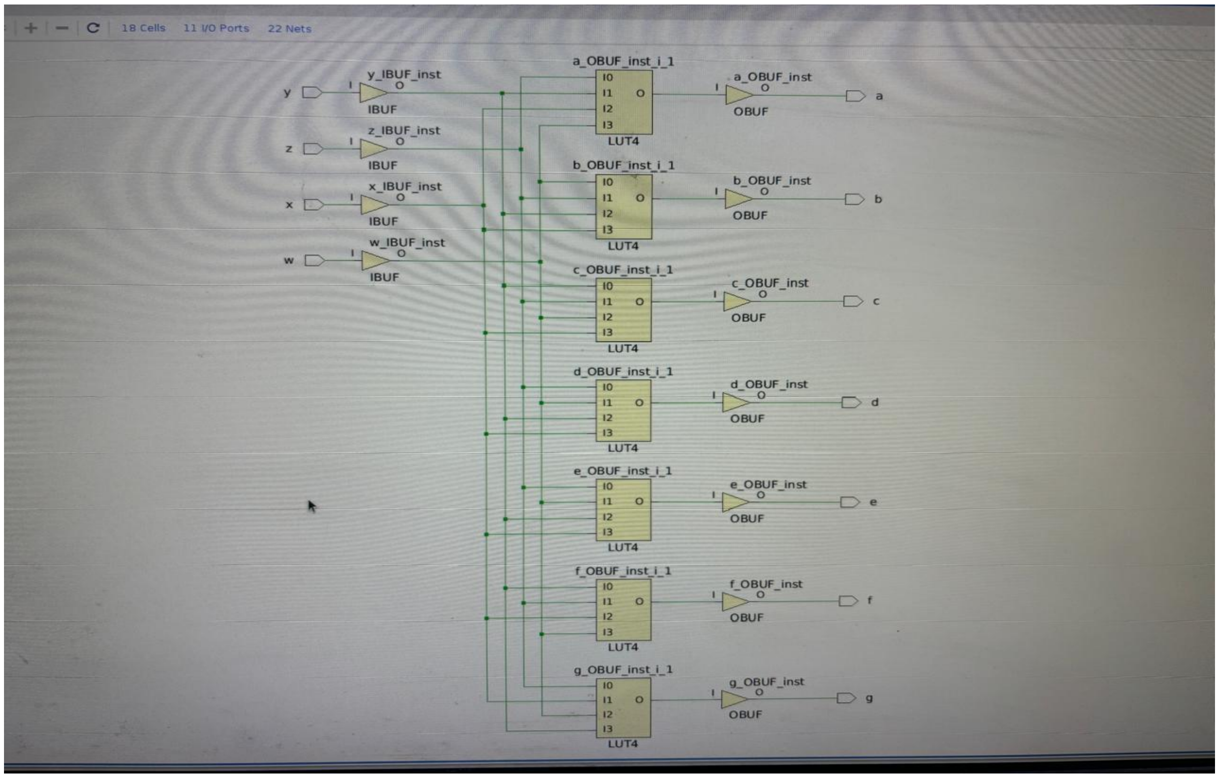


Figure of 7 SEGMENT DECODER:-

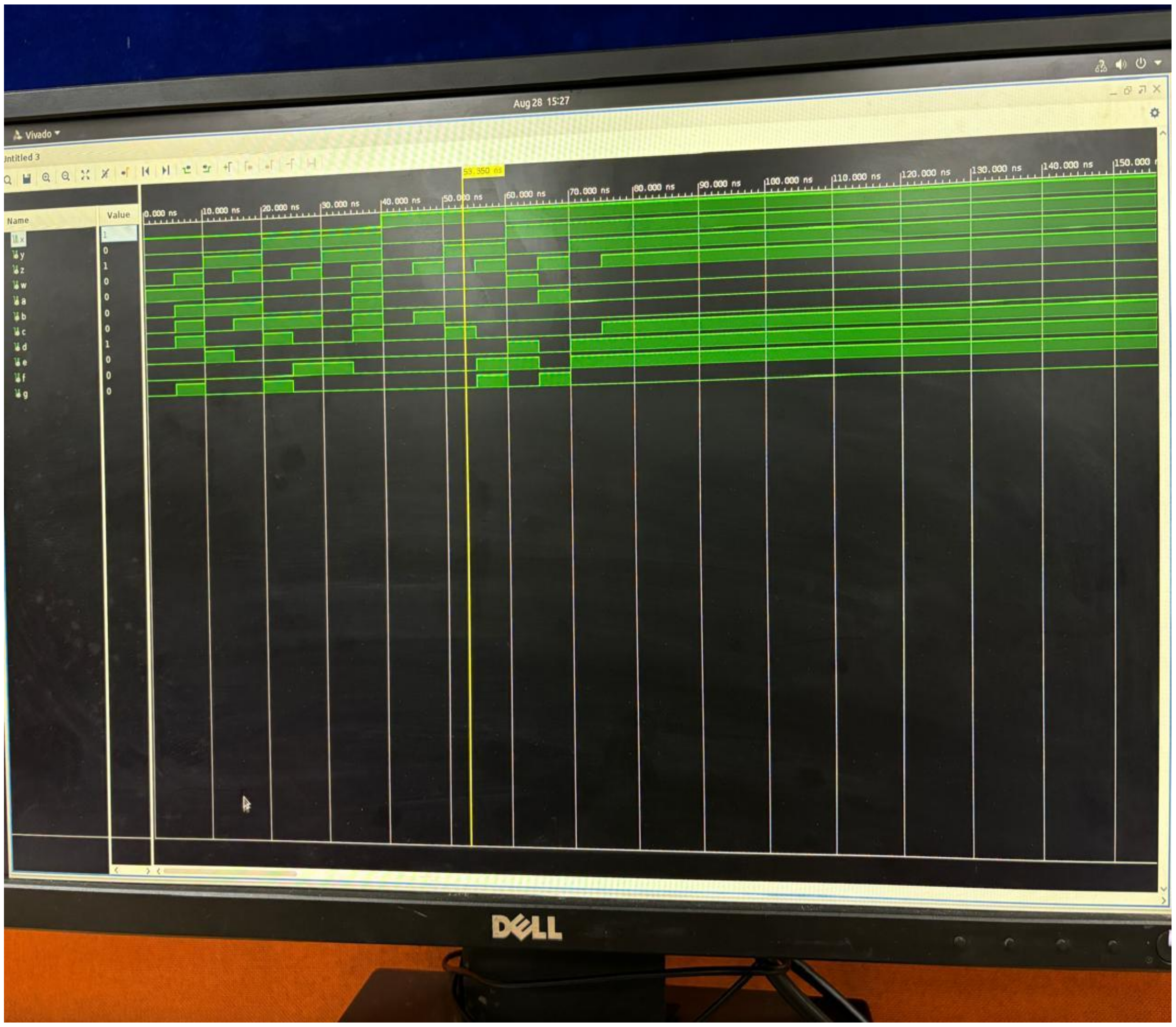


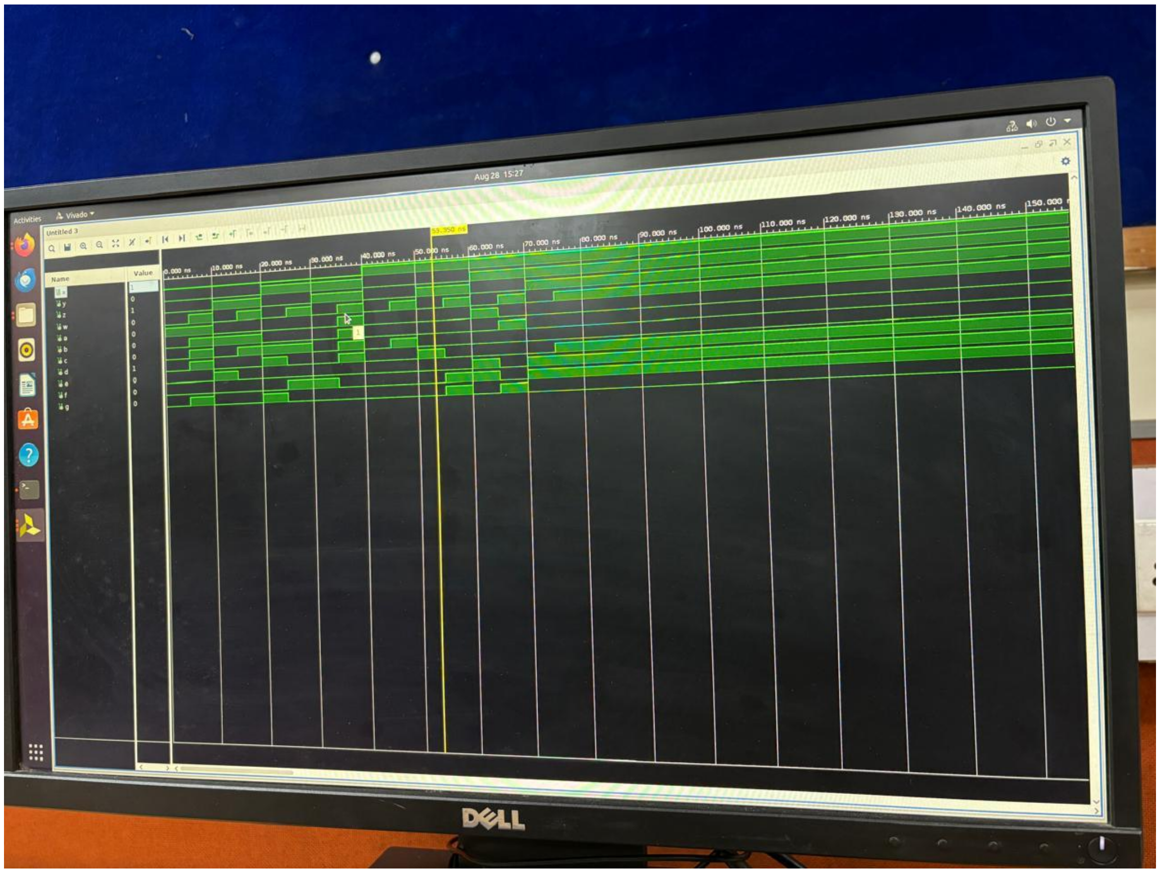
Driving All Four LED Displays

Schematic Diagram

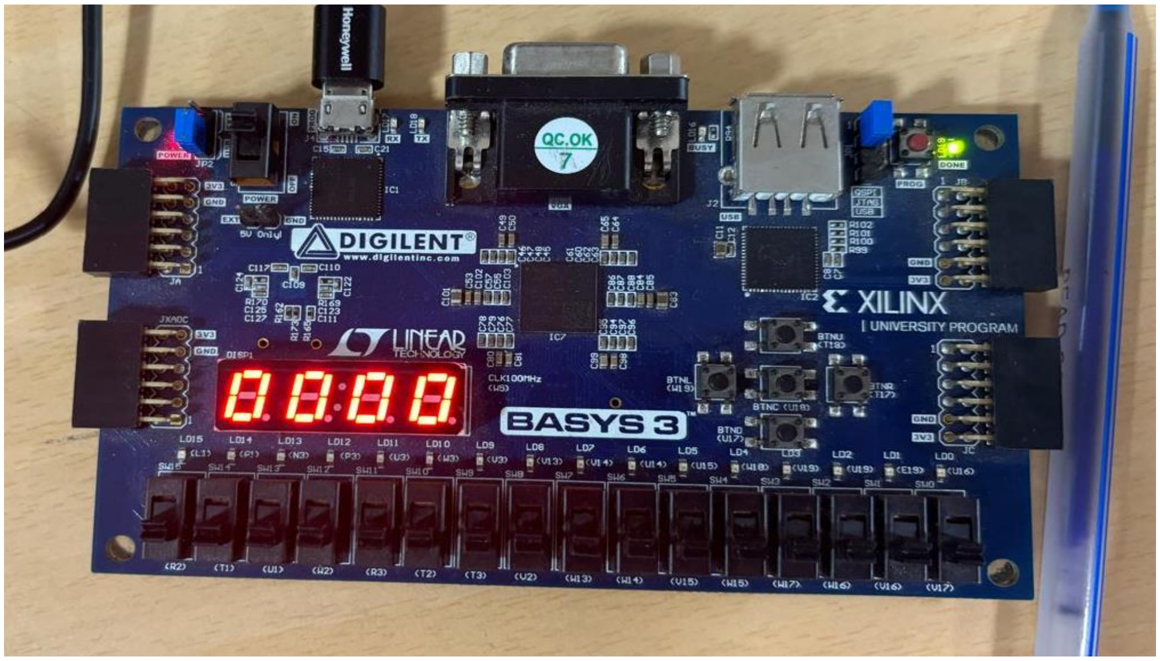


Simulation Snapshots

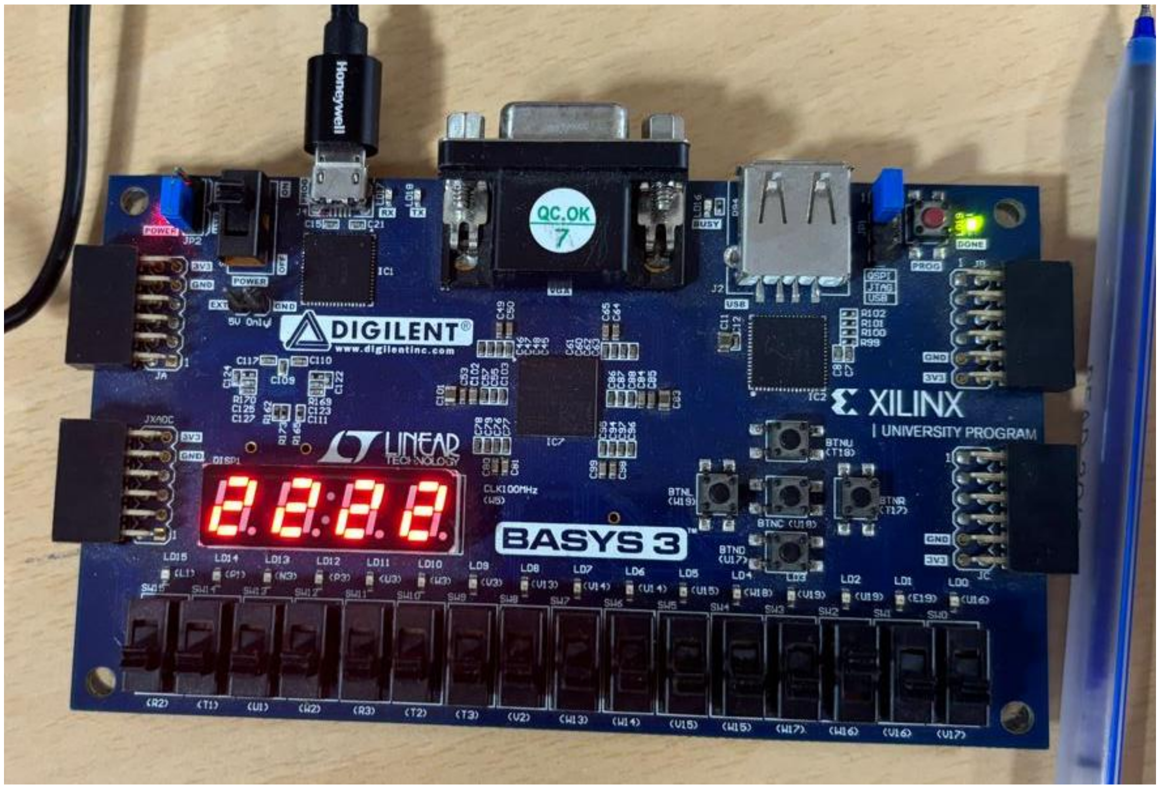




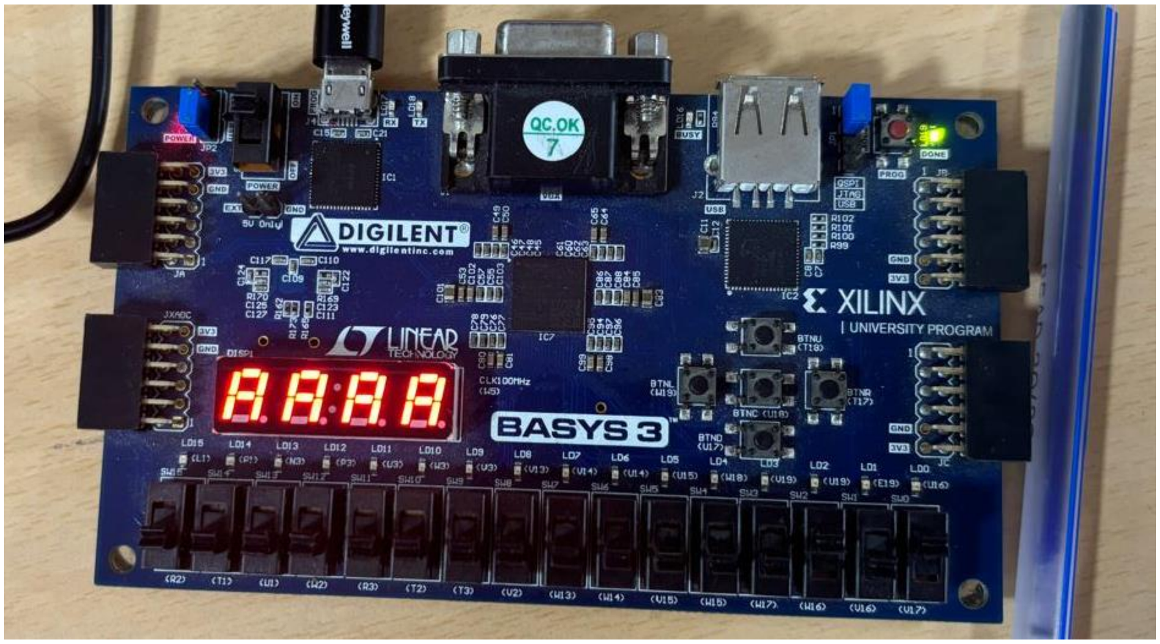
SNAPSHOTS OF BOARD:-



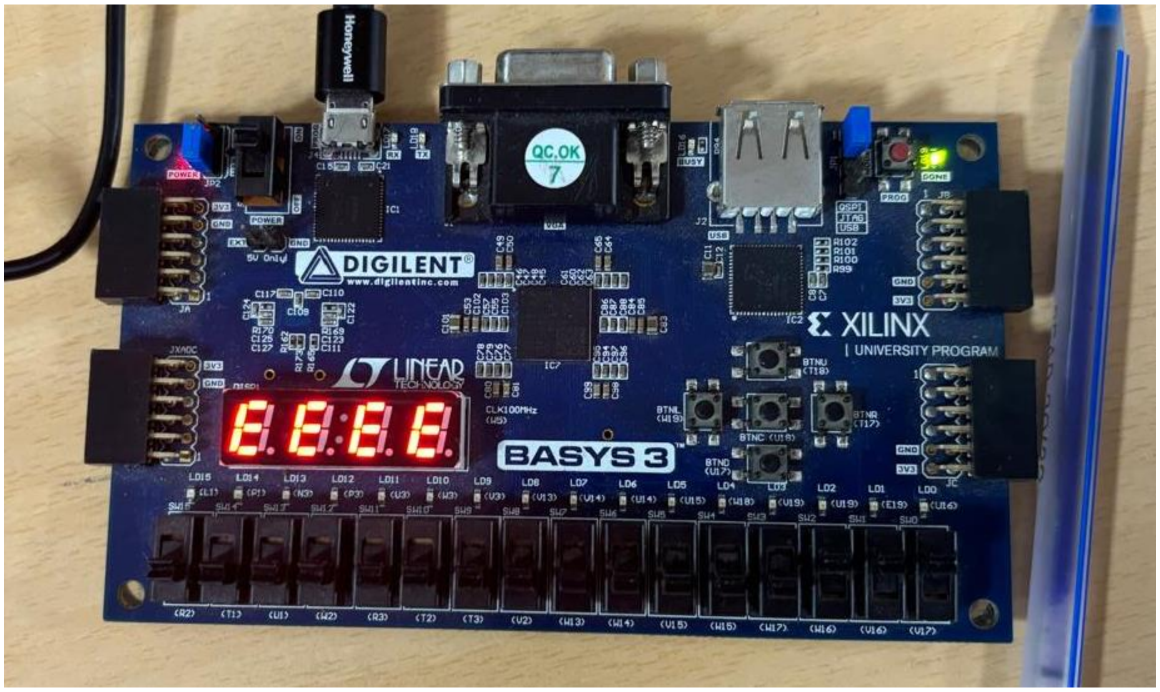
2.



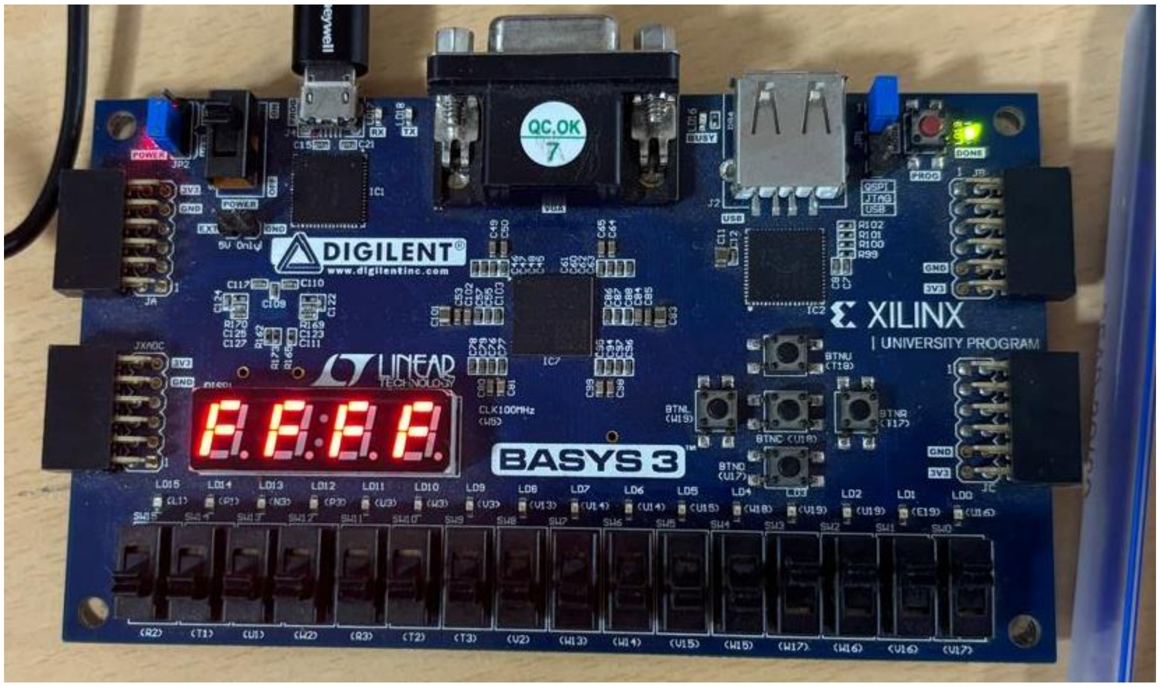
