**EEL2024 Digital Design Lab Report**

**Sem II AY 2023-24**

|  |  |  |
| --- | --- | --- |
| **Experiment No.** | **:** | **05** |
| **Name** | **:** | **Lavangi Parihar** |
| **Roll No.** | **:** | **B22EE044** |
| **Partner Name (Roll Number)** | **:** | **Mansi Choudhary(B22EE045)** |

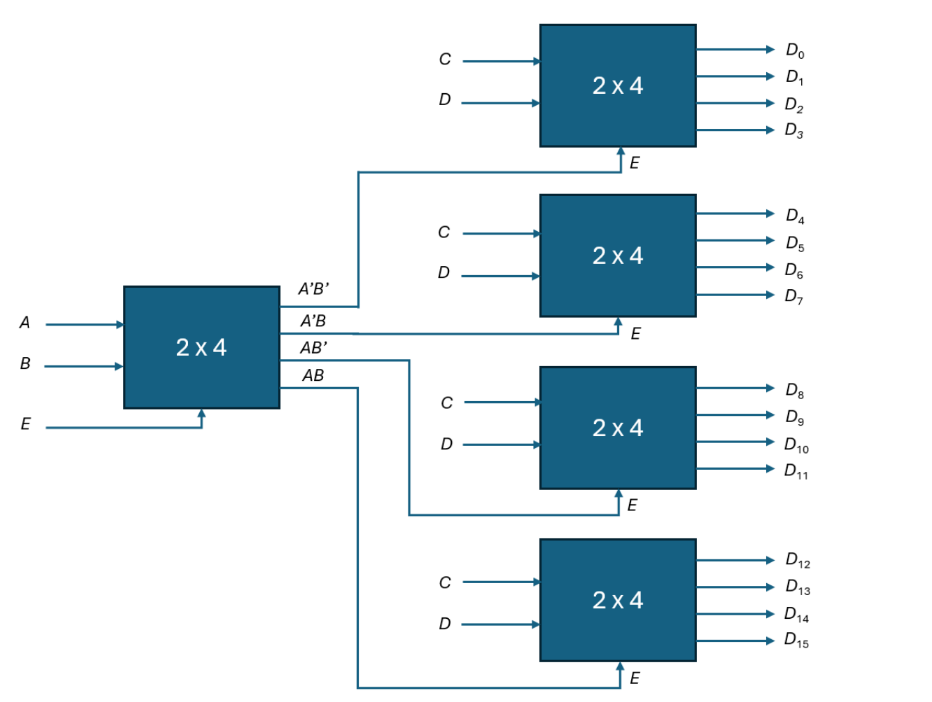
# Objectives

# (i) Design and simulate a 4 × 16 decoder using 2 × 4 decoders (with Enable)

# (ii) Design, simulate, and implement a Priority Encoder

# (iii) Design, simulate, and implement a Binary-to-Gray Code Converter

# Logic Design

Decoder 4 x 16 using 2 x 4 Decoder with Enable Input  
  


# Source Description

## - Design source

module decoder\_2x4(

input [1:0] I,

input enable,

output reg[3:0] Y

);

always @\* begin

if(enable) begin

case(I)

2'b00: Y = 4'b0001;

2'b01: Y = 4'b0010;

2'b10: Y = 4'b0100;

2'b11: Y = 4'b1000;

default : Y= 4'b0000;

endcase

end else begin

Y = 4'b0000;

end

end

endmodule

module decoder\_4x16(input A,B,C,D, output [15:0] O);

wire [3:0] Y\_enable, Y1,Y3,Y3,Y4;

decoder\_2x4 d\_enable({A, B}, 1, Y\_enable);

decoder\_2x4 d1({C, D}, Y\_enable[0], Y1);

decoder\_2x4 d2({C, D}, Y\_enable[1], Y2);

decoder\_2x4 d3({C, D}, Y\_enable[2], Y3);

decoder\_2x4 d4({C, D}, Y\_enable[3], Y4);

assign O = {Y4, Y3, Y2, Y1};

endmodule

## - Constraint file **PYNQ-Z2 v1.0.xdc**

|  |  |
| --- | --- |
| Variable | port |
| O[6] | LED3 |
| O[7] | LED2 |
| O[8] | LED1 |
| O[9] | LED0 |
| O[10] | R-LED4 |
| O[11] | G-LED4 |
| O[12] | B-LED4 |
| O[13] | R-LED5 |
| O[14] | G-LED5 |
| O[15] | B-LED5 |

