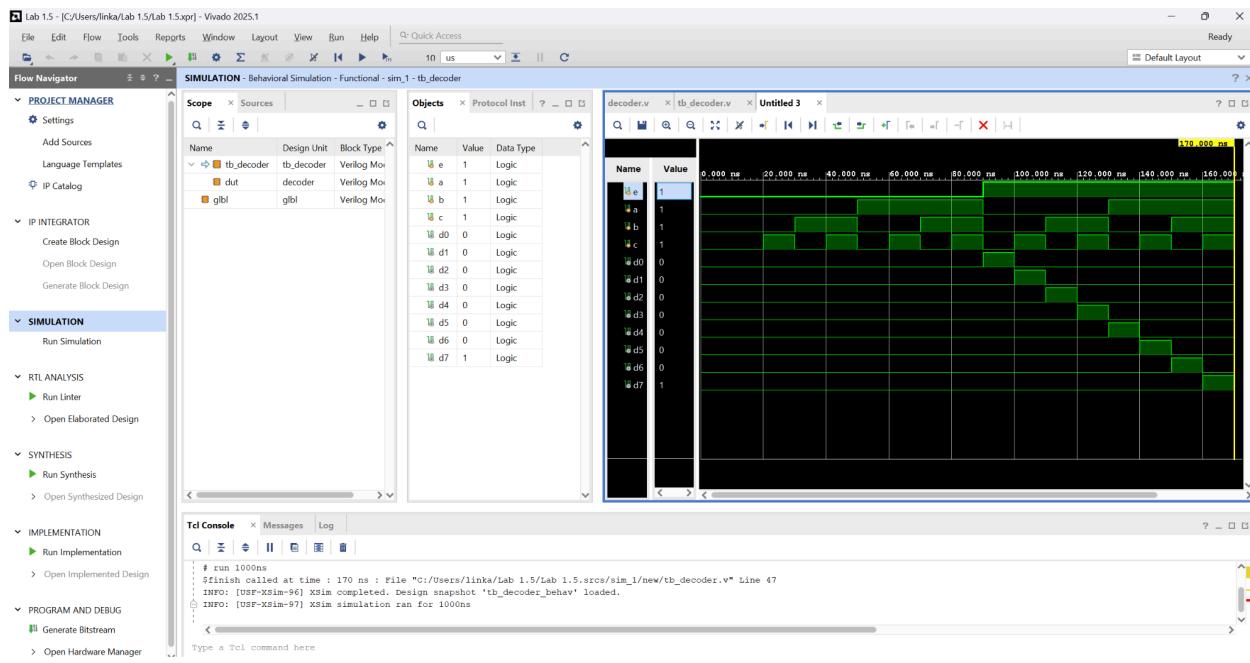


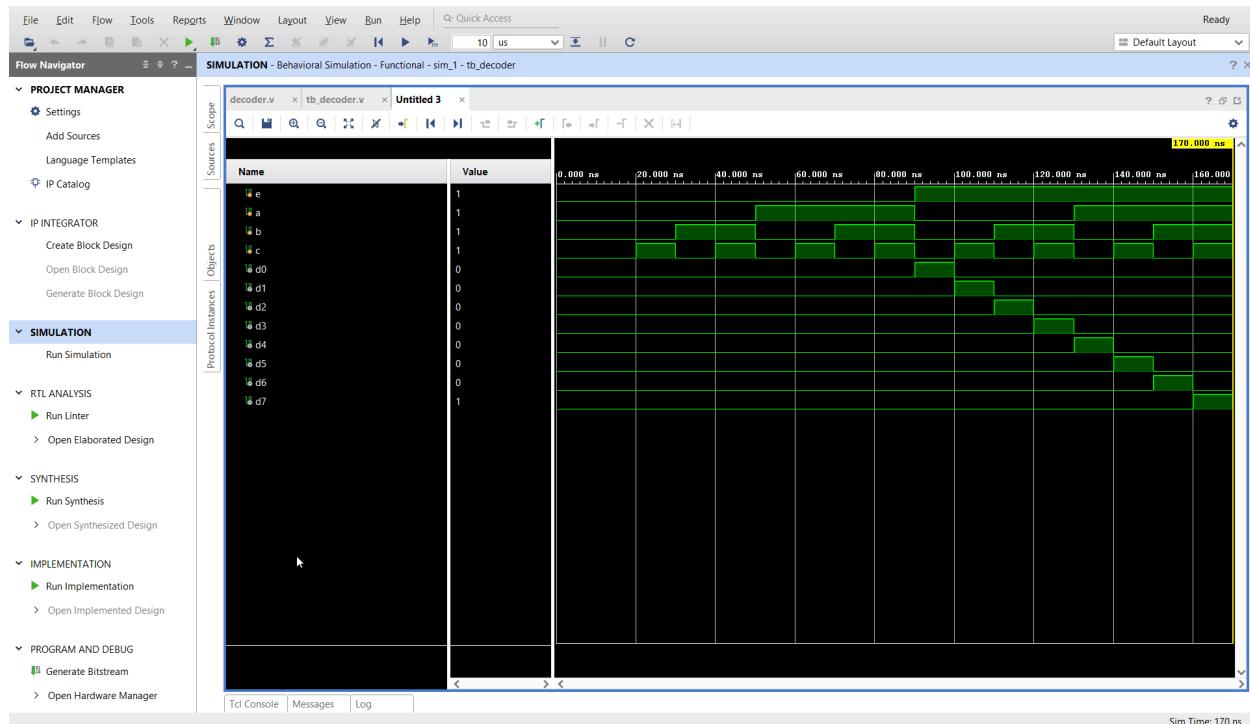
## Part 1

i)

### Simulation (Structural):



### Simulation (Behavioral):



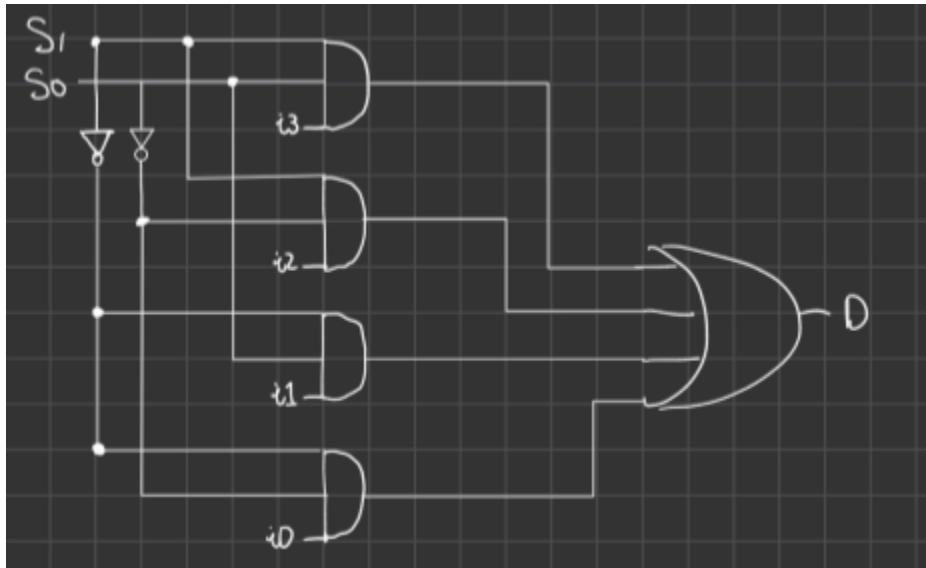
Part 2

ii. Truth table of the function iii. Algebraic expression of the logic function

$S_1$	$S_0$	d (output)
0	0	i0
0	1	i1
1	0	i2
1	1	i3

$d = S_1' S_0' \cdot i0 + S_1' S_0 \cdot i1 + S_1 S_0' \cdot i2 + S_1 S_0 \cdot i3$

