Lab 2 Report

Name: Dhruv Sandesara

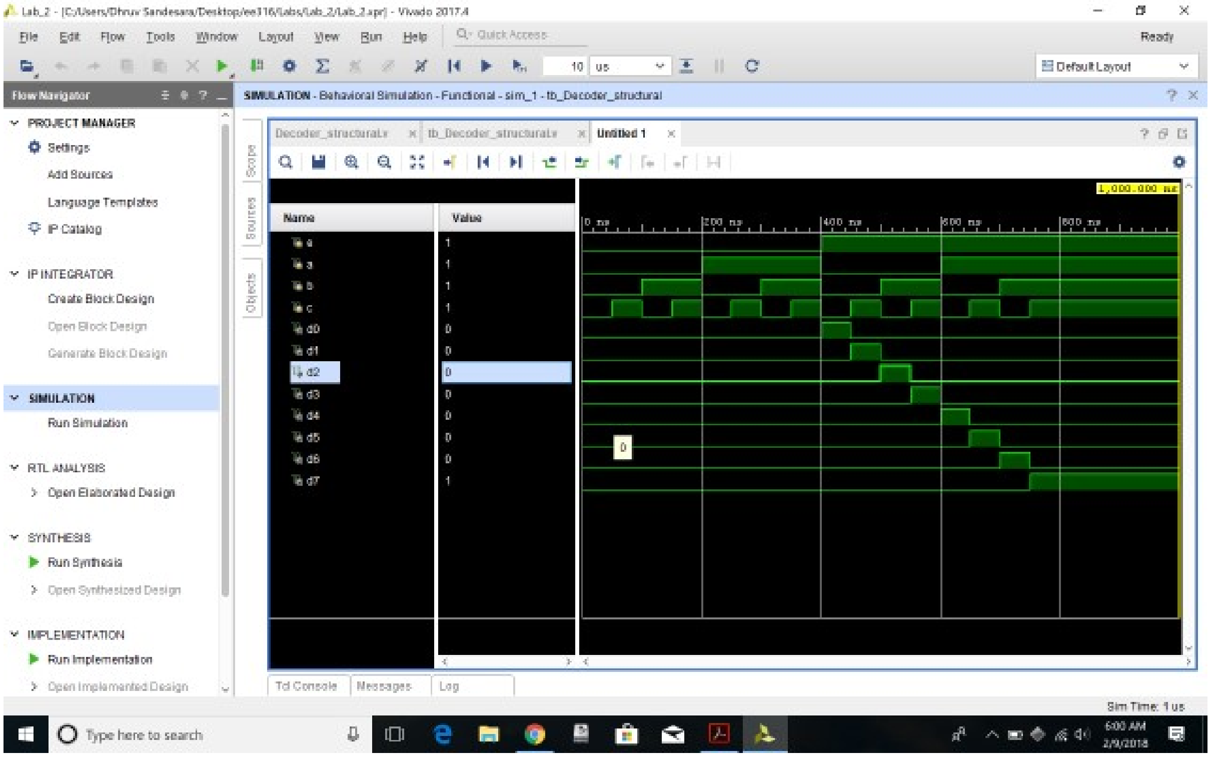
UT EID: djs3967

Section: 15295 (Lab Wednesday 12-1)

Checklist:

