IIT Delhi

Indian Institute of Technology Delhi

Hardware Assignment 2

Hexadecimal Digit Display

By –

Jatin Kaushik (2023CS11169)

Vijay Balaji N (2023CS51103)

4-Digit Hexadecimal Display

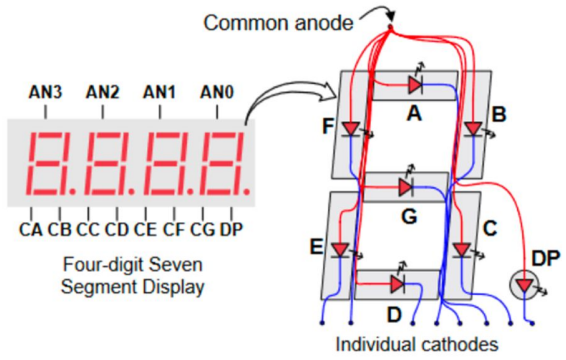
AIM:

Design and implement a circuit that takes a 4-digit decimal/hexadecimal number as an input (so each number is 4-bit) from switches in the Basys3 board and displays it on the 4-seven segment displays on the board.

Theory:

7-Segment Display:

A 7-segment display is an electronic device used to display decimal numbers and certain alphabetic characters. It consists of seven LEDs (labeled 'a' to 'g') arranged to form digits, with an optional decimal point. It has 7 cathode pins, 1 anode pin (connected to all 7 LEDs) and 1 pin for decimal point. In multi-digit displays, multiplexing is used, where each digit is activated one at a time in a rapid cycle to create the illusion of continuous display. Proper timing and refresh rates between 1 kHz and 60 Hz are essential to prevent flickering. They are commonly used in digital clocks, calculators, and other numeric displays.

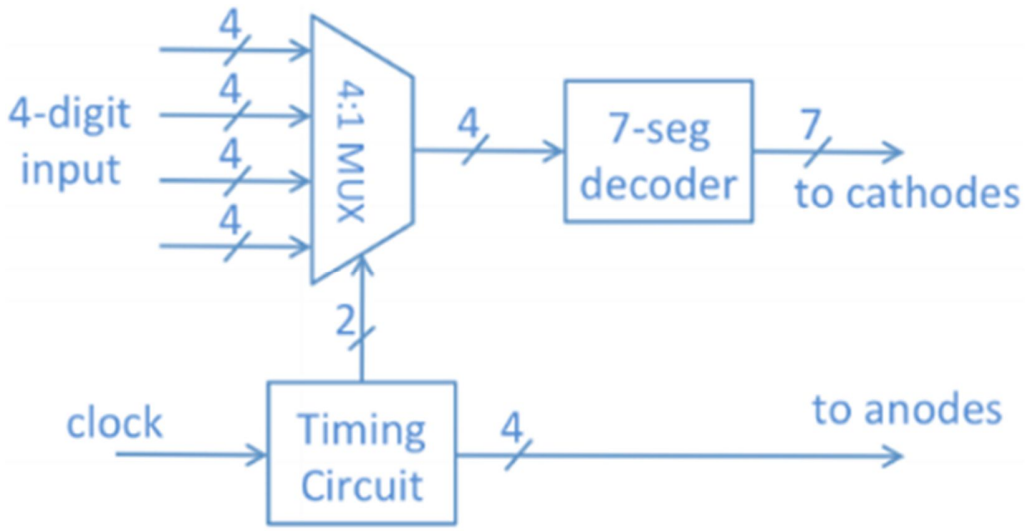


Design Decisions:

There were three main components to be made

1. The seven digit decoder
2. The Mux4X1
3. The timing circuit

All of these components had to be connected in the appropriate manner to get the final output. This was done using the assembly file.



1. The Timing circuit:

   The timing circuit takes the input from the clock which is inbuilt and has a frequency of 100MHz and decreases it to create a new clock which has an optimal frequency so that it ensures that the display is stable and does not flicker.

   This is done by periodically updating a signal counter whenever a rising edge is noticed in the input clock and whenever the value of the counter has reached N-1 where N is the factor by which we want to reduce the frequency, we change the bit of the new clock which we are making.

   It then periodically gives an input selector to the Mux4X1 to select which digit must displayed which time and also gives the anode's input specifying which anode to turn on corresponding to the digit.

These are done using the when case constructs in processes having the new clock and the mux select as the inputs respectively.

2. The Mux4X1:

   The Mux4X1 takes in inputs from the timing circuit which acts as the selector as well from the 16 switches in the Basys3 board which is given in the form of 4-bit vectors (Hexadecimal).

