CS201P Assignment 1

Group 23

B20218 Narmit Kumar

B20217 Monika Meena

1st and 2nd Question

Half adder code:

full_adder.v | half_adder.v | ha_testbench.v | fa_testbench.v | multiplexer_4x1.v | mux_testbench.v | Unti

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...


module half_adder(
    input A,
    input B,
    output Sum,
    output Carry
    );

    assign Sum = A ^ B; //sum = A xor B
    assign Carry = A & B;  //carry = A and B
endmodule
```

Half adder testbench:

full_adder.v | half_adder.v | ha_testbench.v | fa_testbench.v | multiplexer_4x1.v | mux_testbench.v

```
module test_ha;
 reg A,B; //inputs
 wire S,C; //outputs


 //half_adder module under test
 half_adder uut(
  .A(A), .B(B), .Sum(S), .Carry(C)
  );

  initial begin

  //initializing values
A = 1'b0; B = 1'b0;

//After 10 ns
#10; A = 0; B = 1;

//After 10 ns
#10; A = 1; B = 0;

//After 10 ns
#10; A = 1; B = 1;



  end
  endmodule
```

Ful Adder code:

full_adder.v | half_adder.v | ha_testbench.v | fa_testbench.v | multiplexer_4x1.v | mux_testbench.v

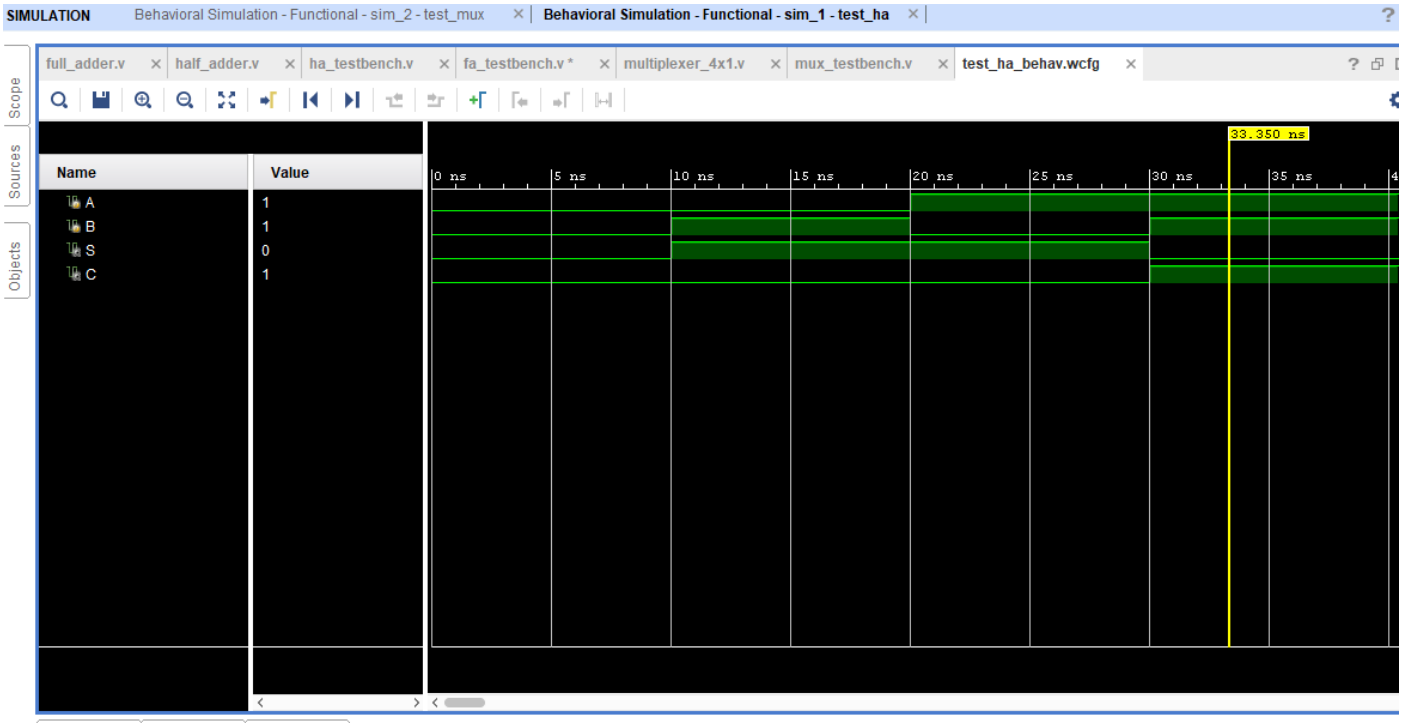
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...
module full_adder(
    input A,
    input B,
    input C,
    output Sum,
    output Carry
    );
    wire s1,s2;
    wire c1,c2;

    half_adder ha0(A,B,s1,c1);   //s1 = A xor B , c1 = A and B
    half_adder ha1(s1,C,Sum,c2);   //Sum = A xor B xor C,   c2 = (A xor B) & C
    or(Carry,c1,c2);    //Carry = (A and B ) or ( ( A xor B ) & C)



endmodule
```

