Lab 2 Report

Name: **Donald Maze-England**

UT EID: **dsm2588**

Section: **15320**

Checklist:

Part 1 –

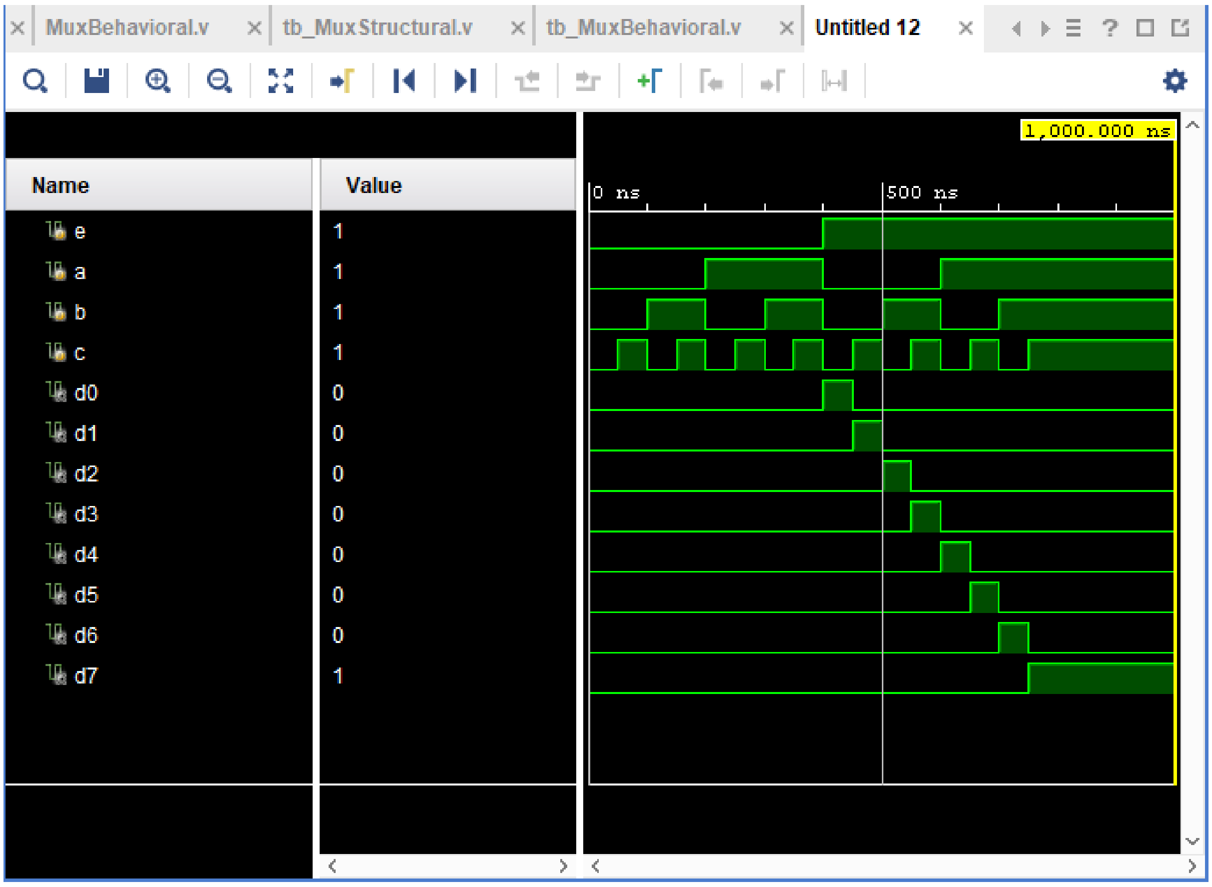
i. Simulation waveforms for Part 1 for Structural as well as Behavioral modelling (Screenshots)