   Then the Mux4X1 selects one of the four 4-bit vectors according to the selector supplied from the timing circuit.

   This selected 4-bit vector is given as the input to the seven-digit decoder.

   The above process is done using the if else constructs in processes which has the selector in its sensitivity list.

3. The seven digit decoder:

   The seven-digit decoder takes input from the Mux4X1 and outputs the signals to the seven cathodes as to which one to be kept on and which to be turned off.

   The above process was done by assigning signals to each of the 7 cathodes which are dependent on 4 input signals using K-Maps and Truth tables.

The assembly makes sure that the correct input is given to each of the above components and the required output is given to the anodes and cathodes using the constraint files.

Truth Table:

The truth table for each cathode is as follows:

Hexadecimal	A	B	C	D	E	F	G
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
b	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
d	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

Logic minimizations using K-maps:

We represent each hexadecimal number as four-bit binary numbers pqrs.

K-maps for cathode A

pq/rs	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	0	0	0	1
10	0	1	0	0

A=p’q’r’s+p’qr’s’+pqr’s+pq’rs

K-maps for cathode B

pq/rs	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	0	1
10	1	0	1	1

B=p’qr’s+qrs’+pqs’+prs

K-maps for cathode C

pq/rs	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	0	0	0	0
10	1	0	1	1

C=p’q’rs’+pqr+pqs’

K-maps for cathode D

pq/rs	00	01	11	10
00	0	1	0	0
01	1	0	0	1
11	0	0	1	0
10	0	0	0	1

D=qrs+p’q’r’s+p’qr’s’+pq’rs’