Ful Adder testbench:

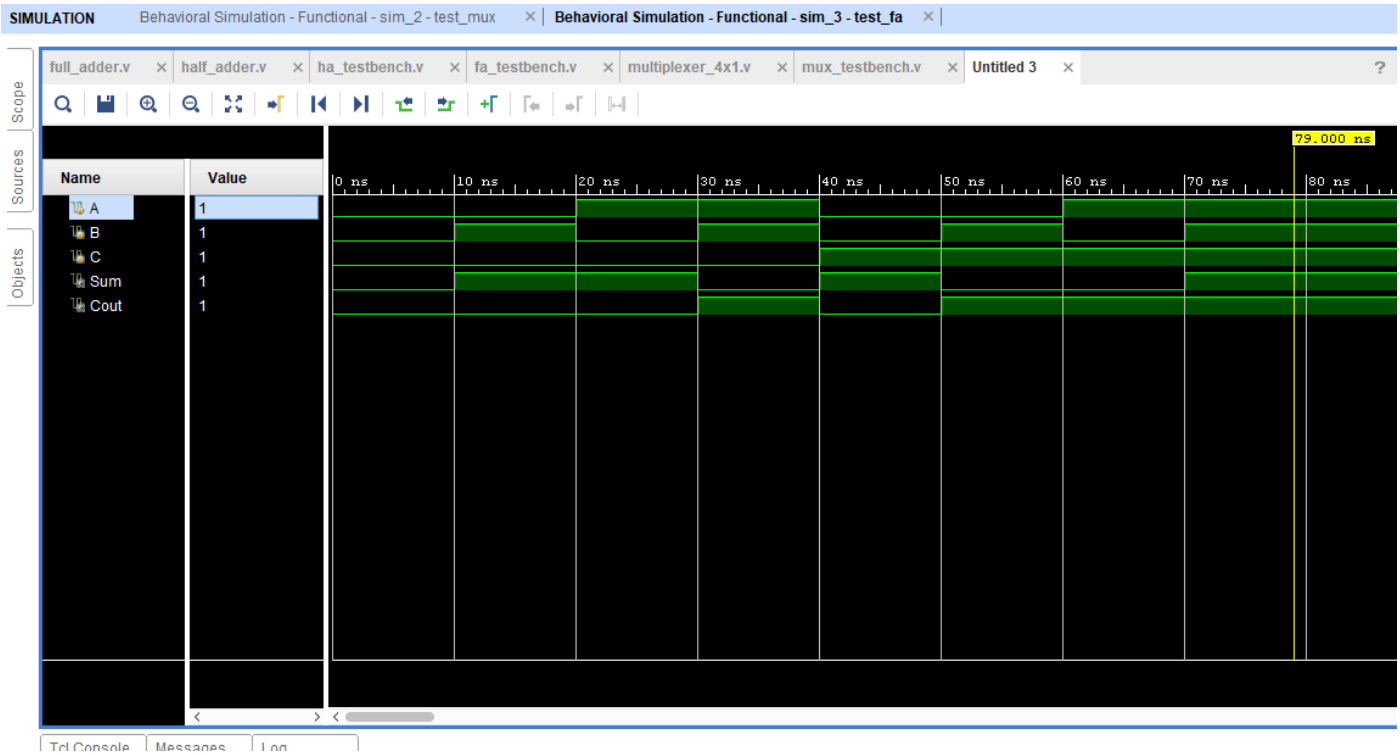
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...