## 

## 

**RPI\_Addon.xdc**

|  |  |  |
| --- | --- | --- |
|  | Variable | Port |
| Input | A[0] | SWH |
| Input | A[1] | SWG |
| Input | A[2] | SWF |
| Input | A[3] | SWE |
| Output | O[0] | LDF |
| Output | O[1] | LDE |
| Output | O[2] | LDD |
| Output | O[3] | LDC |
| Output | O[4] | LDB |
| Output | O[5] | LDA |

## 

## - Simulation source module decoder\_tb();

## reg A, B, C, D;

## wire[15:0] O;

## decoder\_4x16 dut(A, B, C, D, O);

## initial begin

## {A, B, C, D} = 4'b0001;

## #10;

## {A, B, C, D} = 4'b0011;

## #10;

## {A, B, C, D} = 4'b1001;

## #10;

## {A, B, C, D} = 4'b1111;

## #10;

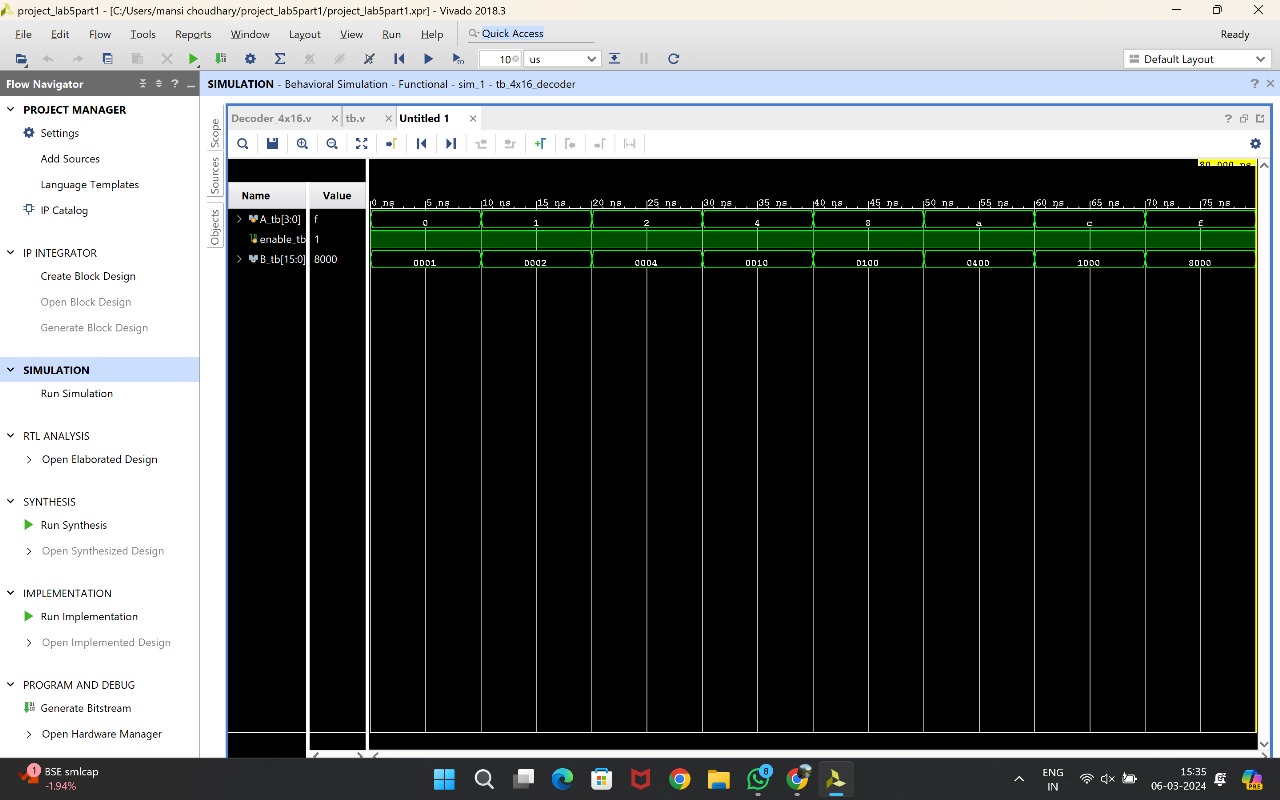
## $finish;