K-maps for cathode E

pq/rs	00	01	11	10
00	0	1	0	1
01	1	1	0	1
11	0	1	0	0
10	0	0	0	0

E=p’s+p’qr’+q’r’s

K-maps for cathode F

pq/rs	00	01	11	10
00	0	1	1	1
01	0	0	0	1
11	0	0	0	0
10	1	0	1	1

F=p’q’s+p’q’r+p’rs+pqr’s

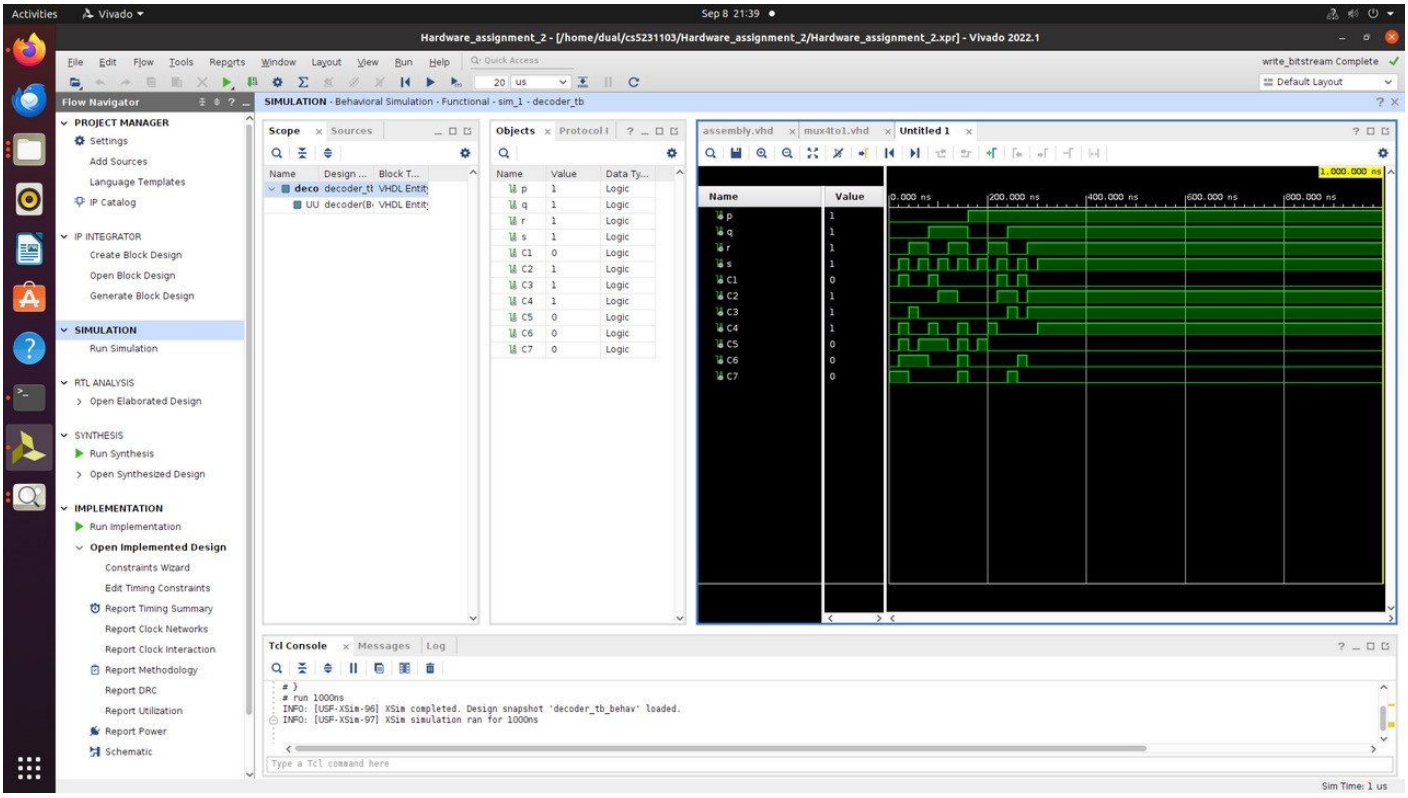
K-maps for cathode G

pq/rs	00	01	11	10
00	1	1	0	0
01	0	0	0	1
11	0	0	0	0
10	0	0	0	0

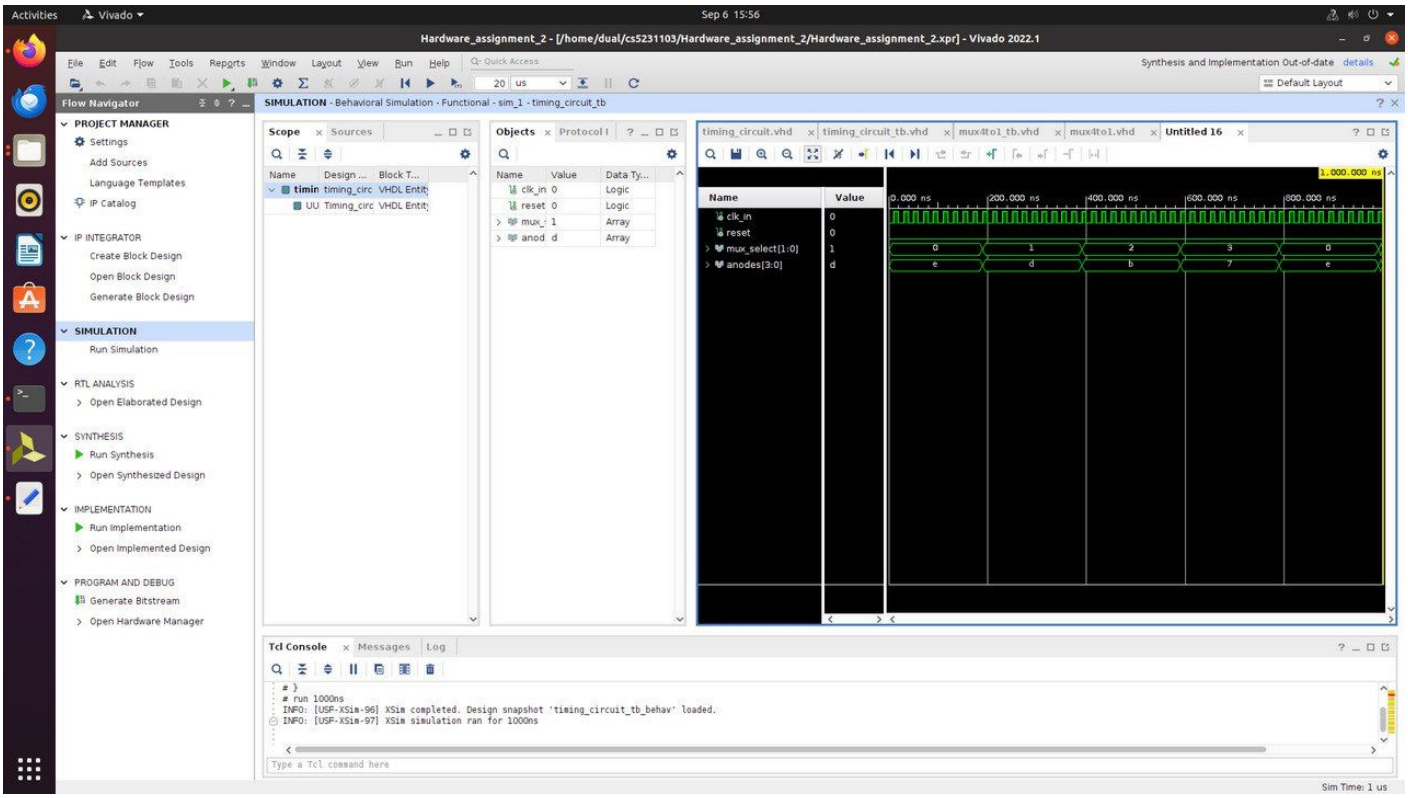
G=p’q’r’+p’qrs+pqr’s’