3.



4.


DIGILENT
XILINX
UNIVERSITY PROGRAM
BASYS 3
LINEAR TECHNOLOGY
QC OK 7

5.


DIGILENT
XILINX
UNIVERSITY PROGRAM
BASYS 3
LINEAR TECHNOLOGY
QC OK 7

1. Slice Logic Utilization

Site Type	Used	Fixed	Prohibited	Available	Utilization (%)
Slice LUTs	4	0	0	20,800	0.02
LUT as Logic	4	0	0	20,800	0.02
LUT as Memory	0	0	0	9,600	0.00
Register as Flip Flop	0	0	0	41,600	0.00

2. Slice Logic Distribution

Site Type	Used	Fixed	Prohibited	Available	Utilization (%)
LUT as Logic	4	0	0	20,800	0.02
LUT as Memory	0	0	0	9,600	0.00
LUT as Distributed RAM	0	0	0	N/A	N/A
LUT as Shift Register	0	0	0	N/A	N/A

3. Memory Utilization

Site Type	Used	Fixed	Prohibited	Available	Utilization (%)
Block RAM Tile	0	0	0	50	0.00
RAMB36/FIFO*	0	0	0	50	0.00
RAMB18	0	0	0	100	0.00

4. DSP Utilization

Site Type	Used	Fixed	Prohibited	Available	Utilization (%)
DSPs	0	0	0	90	0.00

PART – 2 (Driving all four LED displays)

Introduction

1. Overview of the Design

The VHDL code provided involves three main modules:

1. **Timing Block**: Generates a lower-frequency clock signal and controls the selection of the active seven-segment display.
2. **Multiplexer (MUX4)**: Selects one of four 4-bit inputs based on a control signal.
3. **Seven-Segment Decoder**: Converts a 4-bit binary input into the signals required to display a corresponding hexadecimal digit on a seven-segment display.