## end

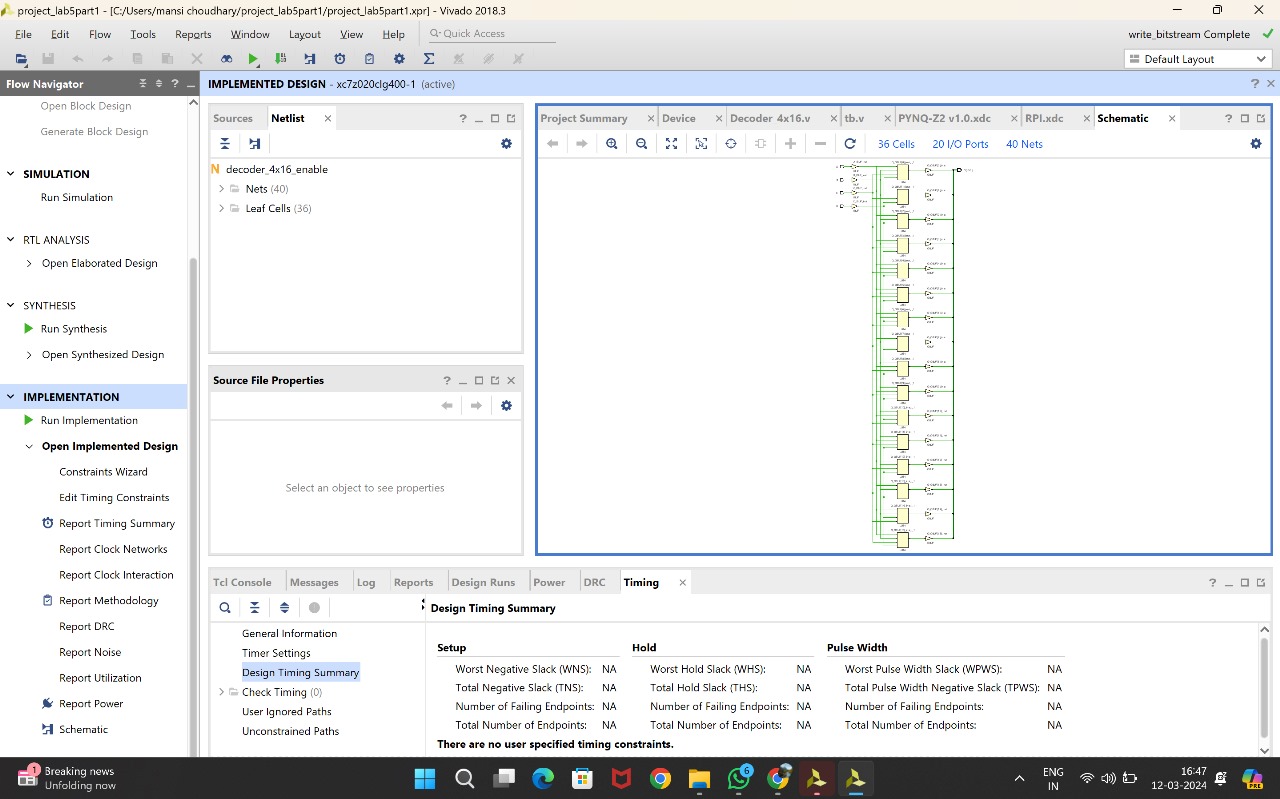
## endmodule

## 

Simulation Results (Timing diagram)