Part 1 –

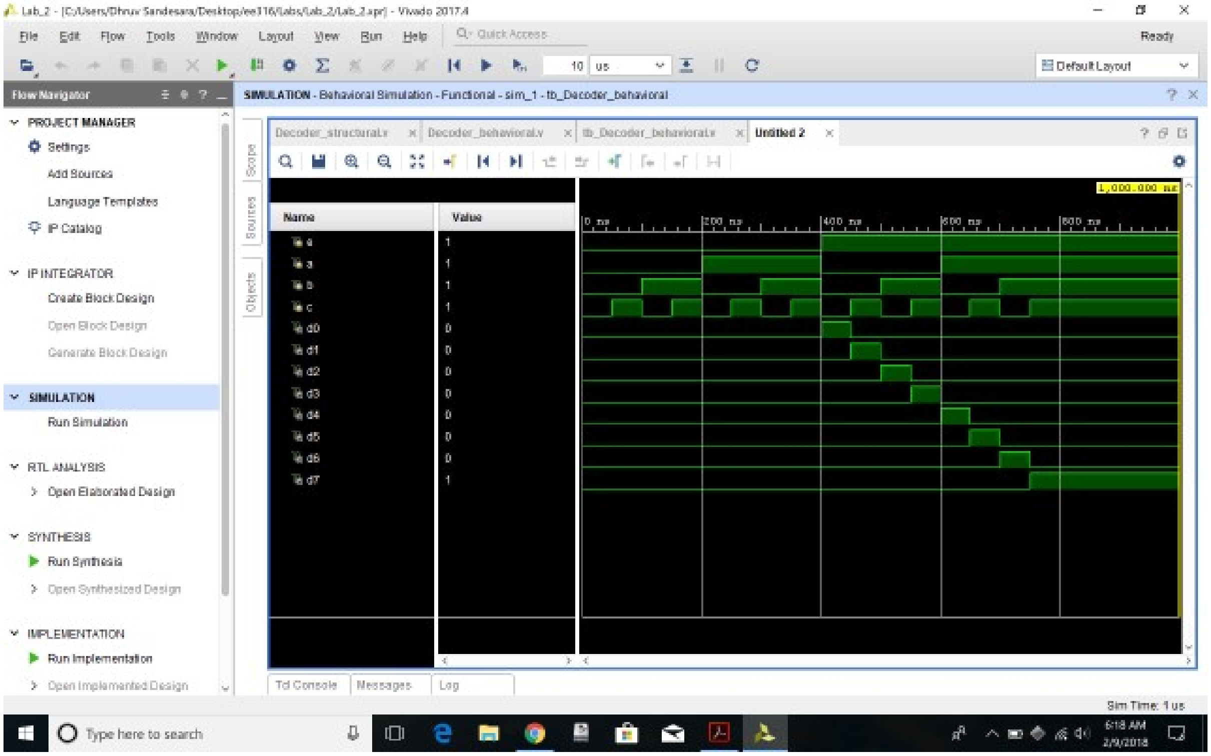
i. Simulation waveforms for Part 1 for Structural as well as Behavioral modelling (Screenshots)



Lab_2 - [C:/Users/Dhruv Sandesara/Desktop/ee316/Labs/Lab_2/Lab_2.xpr] - Vivado 2017.4

SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_Decoder_structural

Name	Value
e	1
a	1
b	1
c	1
d0	0
d1	0
d2	0
d3	0
d4	0
d5	0
d6	0
d7	1



Lab_2 - [C:/Users/Dhruv Sandesara/Desktop/ee316/Labs/Lab_2/Lab_2.xpr] - Vivado 2017.4

SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_Decoder_behavioral

Name	Value
e	1
a	1
b	1
c	1
d0	0
d1	0
d2	0
d3	0
d4	0
d5	0
d6	0
d7	1

Sim Time: 1 us

Part 2 –

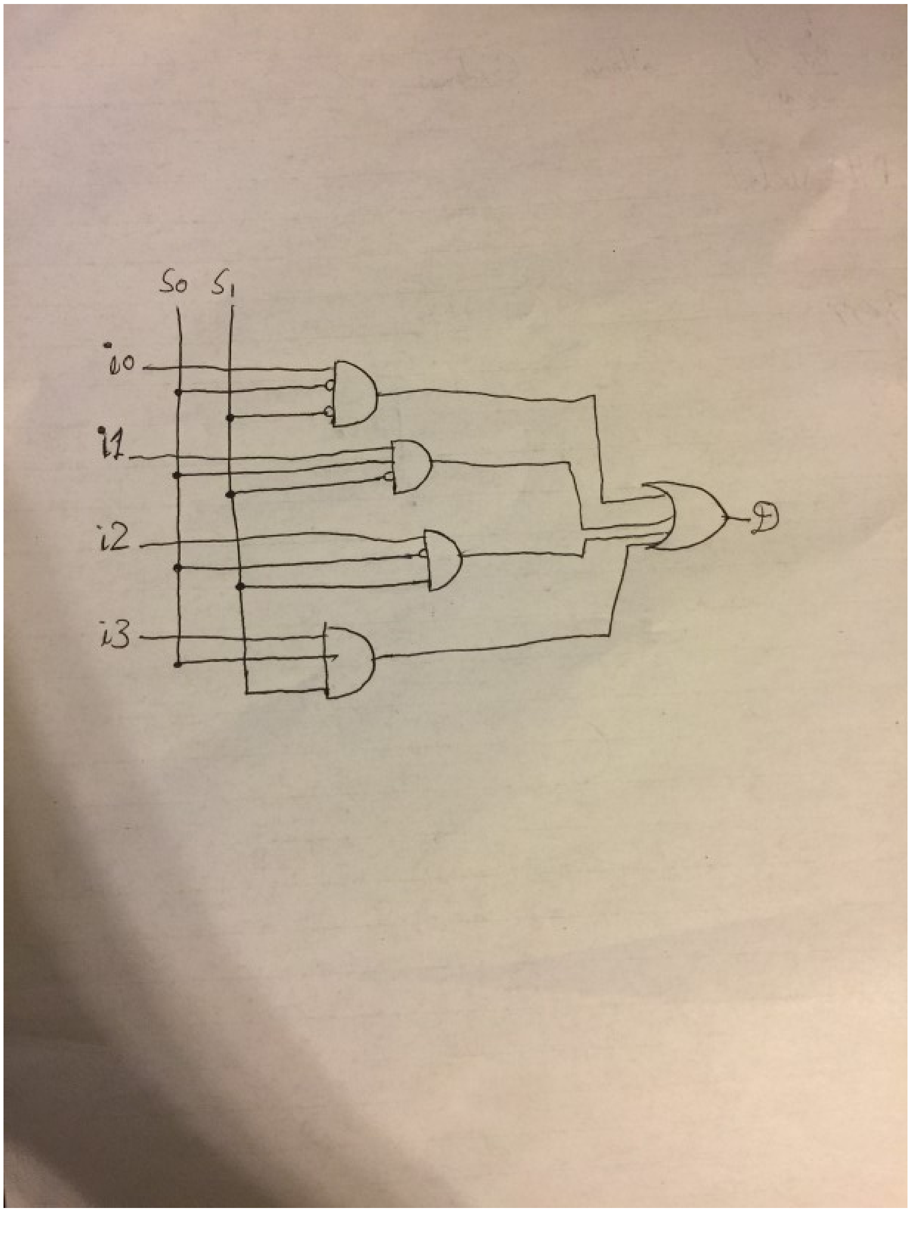
ii. Truth table of the function

I0	I1	I2	I3	S1	S0	D
0	X	X	X	0	0	0
1	X	X	X	0	0	1
X	0	X	X	0	1	0
X	1	X	X	0	1	1
X	X	0	X	1	0	0
X	X	1	X	1	0	1
X	X	X	0	1	1	0
X	X	X	1	1	1	1

iii. Algebraic expression of the logic function

D= (i0*s1’*s0’)+(i1*s1’*s0)+(i2*s1*s0’)+(i3*s1*S0)

iv. Logic circuit schematic



v. Verilog codes for module and testbench for structural modelling

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/09/2018 06:25:10 AM
// Design Name:
// Module Name: Multiplexer_structural
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
module Multiplexer_structural(
    input i0,
    input i1,
    input i2,
    input i3,
    input s0,
    input s1,
    output d
    );
    //defining the not values

    wire s0_not, s1_not;

    not n0(s0_not, s0);
    not n1(s1_not, s1);