The top-level module, LED_Display1, instantiates these components and connects them to manage multiple seven-segment displays using time-multiplexing.

2. Top-Level Module (LED_Display1)

The LED_Display1 module serves as the top-level entity for the project. It integrates the three components (Timing Block, Multiplexer 4X1, and Seven-Segment Decoder) to drive four seven-segment displays.

Ports:

- **Inputs**:
  - clk_in: 100 MHz clock input.

- reset: Reset signal.
- input0, input1, input2, input3: Four 4-bit inputs representing binary values to be displayed on each of the four seven-segment displays.

- **Outputs**:
  - anodes: 4-bit vector controlling which display is active.
  - a, b, c, d, e, f, g: Cathode signals for segments a to g on the seven-segment displays.

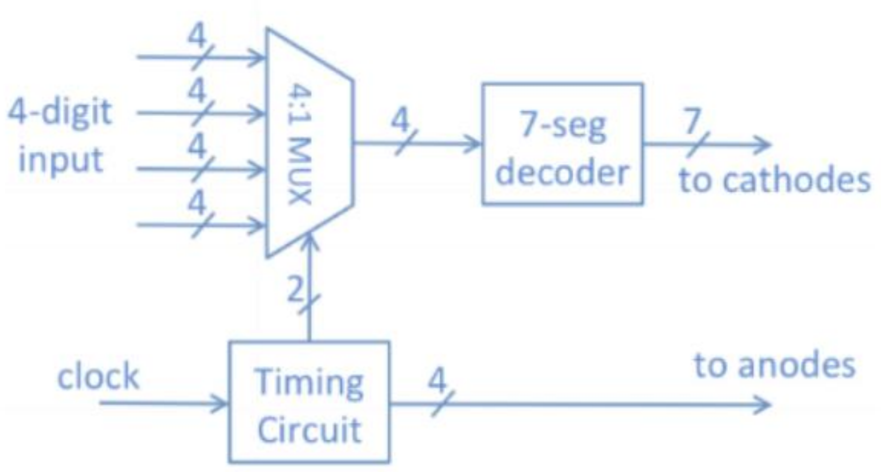


Figure 5: Block diagram for driving four 7 seven segment displays

Instantiated Components:

- **Timing Block**: Generates a slower clock signal to control the refresh rate between 1ms to 16ms and provides a selection signal for the multiplexer.

- **MUX4x1**: Selects one of the four 4-bit inputs based on the mux_select signal generated by the Timing_block.
- **Seven-Segments Decoder**: Decodes the selected 4-bit binary input into the appropriate segment signals for the seven-segment display.

3. Timing Block (Timing_block)

The Timing_block entity handles the clock division and anode signal control to achieve a time-multiplexed display.

Functionality:

1. **Clock Division**:
    - A process named NEW_CLK divides the input clock (clk_in) from 100 MHz to a much slower frequency (adjustable by the constant N) to achieve the desired refresh rate.
    - A signal new_clk1 is toggled at the desired frequency to control the refresh rate.
2. **Multiplexer Select Signal Generation**:
    - A second process, MUX_select1, uses the slower clock (new_clk1) to generate a 2-bit signal (mux_select) that cycles through 00, 01, 10, and 11.
    - This cycling allows the multiplexer to select each of the four 4-bit inputs in turn.
3. **Anode Control**:

- A third process, ANODE_select, controls which display is active by adjusting the anodes output based on the mux_select value.
- The anodes signal is active-low, meaning 1110 activates the first display, 1101 the second, and so on.

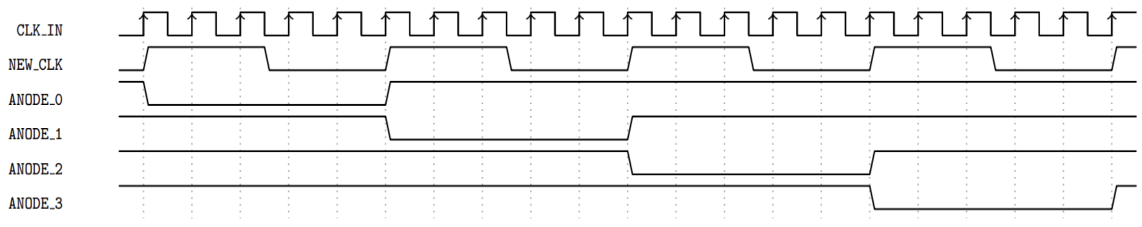


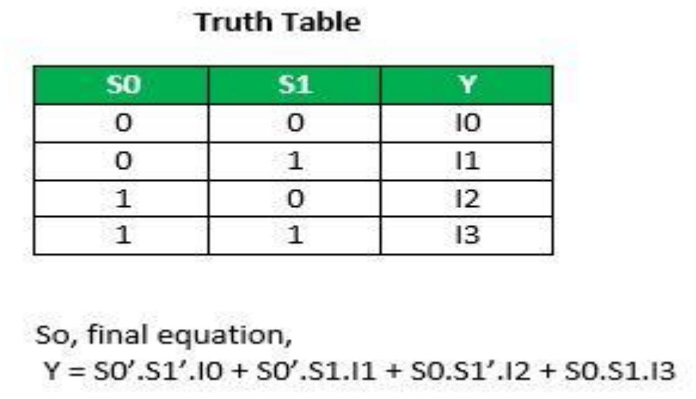
Figure 6: Illustration of refresh clock signal

4. Multiplexer (MUX4)

The MUX4 entity is a simple 4-to-1 multiplexer that selects one of the four 4-bit inputs to pass to the output based on a 2-bit select1 input.