(Elaborated Design)



**PYNQ Working Video** (to be recorded during lab session)

**(with voiceover briefly explaining the working, not exceeding 1-2 minutes)**

[Video Link](https://drive.google.com/file/d/1E_ylTu1CLvPBNPWjybikeJREZGyBEMsC/view?usp=drive_link)

# PART II 4-BIT PRIORITY ENCODER

# Logic Design

# 4-bit Priority Encoder Shows the logic diagram of 4 bit priority encoder which consists two 2... | Download Scientific Diagram

**Source Description**- Design source

# Priority.v

# `timescale 1ns / 1ps

# module priority\_encoder(input [3:0] I, output reg [1:0] Y, output reg V);

# always @ (I)

# begin

# casex(I)

# 4'b0000: {Y, V} = 3'bxx0;

# 4'b1000: {Y, V} = 3'b001;

# 4'bx100: {Y, V} = 3'b011;

# 4'bxx10: {Y, V} = 3'b101;

# 4'bxxx1: {Y, V} = 3'b111;

# //default: {Y, V} = 3'b111; Assuming default means all 1's

# endcase

# end

endmodule

# - Constraint file PYNQ-Z2 v1.0.xdc

OUTPUTS ARE SHOWN

|  |  |
| --- | --- |
| VARIABLE | PORT |
| Y[0] | LED3 |
| Y[1] | LED2 |
| V | LED0 |

# 

RPI\_Addon.xdc

|  |  |
| --- | --- |
| Variable | Switches |
| I[0] | SWE |
| I[1] | SWF |
| I[2] | SWG |
| I[3] | SWH |

INPUTS ARE GIVEN

- Simulation source

module priority\_encoder\_4bit\_tb;

reg [3:0] I;

wire V;

wire [1:0] Y;

priority\_encoder pe(

.I(I),

.Y(Y),

.V(V)

);

initial begin

I= 4'b0100;

#10;

I= 4'b0001;

#10;

I= 4'b1100;

#10;

I= 4'b1010;

#10;

I= 4'b0011;

#10;

I= 4'b0000;

#10;

I= 4'b1000;

