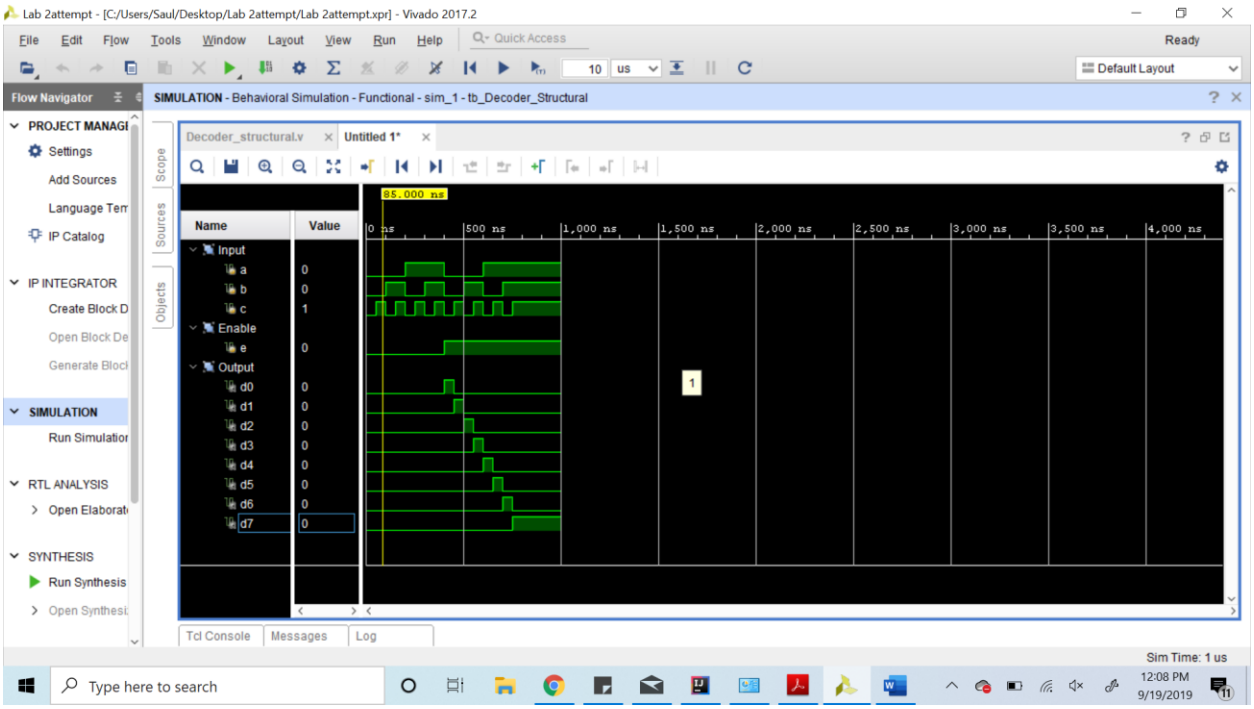
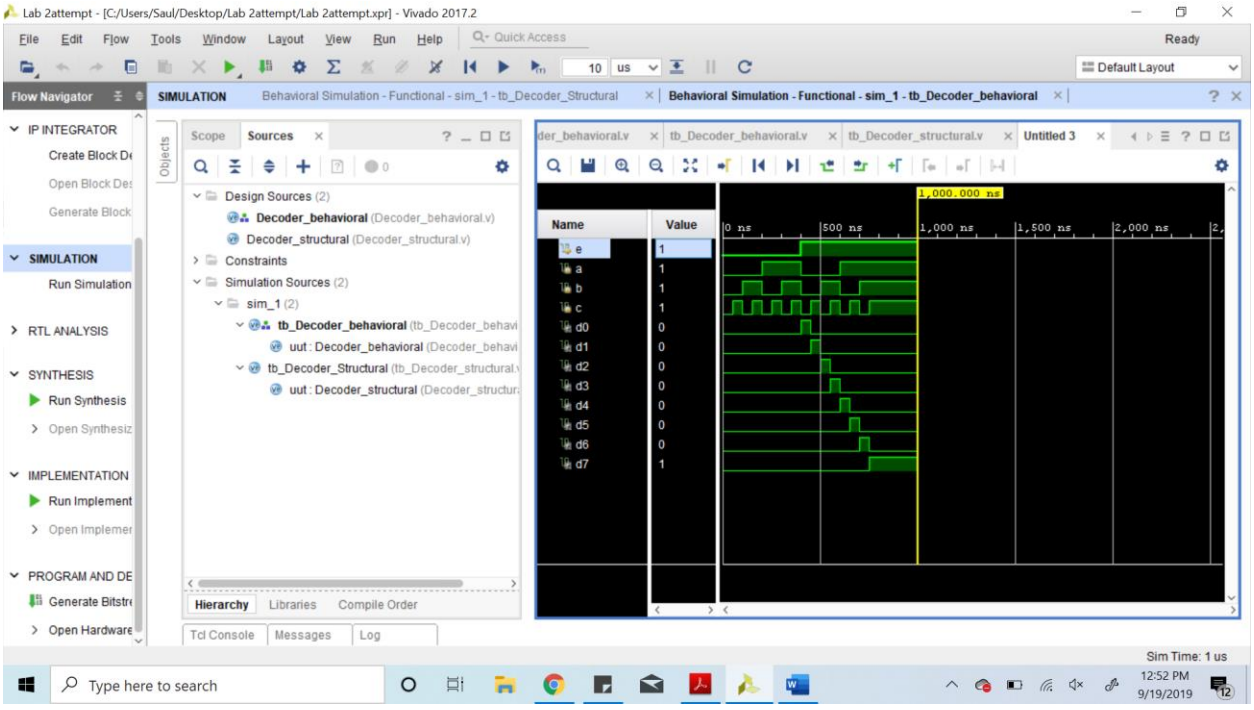
Part 1