Part 2 –

ii. Truth table of the function
iii. Algebraic expression of the logic function
iv. Logic circuit schematic
v. Verilog codes for module and testbench for structural modelling
vi. Simulation waveform for structural modelling (Screenshot)
vii. Verilog codes for module and testbench for behavioral modelling
viii. Simulation waveform for behavioral modelling (Screenshot)

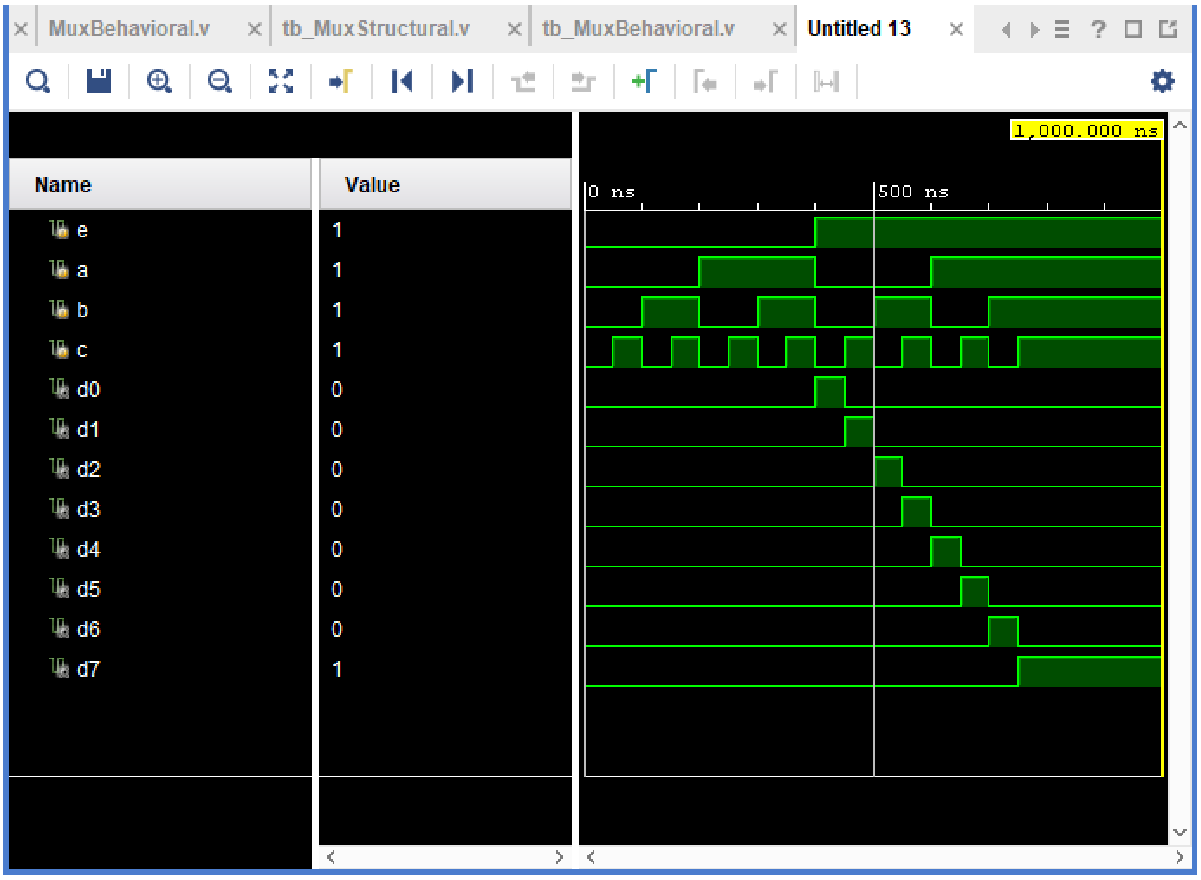
Note *–> The Verilog codes should be copied in your lab report, and the actual Verilog (.v) files need to be zipped and submitted as well on Canvas. You are not allowed to change your Verilog codes after final submission as the TAs may download the submitted codes from Canvas during checkouts. For the 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.*