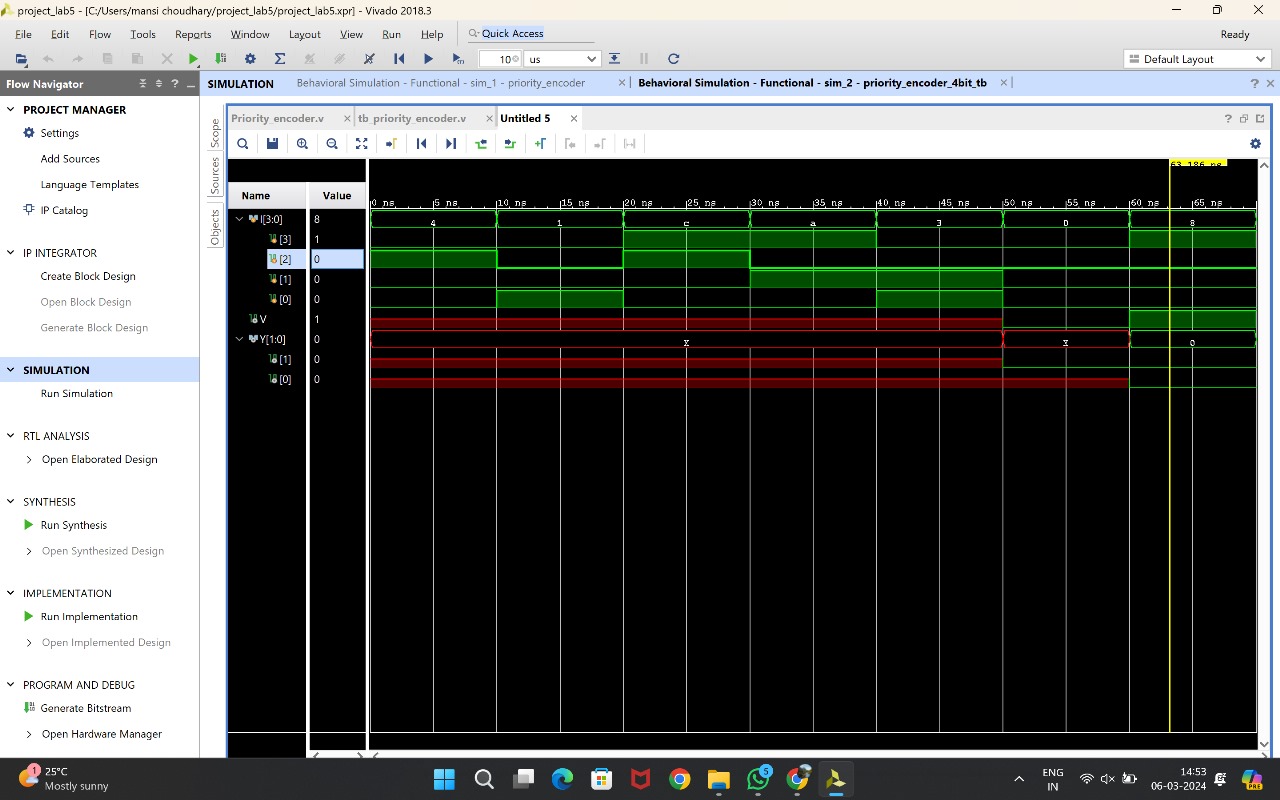
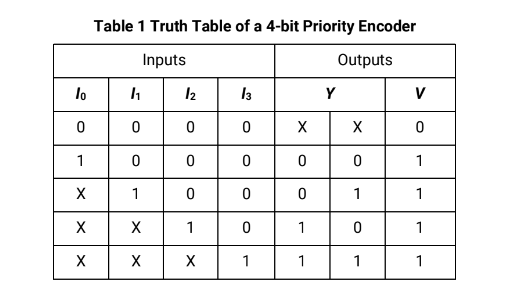
#10;