Functionality:

- The select1 input determines which input (input0 to input3) is routed to the output.
- The selected 4-bit input is sent to the Seven_Segments_Decoder for decoding and display.


Truth Table

S0	S1	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

So, final equation,

$Y = S0'.S1'.I0 + S0'.S1.I1 + S0.S1'.I2 + S0.S1.I3$

5. Seven-Segment Decoder (Seven_Segments_Decoder)

The Seven_Segments_Decoder entity decodes a 4-bit binary input into seven individual segment signals (a to g) to drive a seven-segment display.

Functionality:

- The four inputs x, y, z, and w correspond to a 4-bit binary number.
- The outputs a to g are derived from combinational logic that generates the appropriate segment pattern for displaying hexadecimal digits (0 to F) on a seven-segment display.
- Each equation for the outputs (a to g) represents the minimized logic required to turn on/off the corresponding segments of the seven-segment display based on the binary input.

6. Summary and Analysis for Project Report

6.1. Design Strategy:

The modular design strategy enhances clarity and scalability. Each module is designed to perform a specific function (timing, multiplexing, decoding), which simplifies debugging and testing.

6.2. Clock Management:

The Timing_block efficiently manages the clock by dividing a high-frequency clock signal to control the refresh rate of the display. This method allows for a flicker-free display experience.

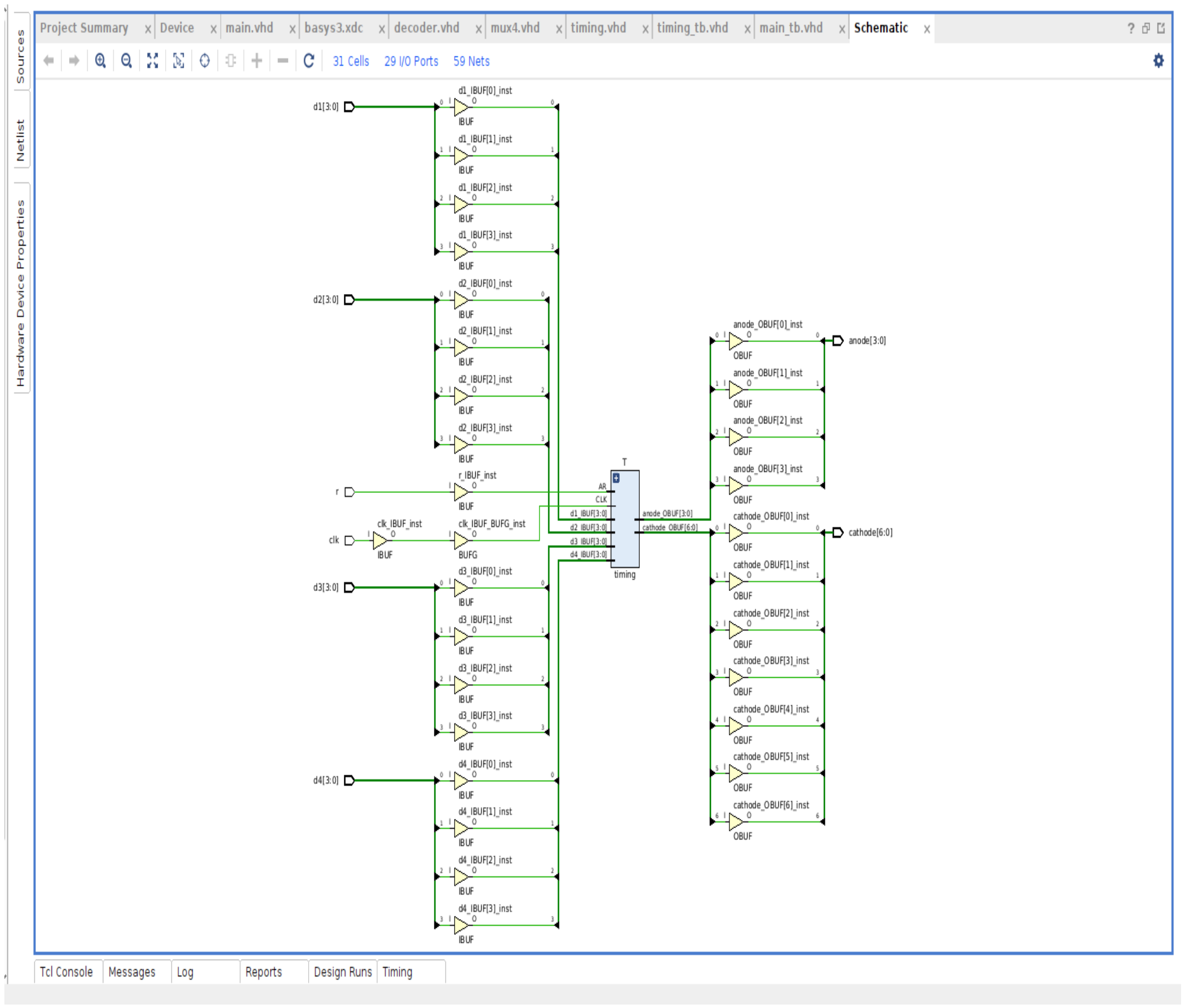
6.3. Multiplexing:

The MUX4 component demonstrates a straightforward way to handle multiple inputs and display them sequentially using time-multiplexing. This approach saves resources by using a single seven-segment display driver for multiple displays.