    // intializing the and wires

    wire d0_and, d1_and, d2_and, d3_and;
    //intialize and gates
    and g0(d0_and, i0, s1_not, s0_not);
    and g1(d1_and, i1, s1_not, s0);
    and g2(d2_and, i2, s1, s0_not);
    and g3(d3_and, i3, s1, s0);

    //final or gate
    or o0(d,d0_and, d1_and, d2_and, d3_and);



endmodule
```

Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/09/2018 06:33:30 AM
// Design Name:
// Module Name: tb_Multiplexer_structural
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_Multiplexer_structural(

    );

    reg i0;
    reg i1;
    reg i2;
    reg i3;
    reg s0;
    reg s1;
    wire d;

    Multiplexer_structural uut(
        .i0(i0),
        .i1(i1),
        .i2(i2),
        .i3(i3),
        .s0(s0),
        .s1(s1),
        .d(d)
    );

    initial begin
```

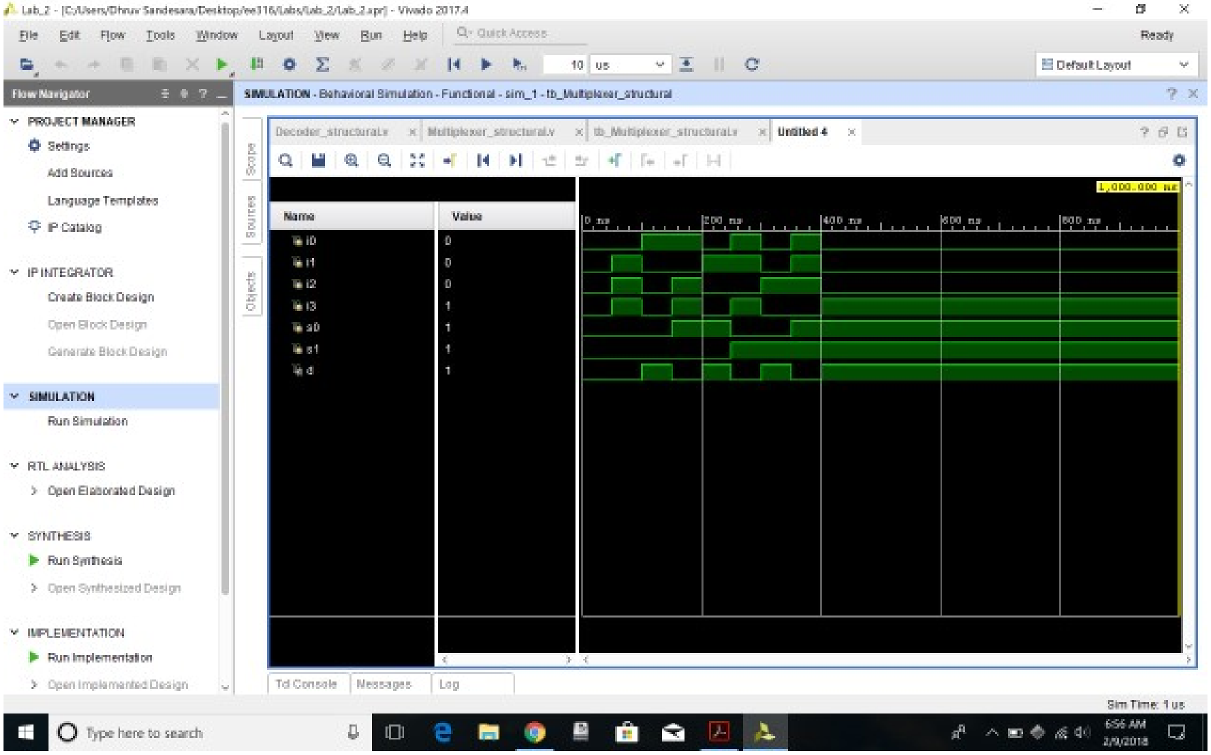
```
//initalize inputs
i0=0;
i1=0;
i2=0;
i3=0;
s0=0;
s1=0;

//wait for global reset to finish
#50;

//test cases begin

i0=0;
i1=1;
i2=1;
i3=1;
s0=0;
s1=0;
#50;
$display("TC01");
if(d!=1'b0)  $display("Resumlt is wrong");

i0=1;
i1=0;
i2=0;
i3=0;
s0=0;
```

```
s1=0;
#50
$display("TC02");
if(d!=1'b1)  $display("Resumlt is wrong");



i0=1;
i1=0;
i2=1;
i3=1;
s0=1;
s1=0;
#50
$display("TC03");
if(d!=1'b0)  $display("Resumlt is wrong");

i0=0;
i1=1;
i2=0;
i3=0;
s0=1;
s1=0;
#50
$display("TC04");
if(d!=1'b1)  $display("Resumlt is wrong");
```

```
i0=1;
i1=1;
i2=0;
i3=1;
s0=0;
s1=1;
#50
$display("TC05");
if(d!=1'b0)  $display("Resumlt is wrong");

i0=0;
i1=0;
i2=1;
i3=0;
s0=0;
s1=1;
#50
$display("TC06");
if(d!=1'b1)  $display("Resumlt is wrong");



i0=1;
i1=1;
i2=1;
i3=0;
s0=1;
```

```
    s1=1;
    #50
    $display("TC07");
    if(d!=1'b0)  $display("Resumlt is wrong");

    i0=0;
    i1=0;
    i2=0;
    i3=1;
    s0=1;
    s1=1;
    #50
    $display("TC08");
    if(d!=1'b1)  $display("Resumlt is wrong");
    end
endmodule
```