Decoder Structural Simulation:



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

Decoder Behavioural Simulation:



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

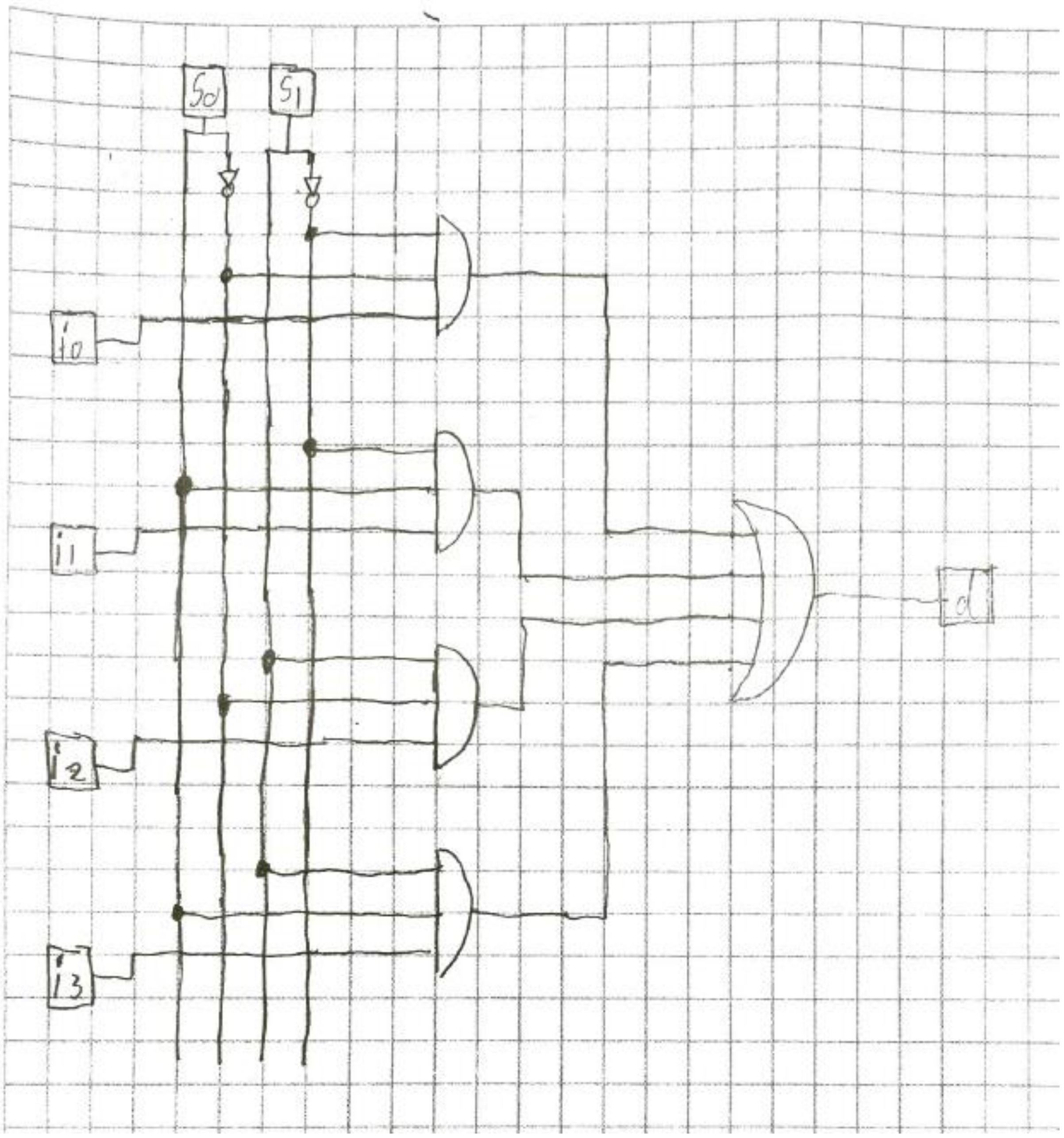
Truth Table for 4x1 MUX:

s1	s0	i0	i1	i2	i3	d
0	0	0	x	x	x	0
0	0	1	x	x	x	1
0	1	x	0	x	x	0
0	1	x	1	x	x	1
1	0	x	x	0	x	0
1	0	x	x	1	x	1
1	1	x	x	x	0	0
1	1	x	x	x	1	1