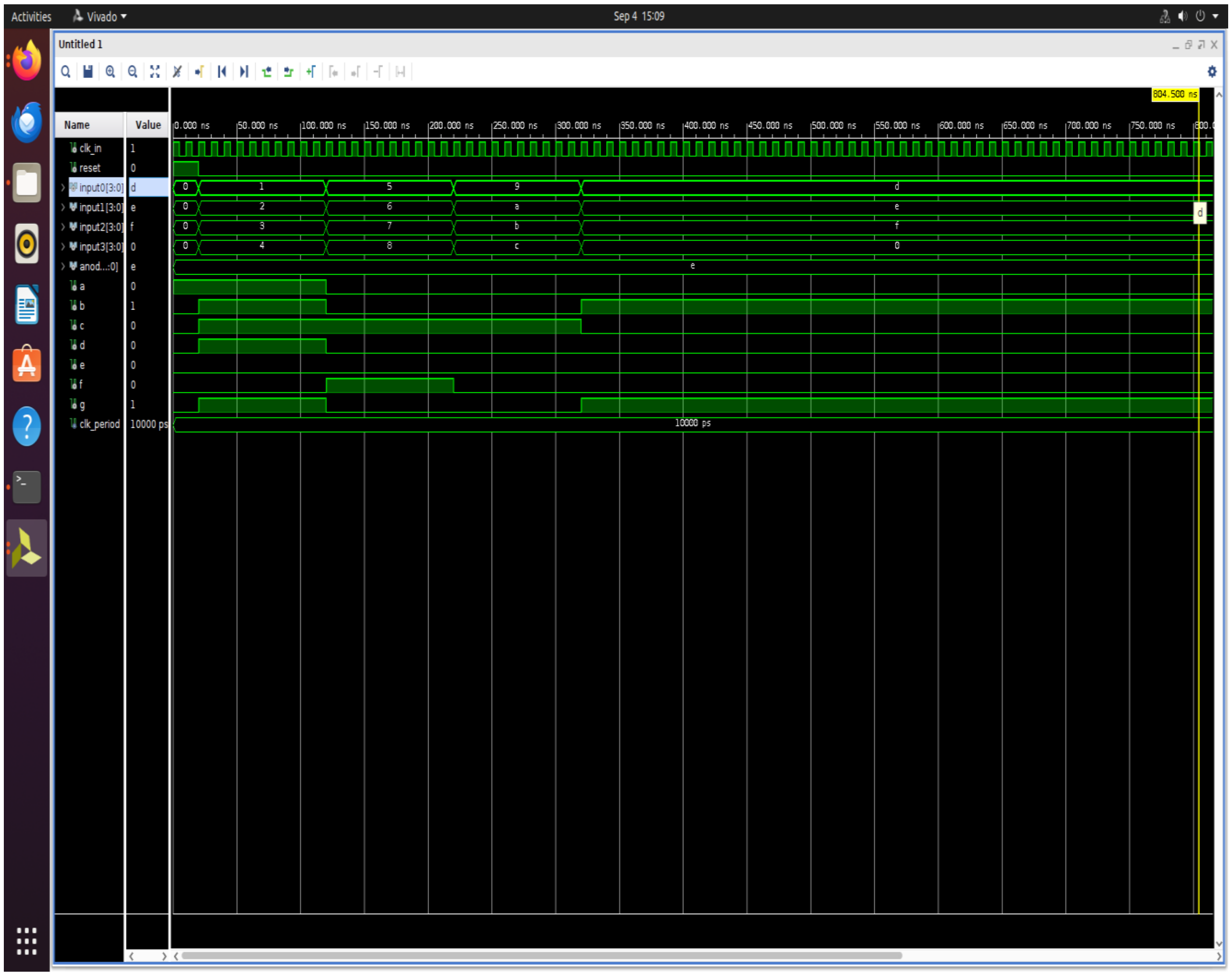
6.4. Seven-Segment Decoding:

The Seven_Segments_Decoder is designed using minimized Boolean expressions to ensure that the number of logic gates used is reduced. This efficiency is crucial for optimizing FPGA resource usage

Schematic Diagram for Part2



Testbench Simulation



Utilization Report

1. Slice Logic Utilization

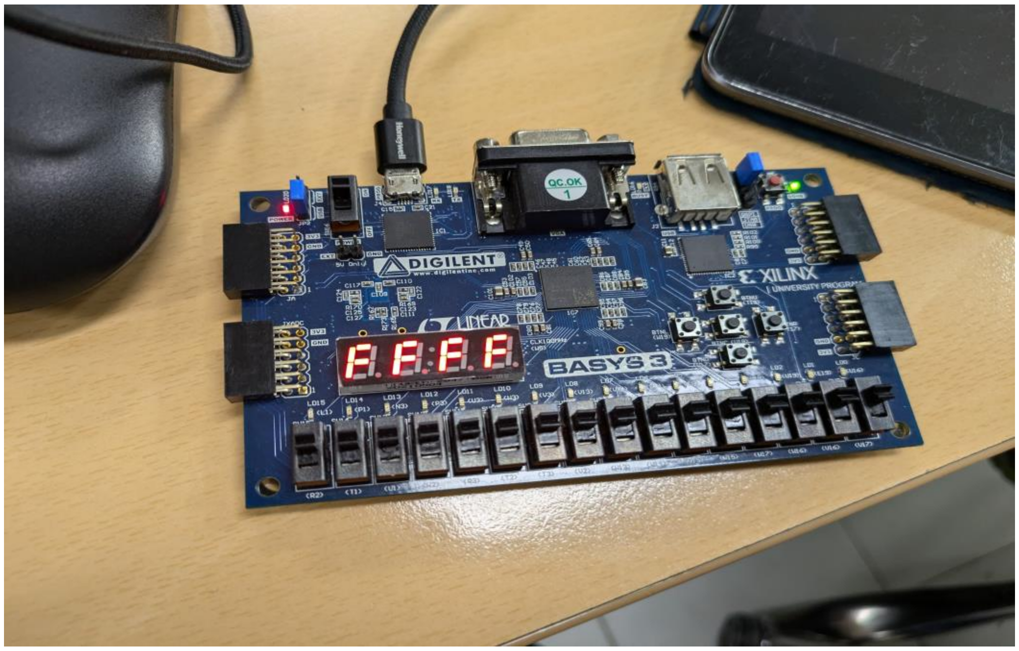
Site Type	Used	Fixed	Prohibited	Available	Utilization (%)
Slice LUTs*	51	0	0	20,800	0.25
LUT as Logic	51	0	0	20,800	0.25
LUT as Memory	0	0	0	9,600	0.00
Slice Registers	35	0	0	41,600	0.08
Register as Flip Flop	35	0	0	41,600	0.08