$stop; // End simulation

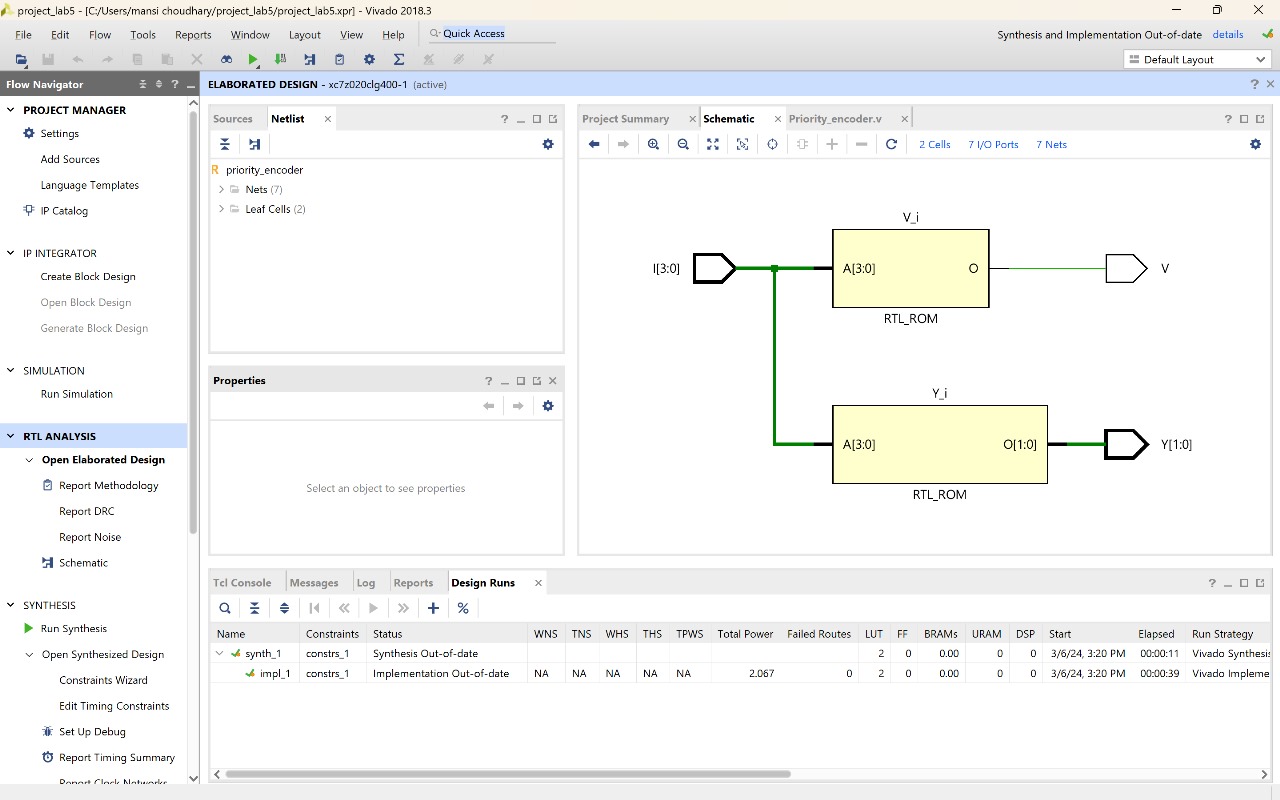
end

endmodule

# Simulation Results (Timing diagram)

(elaborated Design)



**PYNQ Working Video** (to be recorded during lab session)

**(with voiceover briefly explaining the working, not exceeding 1-2 minutes)**

[Video Link](https://drive.google.com/file/d/1Ee-d-1MKOMN0aZ59wwqi8hUPGu_JzLch/view?usp=drive_link)

# PART III 4.BIT BINARY-TO-GRAY-CONVERTER

# Logic Design

# 4-bit Binary-to-Gray Converter

**Source Description**- Design source

`timescale 1ns / 1ps

module binary\_to\_gray( input [3:0] binary, output reg [3:0] gray );  
always @\*  
begin

gray[3] = binary[3];  
 gray[2] = binary[3] ^ binary[2];  
 gray[1] = binary[2] ^ binary[1];  
 gray[0] = binary[1] ^ binary[0];

end

endmodule

- Constraint file

PYNQ-Z2 v1.0.xdc

|  |  |
| --- | --- |
| Variable | Port |
| G[0] | LED3 |
| G[1] | LED2 |
| G[2] | LED1 |
| G[3] | LED0 |

RPI ADDON.Xdc

|  |  |
| --- | --- |
| Variable | Switch |
| B[0] | SWH |
| B[1] | SWG |
| B[2] | SWF |
| B[3] | SWE |

**- Simulation source**   
module binary\_to\_gray(

input [3:0] binary,

output reg [3:0] gray

);

always @(\*) begin

gray[0] = binary[0];

gray[1] = binary[0] ^ binary[1];