4x1 MUX Algebraic Expression:

d = (s0’s1’i0) + (s0s1’i1) + (s0’s1i2) + (s0s1i3)

4x1 Mux Schematic:



Verilog code for structural mux:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/06/2018 08:19:03 PM
// Design Name:
// Module Name: MuxStructural
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


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

    //defining wires for not signals
    wire s0_not, s1_not;

    //applying not logic gates to not wires
    not n0 (s0_not, s0);
    not n1 (s1_not, s1);

    //defining wires for and signals
    wire and0, and1, and2, and3;

    //instantiating and gates as per the schematic
    and g0 (and0, s0_not, s1_not, i0);
    and g1 (and1, s0, s1_not, i1);
    and g2 (and2, s0_not, s1, i2);
    and g3 (and3, s0, s1, i3);
    //instantiating or gate as per schematic

    or orGate (d, and0, and1, and2, and3);

endmodule
```

Testbench Code For Structural Mux:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/06/2018 08:56:38 PM
// Design Name:
// Module Name: tb_MuxStructural
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_MuxStructural;

    //inputs to be defined as registers
    reg s0;
    reg s1;
    reg i0;
    reg i1;
    reg i2;
    reg i3;

    //outputs to be defined as wires
    wire d;

    //Initiat the unit under test (UUT)
    MuxStructural uut (
        .s0(s0),
        .s1(s1),
        .i0(i0),
        .i1(i1),
        .i2(i2),
        .i3(i3),
        .d(d)
    );

    initial begin
    //initialze inputs
    s0 = 0;
    s1 = 0;
    i0 = 0;
    i1 = 1;
    i2 = 1;
    i3 = 1;

    #50; //wait 50 seconds for global reset to finish

    //stimulus - all input combinations followed by some wait time to observe the o/p

    $display ("TC01");
    if (d != 1'b0) $display ("Result is wrong");

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

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

```
#50
$display ("TC03");
if (d != 1'b0) $display ("Result is wrong");