Simulation Snapshots:

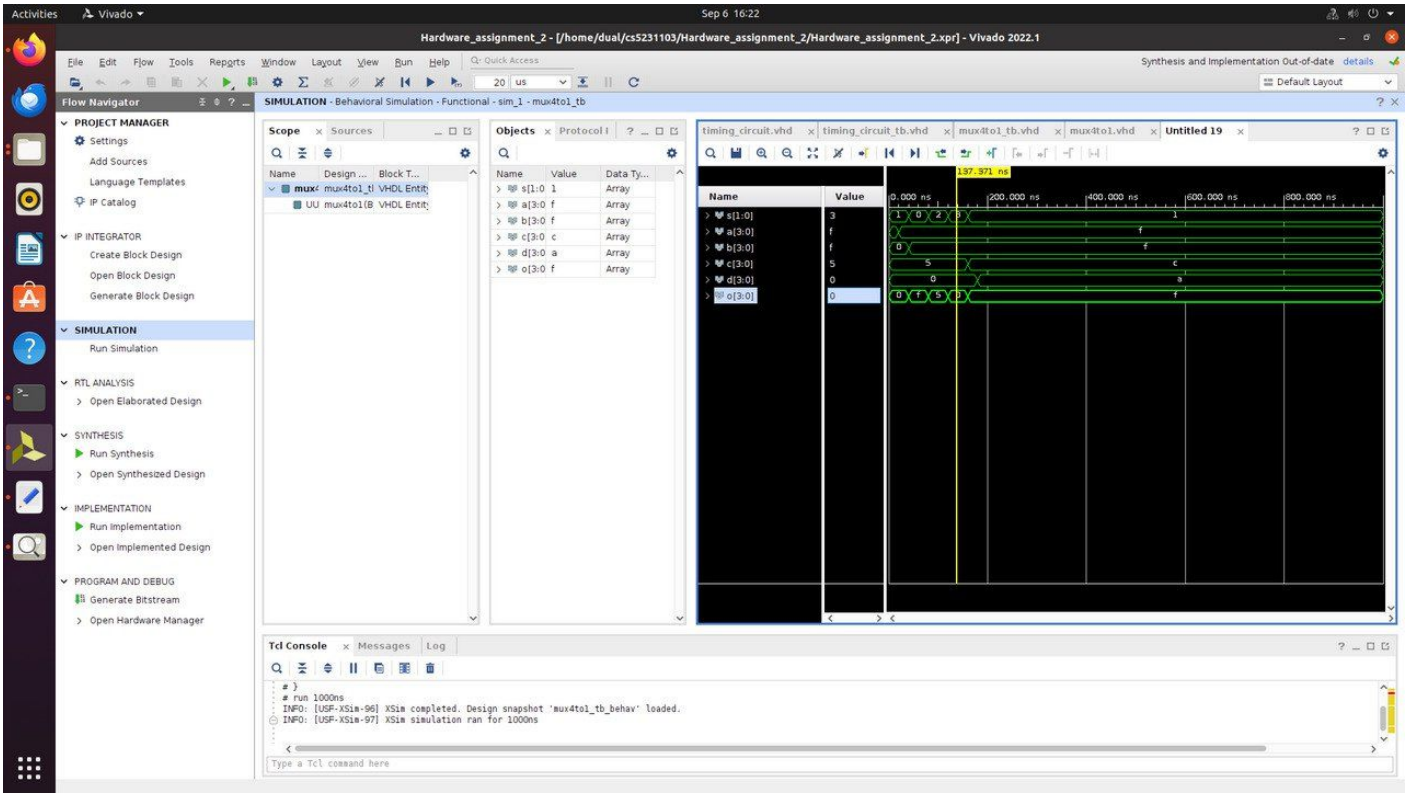
Simulation of seven-digit decoder:



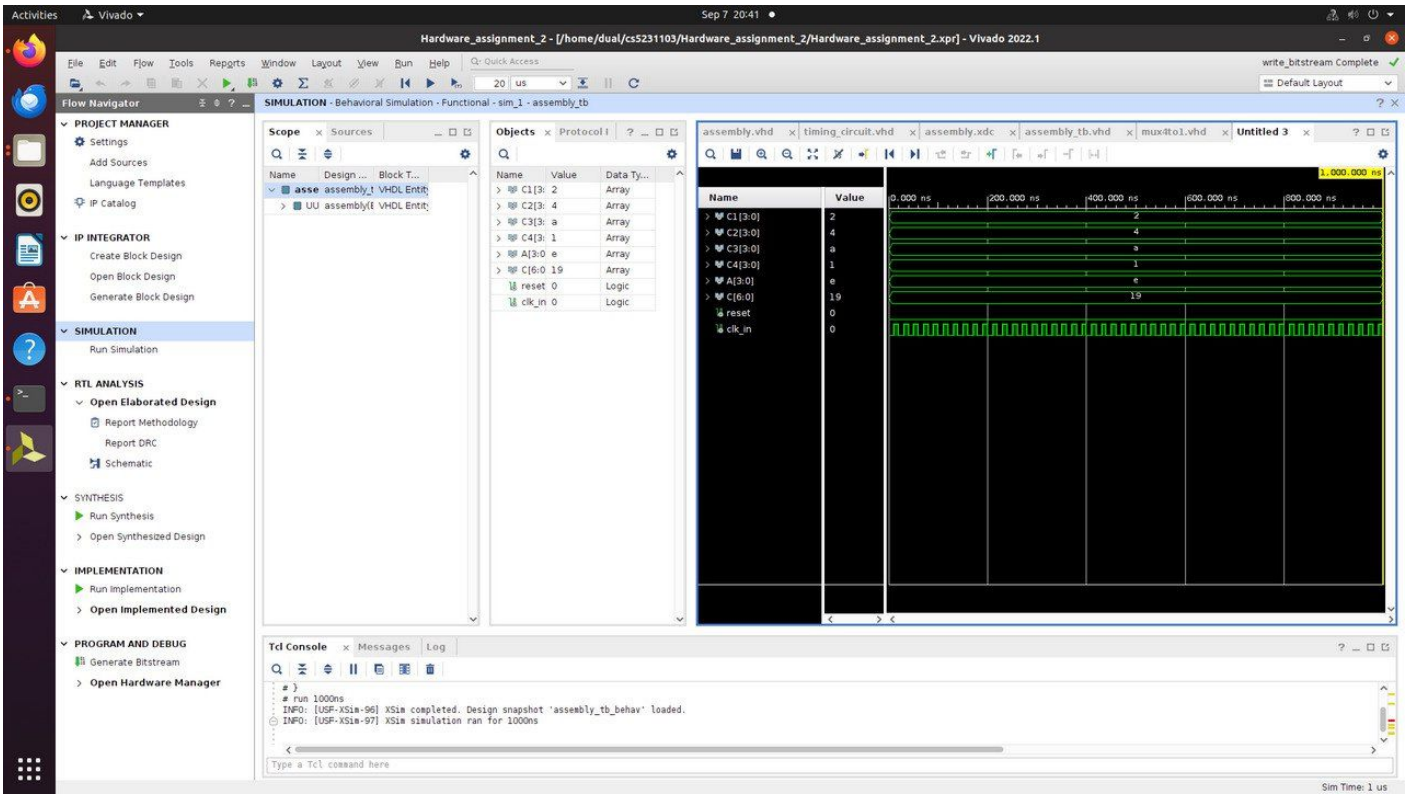
Simulation of the timing circuit:



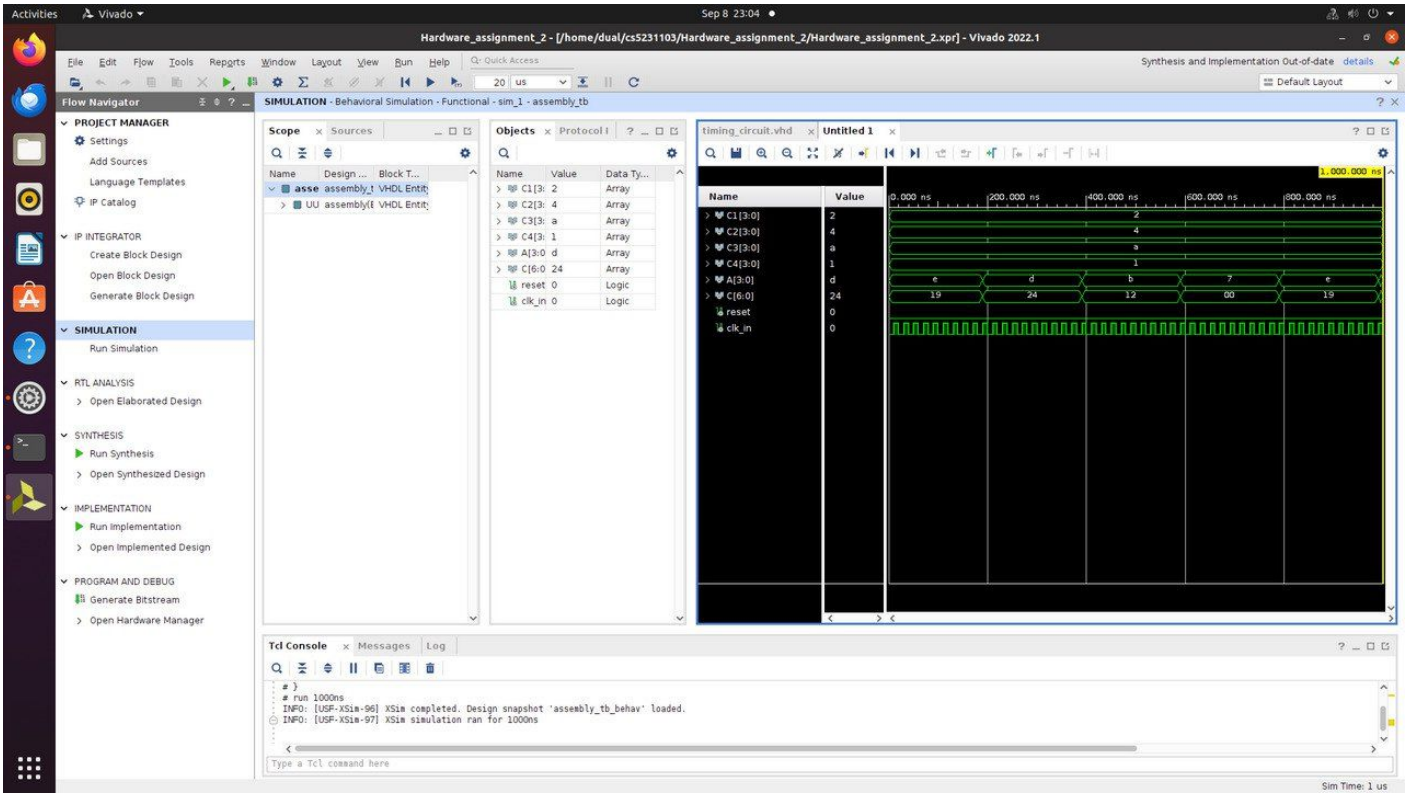
Simulation of Mux4X1:



Simulation of Assembly:



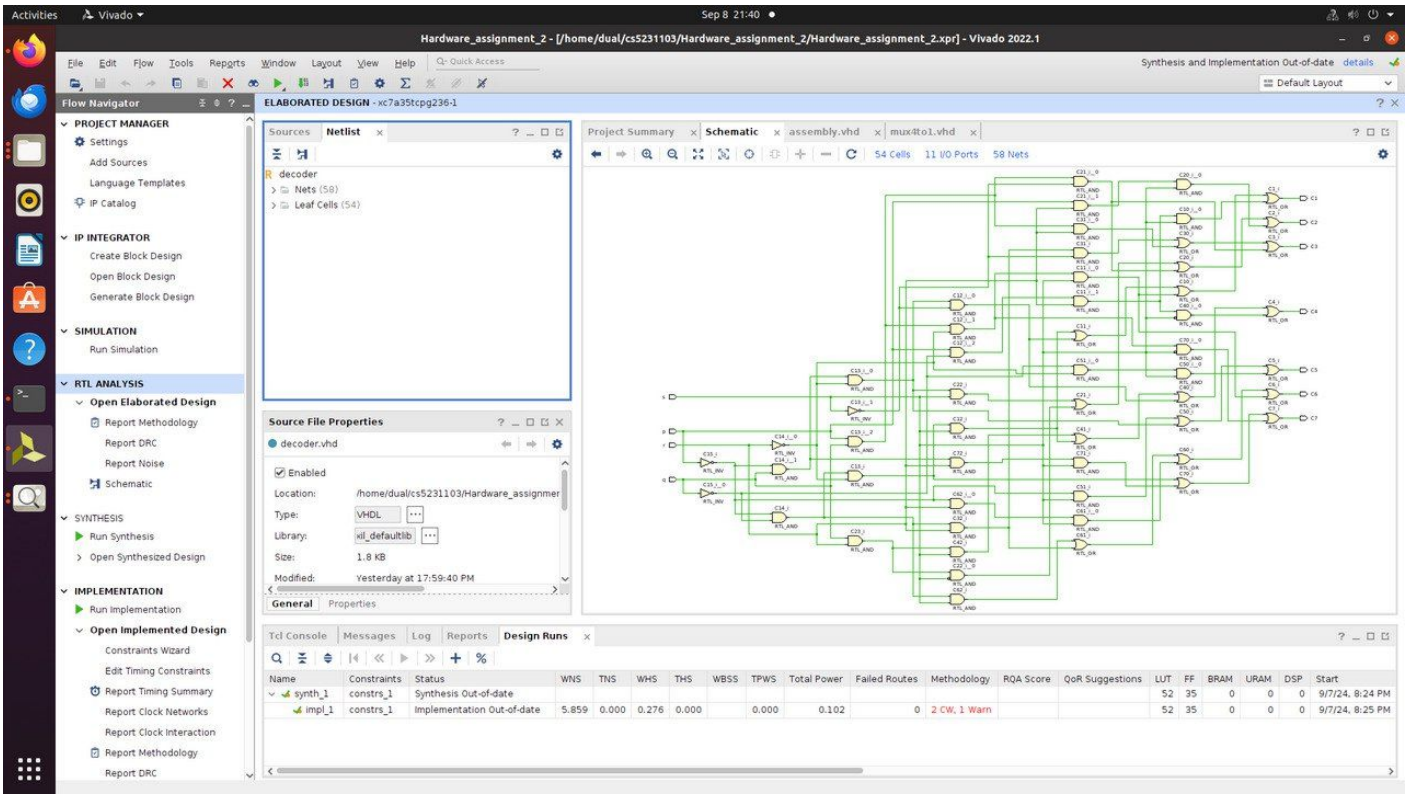
Case 1: when the value of N is high



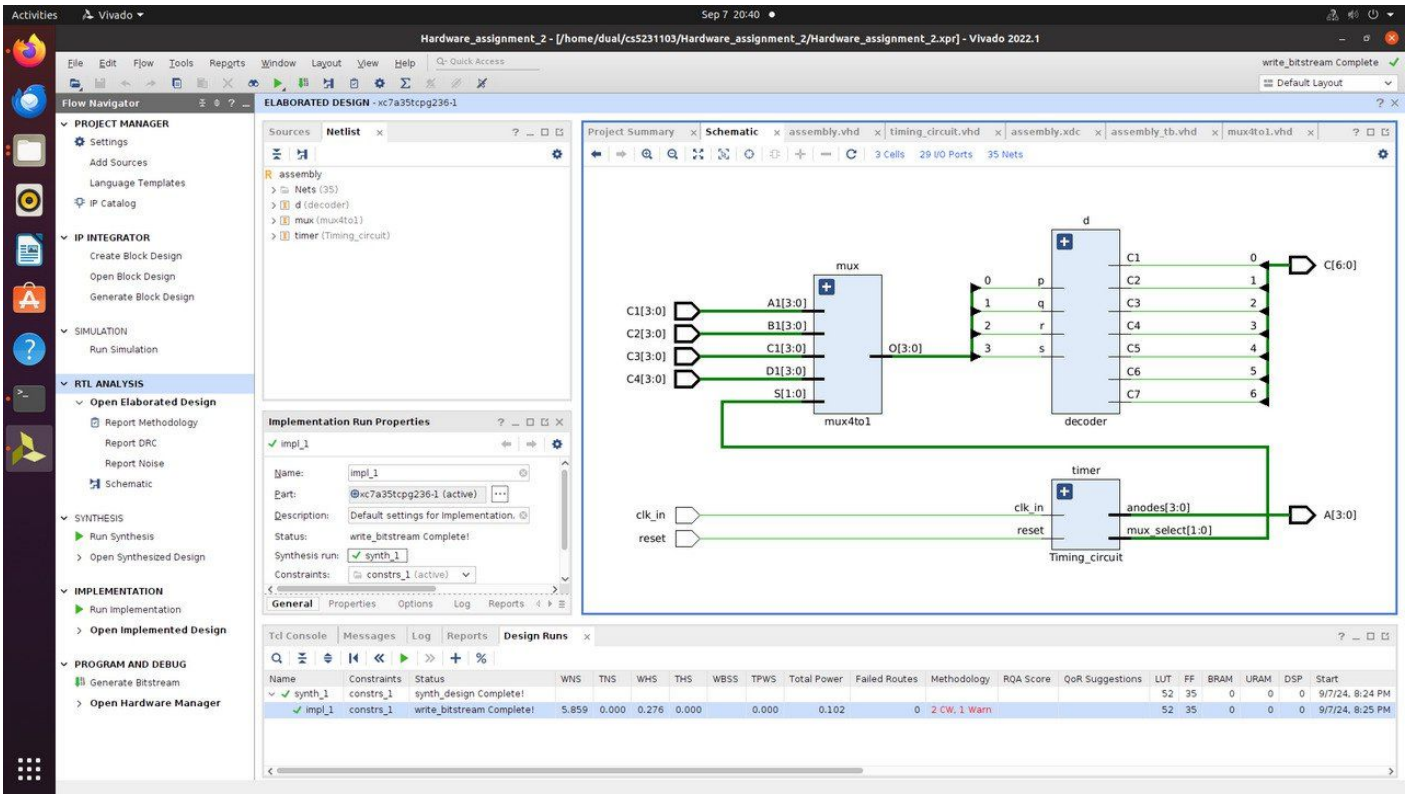
Case 2: when the value of N is low (5) for simulation purposes