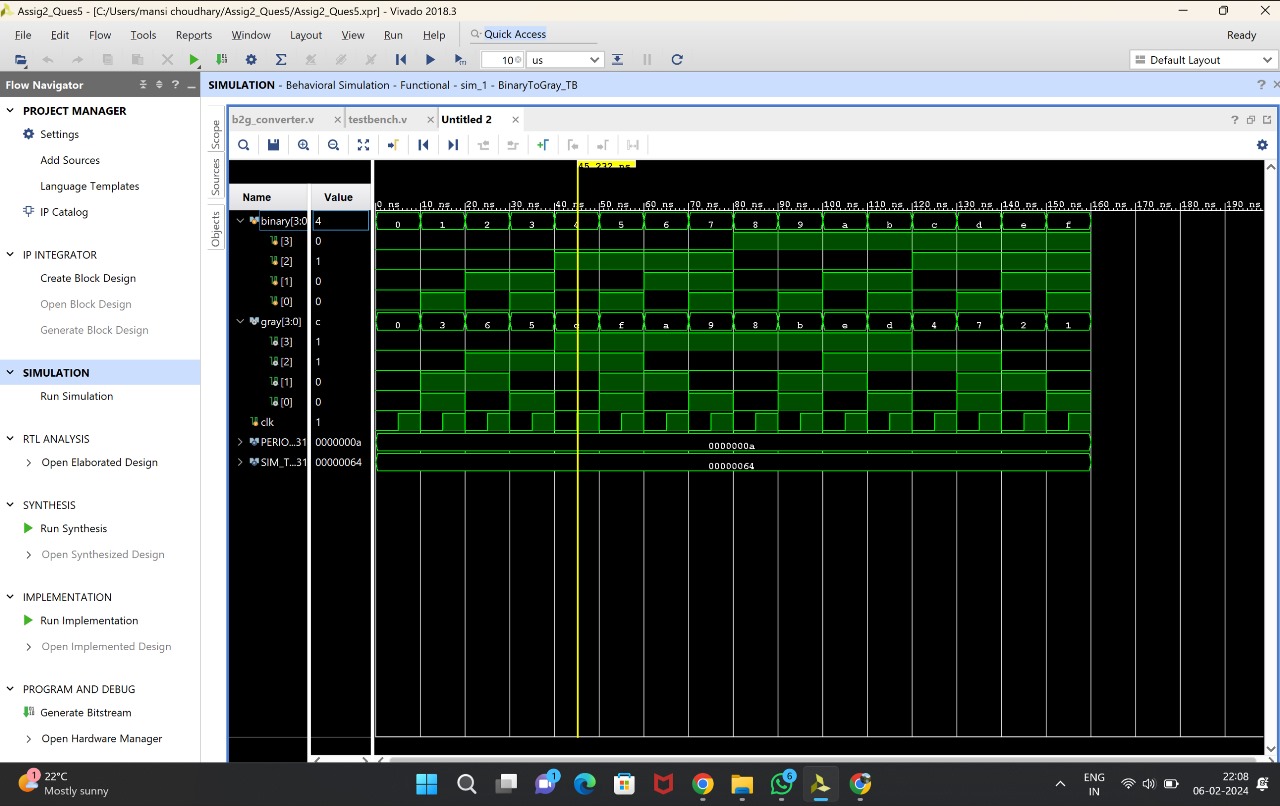
gray[2] = binary[1] ^ binary[2];

gray[3] = binary[2] ^ binary[3];