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

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

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

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

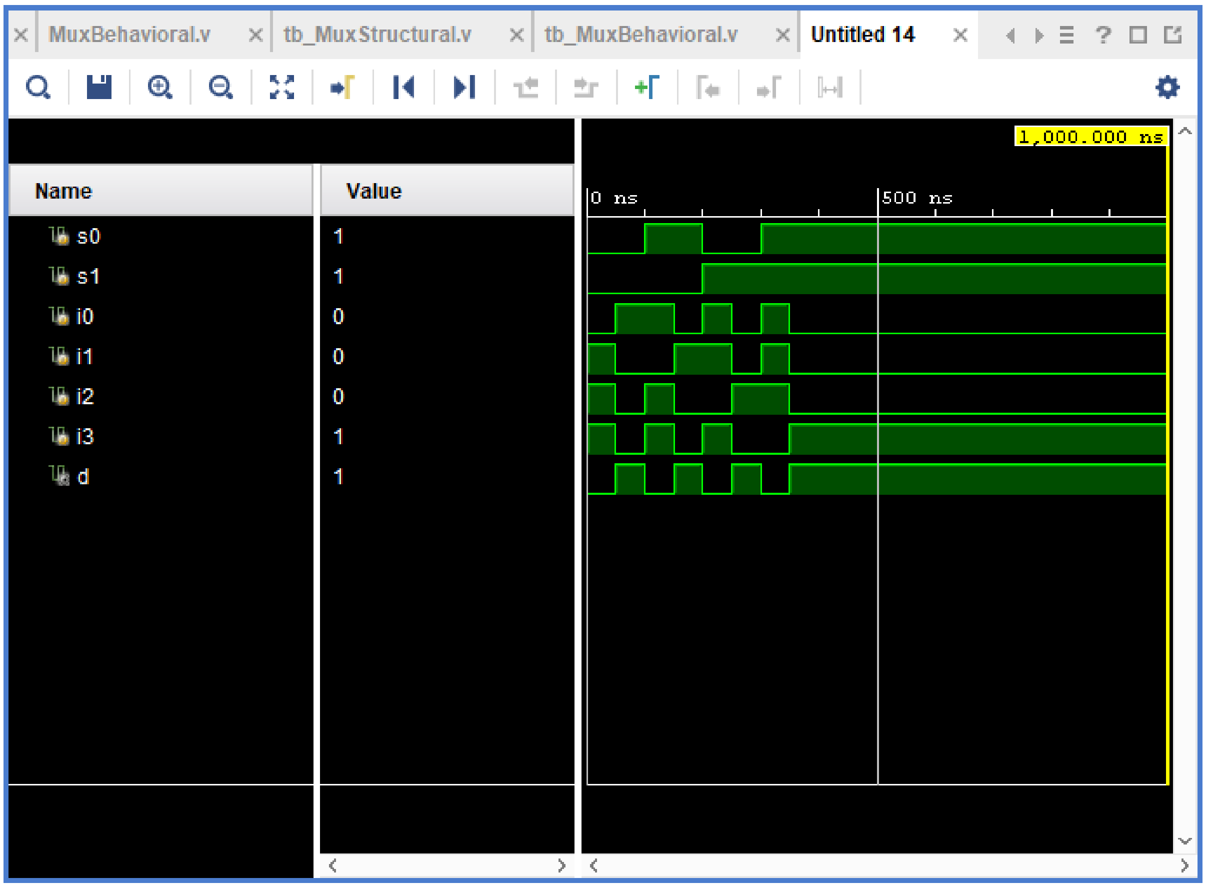
```
    #50
    $display ("TC07");
    if (d != 1'b0) $display ("Result is wrong");

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

    end

endmodule
```

Mux Structural Simulation:



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

Verilog Code For Behavioural Mux:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/06/2018 08:29:00 PM
// Design Name:
// Module Name: MuxBehavioral
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


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

    always@*
    begin

        case({s1,s0})
            2'b00 : d = i0;
            2'b01 : d = i1;
            2'b10 : d = i2;
            2'b11 : d = i3;
            default : d = 0;
        endcase

    end
endmodule
```

Testbench Code For Behavioural Mux:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/07/2018 10:04:28 AM
// Design Name:
// Module Name: tb_MuxBehavioral
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module tb_MuxBehavioral;

    //inputs to be defined as registers
    reg s0;
    reg s1;
    reg i0;
    reg i1;
    reg i2;
    reg i3;

    //outputs to be defined as wires
    wire d;

    //Initiat the unit under test (UUT)
    MuxBehavioral uut (
        .s0(s0),
        .s1(s1),
        .i0(i0),
        .i1(i1),
        .i2(i2),
        .i3(i3),
        .d(d)
    );

    initial begin
    //initialze inputs
    s0 = 0;
    s1 = 0;
    i0 = 0;
    i1 = 1;
    i2 = 1;
    i3 = 1;

    #50; //wait 50 seconds for global reset to finish

    //stimulus - all input combinations followed by some wait time to observe the o/p

    $display ("TC01");
    if (d != 1'b0) $display ("Result is wrong");

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

    s0 = 1;
    s1 = 0;
    i0 = 1;
    i1 = 0;
    i2 = 1;
    i3 = 1;
    #50
```

```
$display ("TC03");
if (d != 1'b0) $display ("Result is wrong");

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

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

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

s0 = 1;
s1 = 1;
i0 = 1;
i1 = 1;
i2 = 1;
i3 = 0;
#50
```