#### iv. Logic circuit schematic for structural modelling



#### v. Verilog codes for module and testbench for structural modelling

The screenshot shows the Quartus II software interface. The left pane displays the Project Manager with various options like IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, IMPLEMENTATION, and PROGRAM AND DEBUG. The right pane shows the Verilog code for the multiplexer module:

```

11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 /////////////////////////////////////////////////
21
22 //////////////////////////////////////////////////////////////////
23 //module multiplexer(
24 //    input 10, i1, i2, i3, s0, s1,
25 //    output d
26 //);
27 //
28 //    wire nots0, nots1;
29 //    not int1(nots0, s0);
30 //    not int2(nots1, s1);
31 //    wire temp0, temp1, temp2, temp3;
32 //    and out0 (temp0, nots1, nots0, i0); // 00 = i0
33 //    and out1 (temp1, nots1, s0, i1); // 01 = i1
34 //    and out2 (temp2, s1, nots0, i2); // 10 = i2
35 //    and out3 (temp3, s1, s0, i3); // 11 = i3
36 //    or outf (d, temp0, temp1, temp2, temp3);
37 //
38 //endmodule
39
40 // Behavioral
41 module decoder(
42     input 10, i1, i2, i3, s0, s1,
43     output reg d // Declare outputs as reg
44 );
45
46 always @(*) begin
47

```

File Edit Flow Tools Reports Window Layout View Help Q: Quick Access Ready Default Layout

Flow Navigator PROJECT MANAGER - test

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

tb\_multiplexer.v multiplexer.v \* Sources Properties

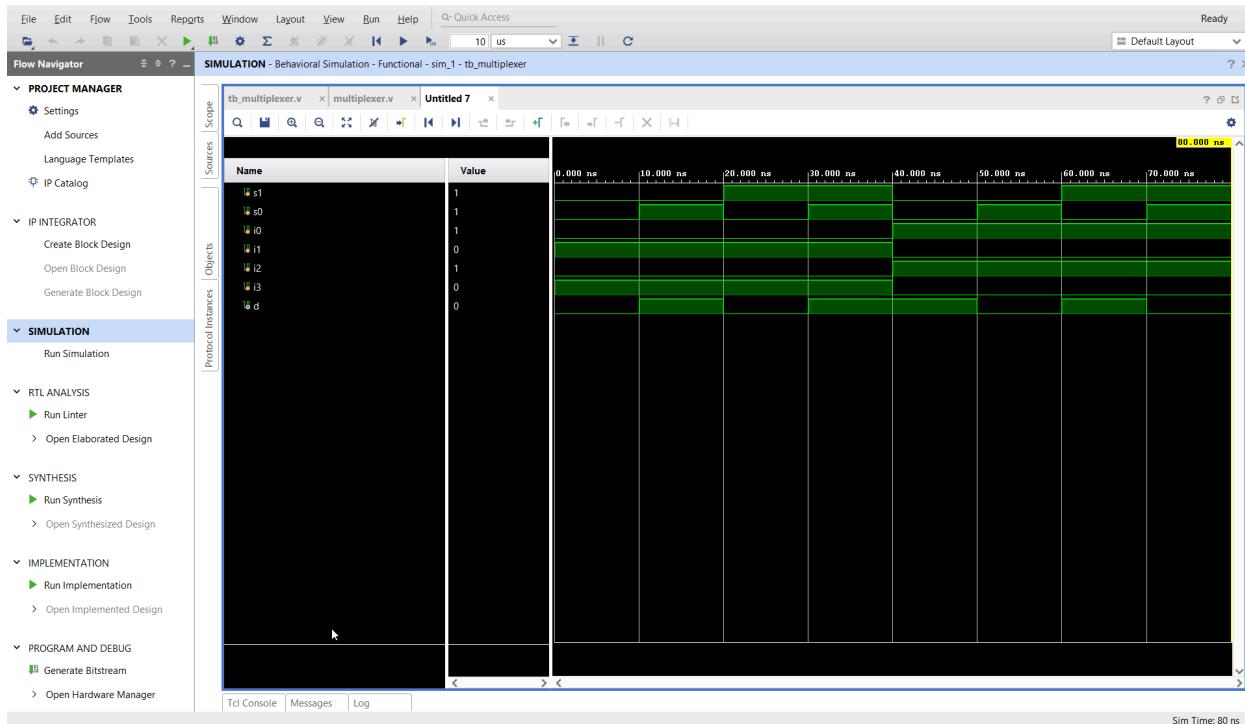
```

18 : // Additional Comments:
19 :
20 ///////////////////////////////////////////////////////////////////
21 :
22 :
23 module tb_multiplexer;
24 :
25   reg s1, s0, i0, i1, i2, i3;
26   wire d;
27 :
28 // dut: design under test
29   multiplexer dut (.s1(s1), .s0(s0), .i0(i0), .i1(i1), .i2(i2), .i3(i3), .d(d));
30 :
31 // test without checker
32 initial begin
33   s0 = 0; s1 = 0; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
34   #10
35   s0 = 1; s1 = 0; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
36   #10
37   s0 = 0; s1 = 1; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
38   #10
39   s0 = 1; s1 = 1; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
40   #10
41   s0 = 0; s1 = 0; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
42   #10
43   s0 = 1; s1 = 0; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
44   #10
45   s0 = 0; s1 = 1; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
46   #10
47   s0 = 1; s1 = 1; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
48   #10
49   $finish;
50 end
51
52
53 endmodule
54