vi. Simulation waveform for structural modelling (Screenshot)



Name	Value
i0	0
i1	0
i2	0
i3	1
s0	1
s1	1
d	1

vii. Verilog codes for module and testbench for behavioral modelling

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/09/2018 06:59:02 AM
// Design Name:
// Module Name: Multiplexer_behavioral
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module Multiplexer_behavioral(
    input i0,
    input i1,
    input i2,
    input i3,
    input s0,
    input s1,
    output reg d
    );

    always@(i0,i1,i2,i3,s0,s1)
    begin

        case({s1,s0})
        2'b00:d=i0;
        2'b01:d=i1;
        2'b10:d=i2;
        2'b11:d=i3;
        endcase




    end



endmodule
```

TEST MODULE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/09/2018 07:14:45 AM
// Design Name:
// Module Name: tb_Multiplexer_behavioral
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
module tb_Multiplexer_behavioral(

    );

    reg i0;
    reg i1;
    reg i2;
    reg i3;
    reg s0;
    reg s1;
    wire d;

    Multiplexer_behavioral uut(
        .i0(i0),
        .i1(i1),
        .i2(i2),
        .i3(i3),
        .s0(s0),
        .s1(s1),
        .d(d)
    );

    initial begin
    //initalize inputs
    i0=0;
    i1=0;
    i2=0;
```