Schematic Snapshots:

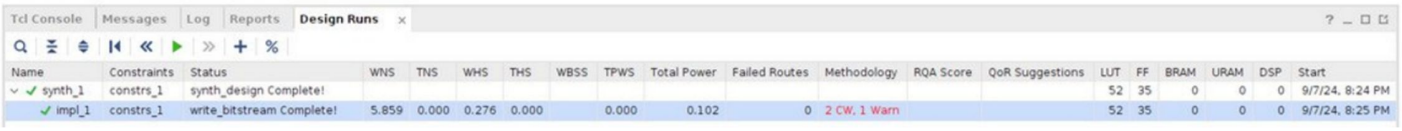
Schematic Snapshot of seven-bit decoder:



Schematic Snapshot of Assembly:



Resource Utilization:



Tcl Console | Messages | Log | Reports | **Design Runs**

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start
synth_1	constrs_1	synth_design Complete!												52	35	0	0	0	9/7/24, 8:24 PM
impl_1	constrs_1	write_bitstream Complete!	5.859	0.000	0.276	0.000		0.000	0.102	0	2 CW, 1 Warn			52	35	0	0	0	9/7/24, 8:25 PM

```
Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.
----------------------------------------------------------------------------------------------------------------------------
| Tool Version : Vivado v.2022.1 (lin64) Build 3526262 Mon Apr 18 15:47:01 MDT 2022
| Date         : Sat Sep  7 20:25:51 2024
| Host         : dhd running 64-bit Ubuntu 20.04.3 LTS
| Command      : report_utilization -file assembly_utilization_placed.rpt -pb assembly_utilization_placed.pb
| Design       : assembly
| Device       : xc7a35tcpg236-1
| Speed File   : -1
| Design State : Fully Placed
----------------------------------------------------------------------------------------------------------------------------

Utilization Design Information

Table of Contents
-----------------
1. Slice Logic
1.1 Summary of Registers by Type
2. Slice Logic Distribution
3. Memory
4. DSP
5. IO and GT Specific
6. Clocking
7. Specific Feature
8. Primitives
9. Black Boxes
10. Instantiated Netlists

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

+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs              |   52 |     0 |          0 |     20800 |  0.25 |
|   LUT as Logic          |   52 |     0 |          0 |     20800 |  0.25 |
|   LUT as Memory         |    0 |     0 |          0 |      9600 |  0.00 |
| Slice Registers         |   35 |     0 |          0 |     41600 |  0.08 |
|   Register as Flip Flop |   35 |     0 |          0 |     41600 |  0.08 |
|   Register as Latch     |    0 |     0 |          0 |     41600 |  0.00 |
| F7 Muxes                |    0 |     0 |          0 |     16300 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |      8150 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
```

2. Slice Logic Distribution

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice	26	0	0	8150	0.32
SLICEL	21	0			
SLICEM	5	0			
LUT as Logic	52	0	0	20800	0.25
using O5 output only	0				
using O6 output only	46				
using O5 and O6	6				
LUT as Memory	0	0	0	9600	0.00
LUT as Distributed RAM	0	0			
LUT as Shift Register	0	0			
Slice Registers	35	0	0	41600	0.08
Register driven from within the Slice	35				
Register driven from outside the Slice	0				
Unique Control Sets	2		0	8150	0.02

4. DSP

Site Type	Used	Fixed	Prohibited	Available	Util%
DSPs	0	0	0	90	0.00

THANK YOU