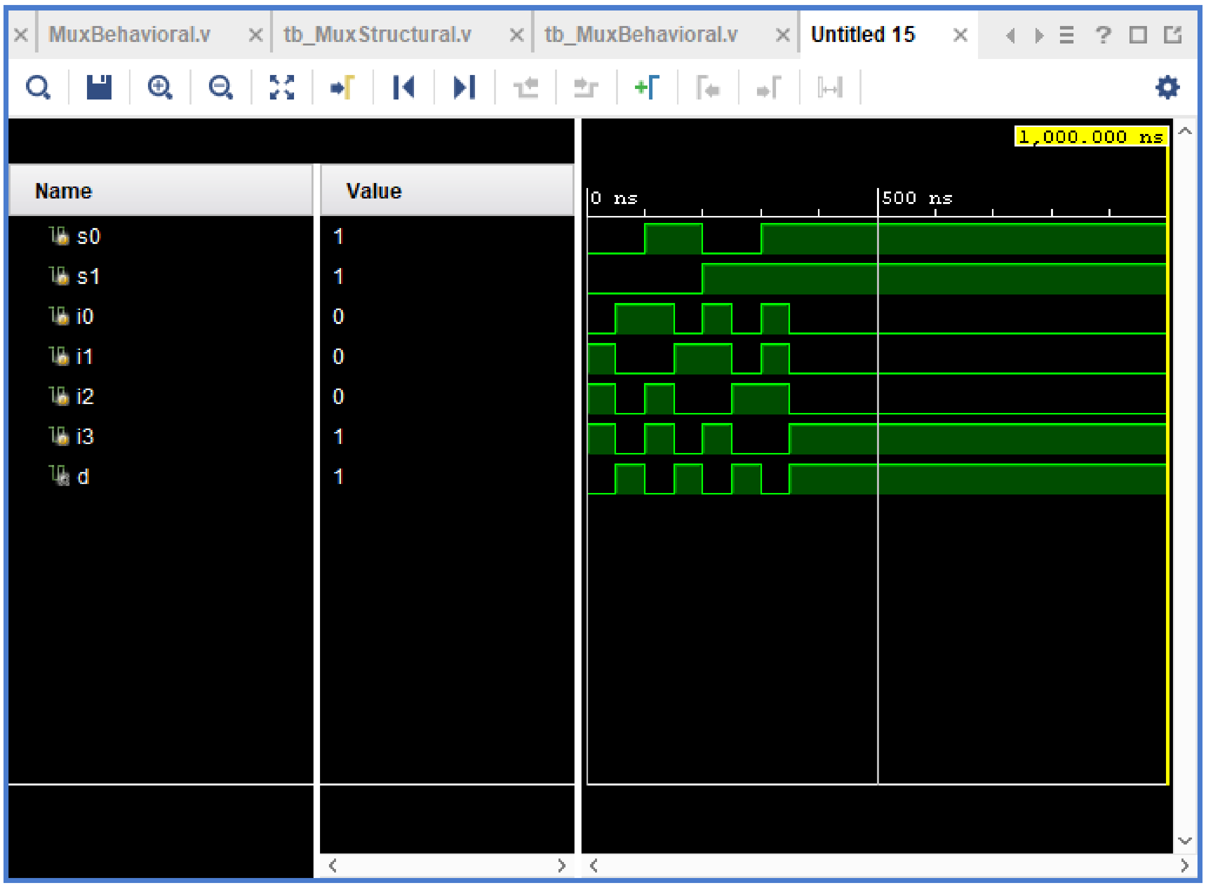
```
    $display ("TC07");
    if (d != 1'b0) $display ("Result is wrong");

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

    end

endmodule
```

Behavioural Mux Simulation:



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