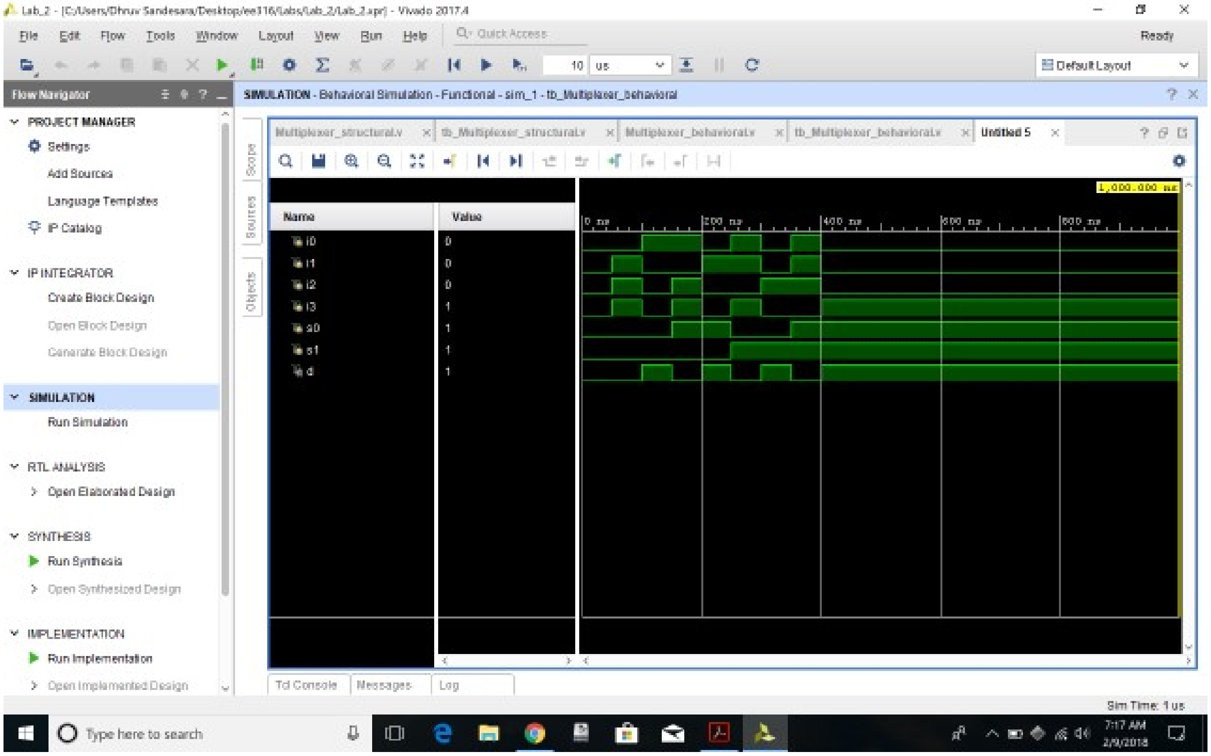
```
i3=0;
s0=0;
s1=0;

//wait for global reset to finish
#50;

//test cases begin

i0=0;
i1=1;
i2=1;
i3=1;
s0=0;
s1=0;
#50;
$display("TC01");
if(d!=1'b0)  $display("Resumlt is wrong");

i0=1;
i1=0;
i2=0;
i3=0;
s0=0;
s1=0;
#50
$display("TC02");
if(d!=1'b1)  $display("Resumlt is wrong");
```

```
i0=1;
i1=0;
i2=1;
i3=1;
s0=1;
s1=0;
#50
$display("TC03");
if(d!=1'b0)  $display("Resumlt is wrong");

i0=0;
i1=1;
i2=0;
i3=0;
s0=1;
s1=0;
#50
$display("TC04");
if(d!=1'b1)  $display("Resumlt is wrong");



i0=1;
i1=1;
i2=0;
```

```
i3=1;
s0=0;
s1=1;
#50
$display("TC05");
if(d!=1'b0)  $display("Resumlt is wrong");

i0=0;
i1=0;
i2=1;
i3=0;
s0=0;
s1=1;
#50
$display("TC06");
if(d!=1'b1)  $display("Resumlt is wrong");



i0=1;
i1=1;
i2=1;
i3=0;
s0=1;
s1=1;
#50
$display("TC07");
if(d!=1'b0)  $display("Resumlt is wrong");
```

```
    i0=0;
    i1=0;
    i2=0;
    i3=1;
    s0=1;
    s1=1;
    #50
    $display("TC08");
    if(d!=1'b1)  $display("Resumlt is wrong");


    end
  endmodule
```

viii. Simulation waveform for behavioral modelling (Screenshot)



Name	Value
i0	0
i1	0
i2	0
i3	1
s0	1
s1	1
d	1

Note *–> The Verilog codes should be copied in your lab report, and the actual Verilog (.v) files need to be zipped and submitted as well on Canvas. You are not allowed to change your Verilog codes after final submission as the TAs may download the submitted codes from Canvas during checkouts. For the truth table, algebraic expression and circuit schematic, you are free to draw it on paper and then put the pictures in your lab report, but please make sure it is legible for the TAs to grade it properly.*