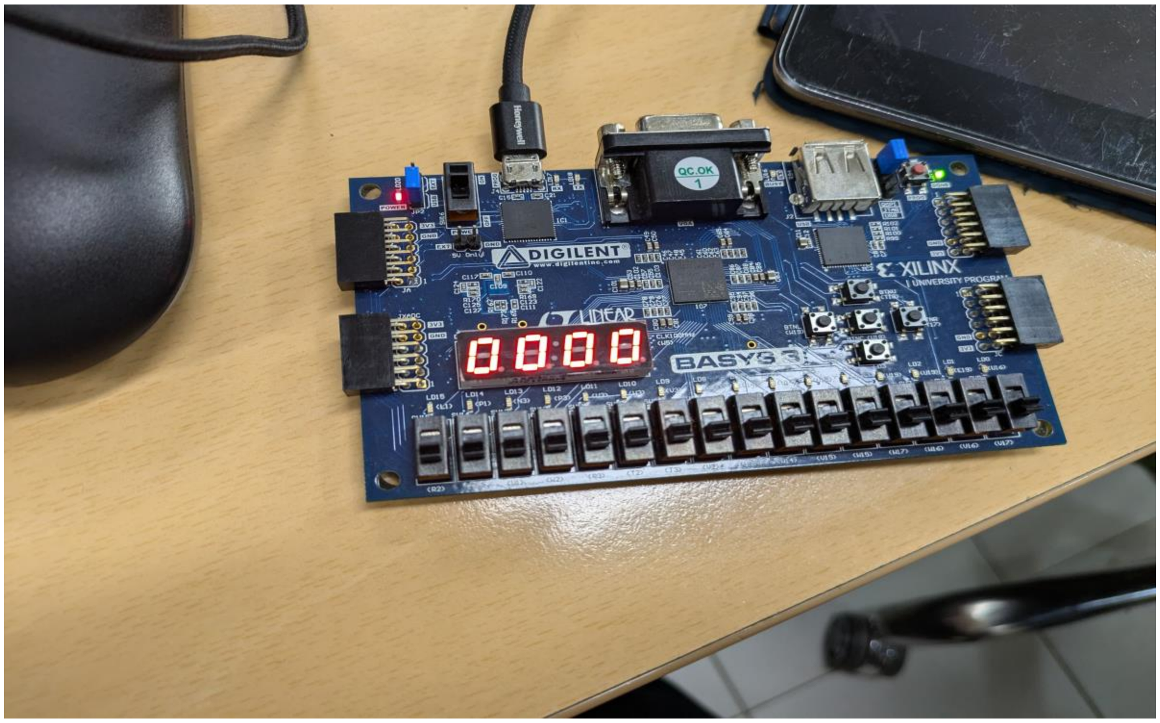
2. Memory Utilization

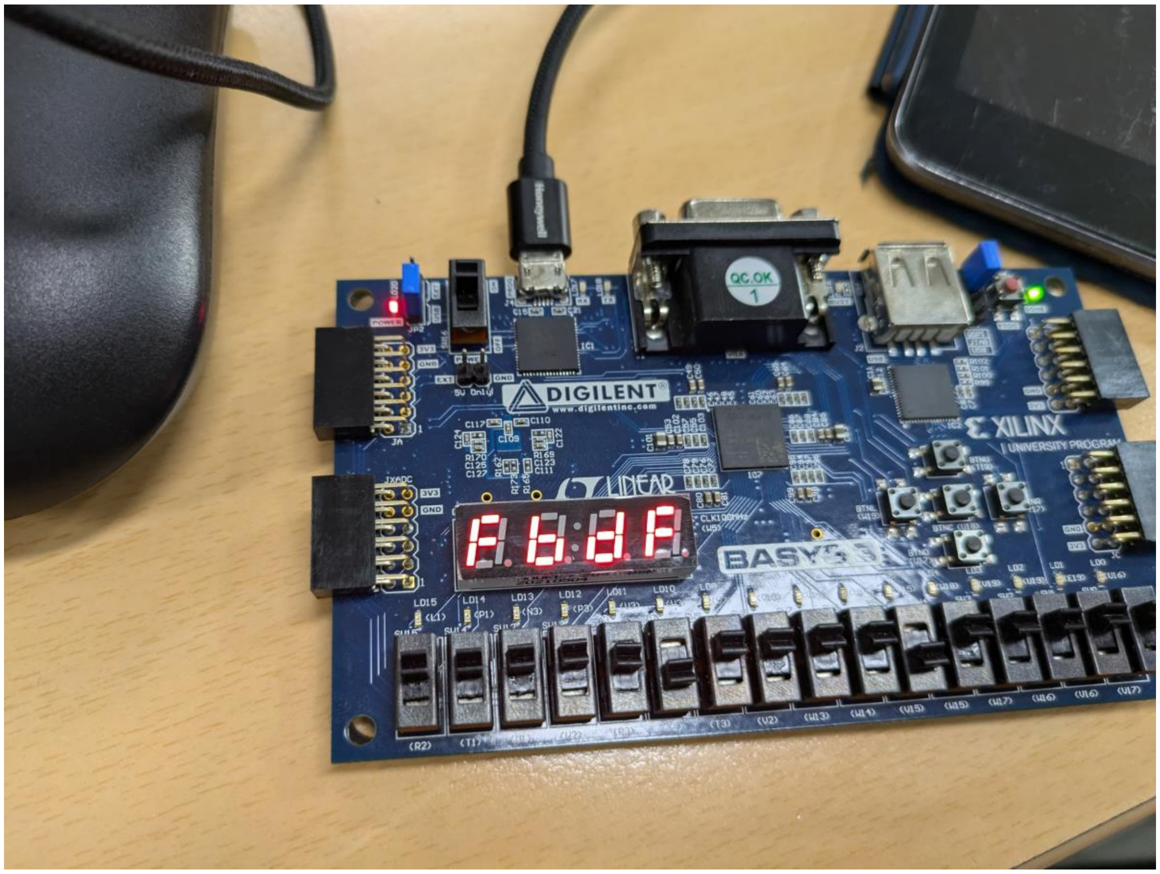
Site Type	Used	Fixed	Prohibited	Available	Utilization (%)
Block RAM Tile	0	0	0	50	0.00
RAMB36/FIFO*	0	0	0	50	0.00
RAMB18	0	0	0	100	0.00

3. DSP Utilization

Site Type	Used	Fixed	Prohibited	Available	Utilization (%)
DSPs	0	0	0	90	0.00




DIGILENT
XILINX
BASYS3
0000


DIGILENT
XILINX
BASYS3
FbdF