Decoder_Structural



Decoder_behavioral



Part 2

Design Source: Mux_structural

```
`timescale 1ns / 1ps
module Mux_structural(
   input s0,
   input s1,
   input i0,
   input i1,
   input i2,
   input i3,
   output d
   );


   wire s0_not, s1_not;

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

   and g0 (d, s1_not, s0_not);
   and g1 (d, s1_not, s0);
   and g2 (d, s1, s0_not);
   and g3 (d, s1, s0);


endmodule
```

Testbench: tb_Mux_Structural

```
`timescale 1ns / 1ps
module tb_Mux_Structural;


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

    wire d;

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

    );

    initial begin

        i0 = 0;
        i1 = 0;
```

```
        i2 = 0;
        i3 = 0;
        s1 = 0;
        s0 = 0;

        #50;

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


        i0 = 0;
        i1 = 0;
        i2 = 0;
        i3 = 0;
        s1 = 0;
        s0 = 1;
        $display ("TC02");
        if (d != 1'b0 ) $display ("Result is wrong");
        #50;
```

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

        i0 = 0;
        i1 = 0;
        i2 = 0;
        i3 = 0;
        s1 = 1;
        s0 = 1;
        $display ("TC04");
        if (d !=  1'b0 ) $display ("Result is wrong");
        #50;
*/
        i0 = 1;
        i1 = 0;
        i2 = 0;
        i3 = 0;
        s1 = 0;
        s0 = 0;
        $display ("TC01");
        if (d !=  1'b1 ) $display ("Result is wrong");
        #50;
```

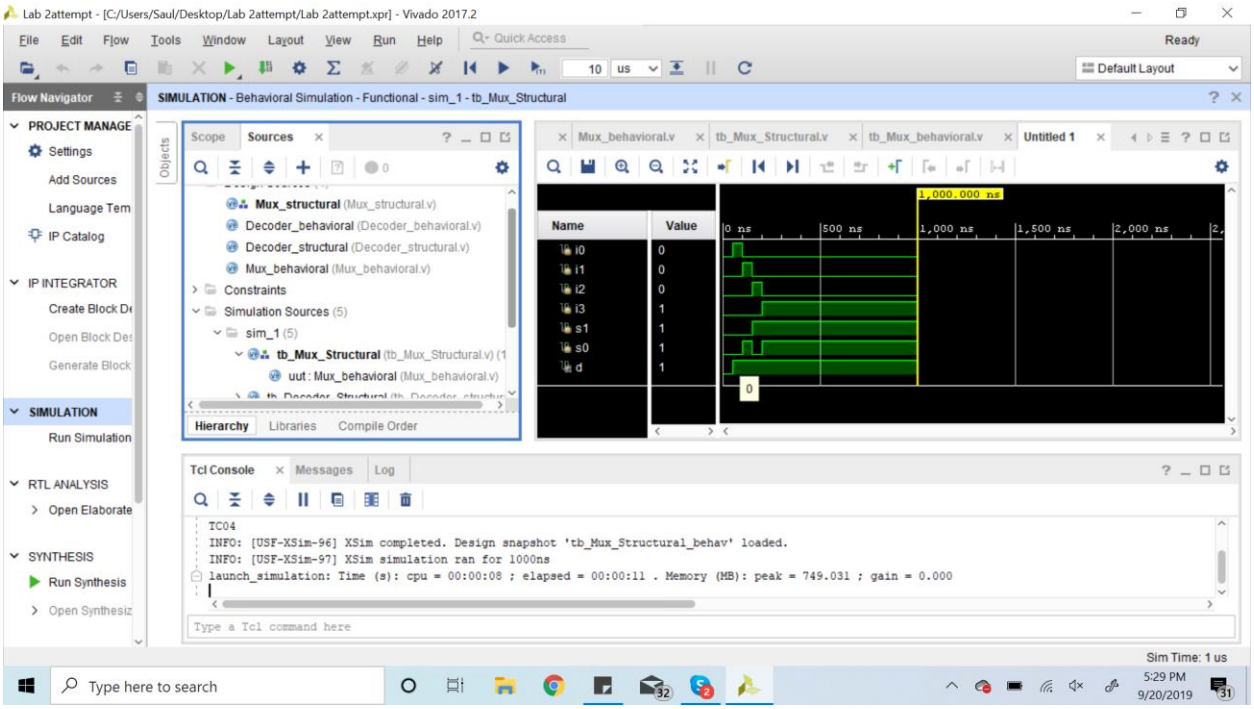
```
i0 = 0;
i1 = 1;
i2 = 0;
i3 = 0;
s1 = 0;
s0 = 1;
$display ("TC02");
if (d != 1'b1 ) $display ("Result is wrong");
#50;