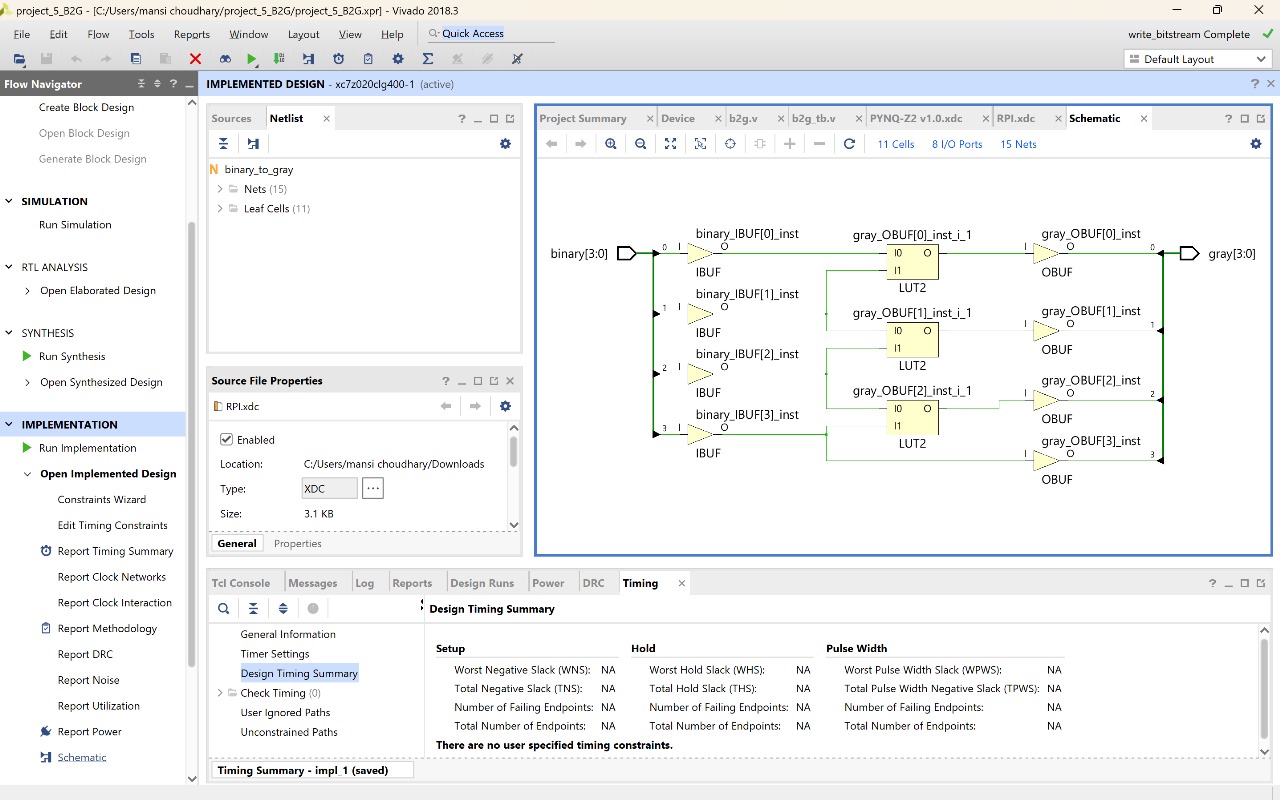
end

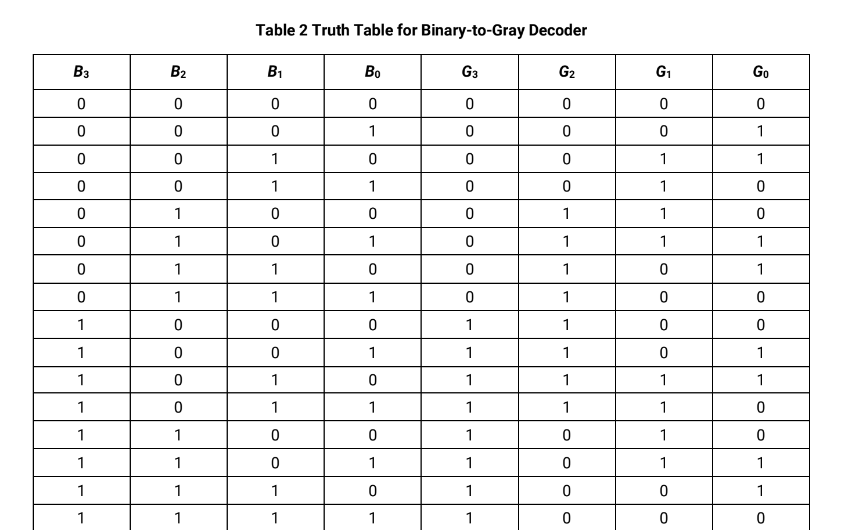
endmodule

# Simulation Results (Timing diagram)



**(Elaborated design)**

****

****

**PYNQ Working Video** (to be recorded during lab session)

**(with voiceover briefly explaining the working, not exceeding 1-2 minutes)**

[**Video Link**](https://drive.google.com/file/d/1EfqEh0ZBq-uJTTBOTIxkMdpwPkPeQACW/view?usp=drive_link)