```

Tcl Console Messages Log Reports Design Runs 28:30 Insert Verilog

## vi. Simulation waveform for structural modelling (Screenshot)



## vii. Verilog codes for module and testbench for behavioral modelling

### //// Behavioral

File Edit Flow Tools Reports Window Layout View Help Quick Access

Ready Default Layout

**PROJECT MANAGER - test**

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

Project Summary tb\_multiplexer.v multiplexer.v

```

26 //);
27
28 // wire nots0, nots1;
29 // not int1(nots0, s0);
30 // not int2(nots1, s1);
31 // wire temp0, temp1, temp2, temp3;
32 // and out0 (temp0, nots0, nots1, i0); // 00 = i0
33 // and out1 (temp1, nots1, s0, i1); // 01 = i1
34 // and out2 (temp2, s1, nots0, i2); // 10 = i2
35 // and out3 (temp3, s1, s0, i3); // 11 = i3
36 // or outf (d, temp0, temp1, temp2, temp3);
37
38 //endmodule
39
40 // Behavioral
41 module decoder(
42   input 10, i1, i2, i3, s0,
43   output reg d // Declare outputs as reg
44 );
45
46 always @(*) begin
47
48   // Default value for all outputs
49   d = 0;
50
51   // Activate outputs according to the truth table
52 case((s1, s0))
53   2'b00: d = i0;
54   2'b01: d = i1;
55   2'b10: d = i2;
56   2'b11: d = i3;
57 endcase
58 end
59
60 endmodule
61
62

```

Tcl Console Messages Log Reports Design Runs

40:1 Insert Verilog

File Edit Flow Tools Reports Window Layout View Help Quick Access

Ready Default Layout

**PROJECT MANAGER - test**

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

Project Summary tb\_multiplexer.v multiplexer.v

```

18 // Additional Comments:
19 //////////////////////////////////////////////////////////////////
20
21
22
23 module tb_multiplexer;
24
25   reg s1, s0, i0, i1, i2, i3;
26   wire d;
27
28 // dut: design under test
29 multiplexer dut (.s1(s1), .s0(s0), .i0(i0), .i1(i1), .i2(i2), .i3(i3), .d(d));
30
31 // test without checker
32 initial begin
33   s0 = 0; s1 = 0; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
34   $display("Initial State: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
35   s0 = 1; s1 = 0; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
36   $display("State 1: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
37   s0 = 0; s1 = 1; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
38   $display("State 2: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
39   s0 = 1; s1 = 1; i0 = 0; i1 = 1; i2 = 0; i3 = 1;
40   $display("State 3: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
41   s0 = 0; s1 = 0; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
42   $display("State 4: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
43   s0 = 1; s1 = 0; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
44   $display("State 5: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
45   s0 = 0; s1 = 1; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
46   $display("State 6: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
47   s0 = 1; s1 = 1; i0 = 1; i1 = 0; i2 = 1; i3 = 0;
48   $display("State 7: s0=%b, s1=%b, i0=%b, i1=%b, i2=%b, i3=%b", s0, s1, i0, i1, i2, i3);
49   $finish;
50
51 end
52
53 endmodule
54

```

Tcl Console Messages Log Reports Design Runs

2830 Insert Verilog

## viii. Simulation waveform for behavioral modelling (Screenshot)