i0 = 0;
i1 = 0;
i2 = 1;
i3 = 0;
s1 = 1;
s0 = 0;
$display ("TC03");
if (d !=  1'b1 ) $display ("Result is wrong");
#50;

i0 = 0;
i1 = 0;
i2 = 0;
i3 = 1;
s1 = 1;
s0 = 1;
```

```
        $display ("TC04");
        if (d !=  1'b1 ) $display ("Result is wrong");

        end
endmodule
```



<u>Design Source: Mux_Behavioral</u>

```
`timescale 1ns / 1ps
module Mux_behavioral(
    input i0,
    input i1,
    input i2,
    input i3,
    input s1,
    input s0,
    output reg d
```

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

    d = 1'b0;

    case ({i0,i1,i2,i3,s1,s0})
       6'b100000 : d = 1'b1;
       6'b010001 : d = 1'b1;
       6'b001010 : d = 1'b1;
       6'b000111 : d = 1'b1;

       default :   begin
                   d = 1'b0;
                   end
    endcase
    end
endmodule
```

Testbench: tb_Mux_behavioral

```
`timescale 1ns / 1ps

module tb_Mux_behavioral;

    reg i0;
    reg i1;
    reg i2;
    reg i3;
    reg s1;
```

```
    reg s0;

    wire d;

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

    );

    initial begin

        i0 = 0;
        i1 = 0;
        i2 = 0;
        i3 = 0;
        s1 = 0;
        s0 = 0;

        #50;
/*
        i0 = 0;
        i1 = 0;
        i2 = 0;
```

```
i3 = 0;
s1 = 0;
s0 = 0;
$display ("TC01");
if (d !=  1'b0 ) $display ("Result is wrong");
#50;


i0 = 0;
i1 = 0;
i2 = 0;
i3 = 0;
s1 = 0;
s0 = 1;
$display ("TC02");
if (d != 1'b0 ) $display ("Result is wrong");
#50;

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

i0 = 0;
```

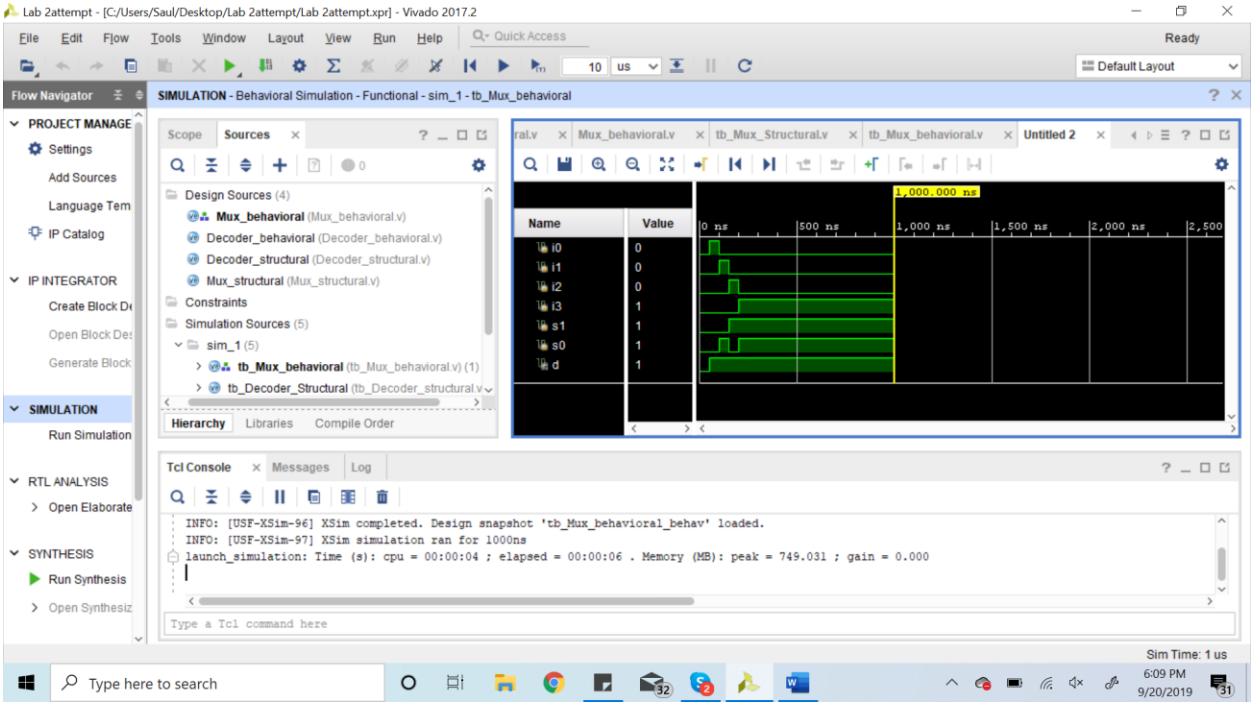
```
        i1 = 0;
        i2 = 0;
        i3 = 0;
        s1 = 1;
        s0 = 1;
        $display ("TC04");
        if (d !=  1'b0 ) $display ("Result is wrong");
        #50;
*/
        i0 = 1;
        i1 = 0;
        i2 = 0;
        i3 = 0;
        s1 = 0;
        s0 = 0;
        $display ("TC01");
        if (d !=  1'b1 ) $display ("Result is wrong");
        #50;

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

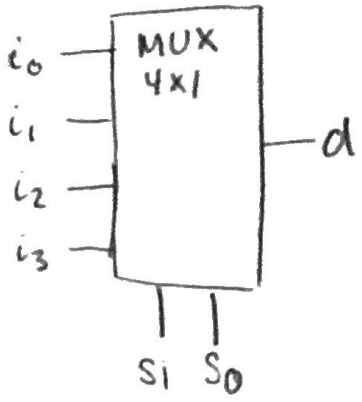
```
        i0 = 0;
        i1 = 0;
        i2 = 1;
        i3 = 0;
        s1 = 1;
        s0 = 0;
        $display ("TC03");
        if (d !=  1'b1 ) $display ("Result is wrong");
        #50;

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

        end
endmodule
```


Lab 2attempt - [C:/Users/Saul/Desktop/Lab 2attempt/Lab 2attempt.xpr] - Vivado 2017.2
File Edit Flow Tools Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_Mux_behavioral
PROJECT MANAGE
Settings
Add Sources
Language Tem
IP Catalog
IP INTEGRATOR
Create Block De
Open Block Des
Generate Block
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborate
SYNTHESIS
Run Synthesis
Open Synthesiz
Scope
Sources
Design Sources (4)
Mux_behavioral (Mux_behavioral.v)
Decoder_behavioral (Decoder_behavioral.v)
Decoder_structural (Decoder_structural.v)
Mux_structural (Mux_structural.v)
Constraints
Simulation Sources (5)
sim_1 (5)
tb_Mux_behavioral (tb_Mux_behavioral.v) (1)
tb_Decoder_Structural (tb_Decoder_structural.v
Hierarchy
Libraries
Compile Order
Mux_behavioral.v
tb_Mux_Structural.v
tb_Mux_behavioral.v
Untitled 2
1,000.000 ns
Name
Value
i0
i1
i2
i3
s1
s0
d
0 ns
500 ns
1,000 ns
1,500 ns
2,000 ns
2,500
Tcl Console
Messages
Log
INFO: [USF-XSim-96] XSim completed. Design snapshot 'tb_Mux_behavioral_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:04 ; elapsed = 00:00:06 . Memory (MB): peak = 749.031 ; gain = 0.000
Type a Tcl command here
Sim Time: 1 us
Type here to search
6:09 PM
9/20/2019

Part 2



Truth Table

S_1	S_0	d
0	0	i_0
0	1	i_1
1	0	i_2
1	1	i_3

Algebraic Expression

$$d = \bar{S}_1 \bar{S}_0 i_0 + \bar{S}_1 S_0 i_1 + S_1 \bar{S}_0 i_2 + S_1 S_2 i_3$$

Circuit