module test_fa;
 reg A,B,C;  //inputs
 wire Sum,Cout;  //outputs


 //full_adder under test
 full_adder uut(
  .A(A), .B(B),.C(C), .Sum(Sum), .Carry(Cout)
  );

  initial begin
  C = 1'b0; A = 1'b0; B = 1'b0;  //initializing values to inputs
#10; C = 0; A = 0; B = 1;  //after 10 seconds we change our input values
#10; C = 0; A = 1; B = 0;
#10; C = 0; A = 1; B = 1;
#10; C = 1; A = 0; B = 0;
#10; C = 1; A = 0; B = 1;
#10; C = 1; A = 1; B = 0;
#10; C = 1; A = 1; B = 1;


  end
  endmodule
```

Outputs:

Half Adder –

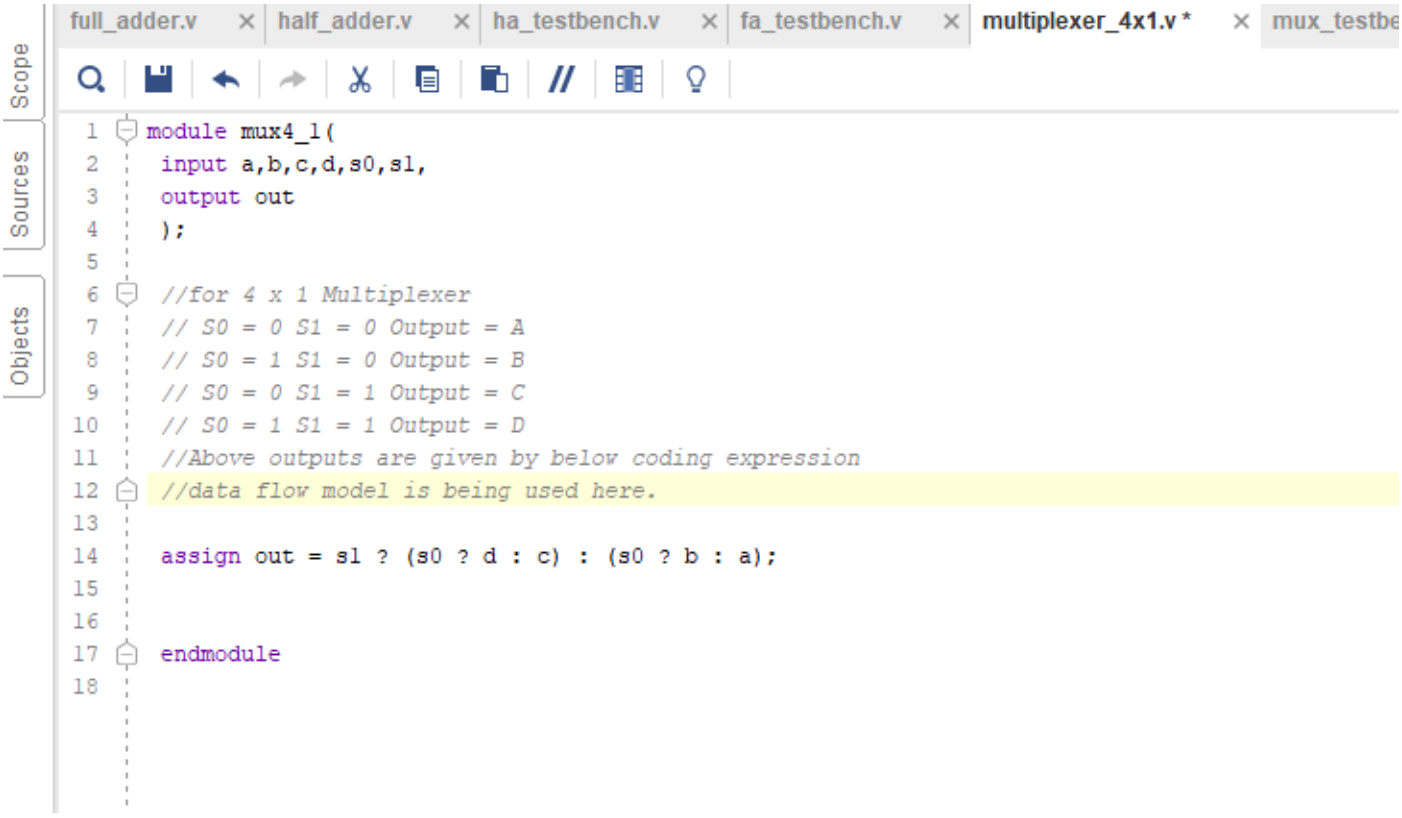


Ful Adder-



3rd and 4th Question

Multiplexer 4 to 1 code:

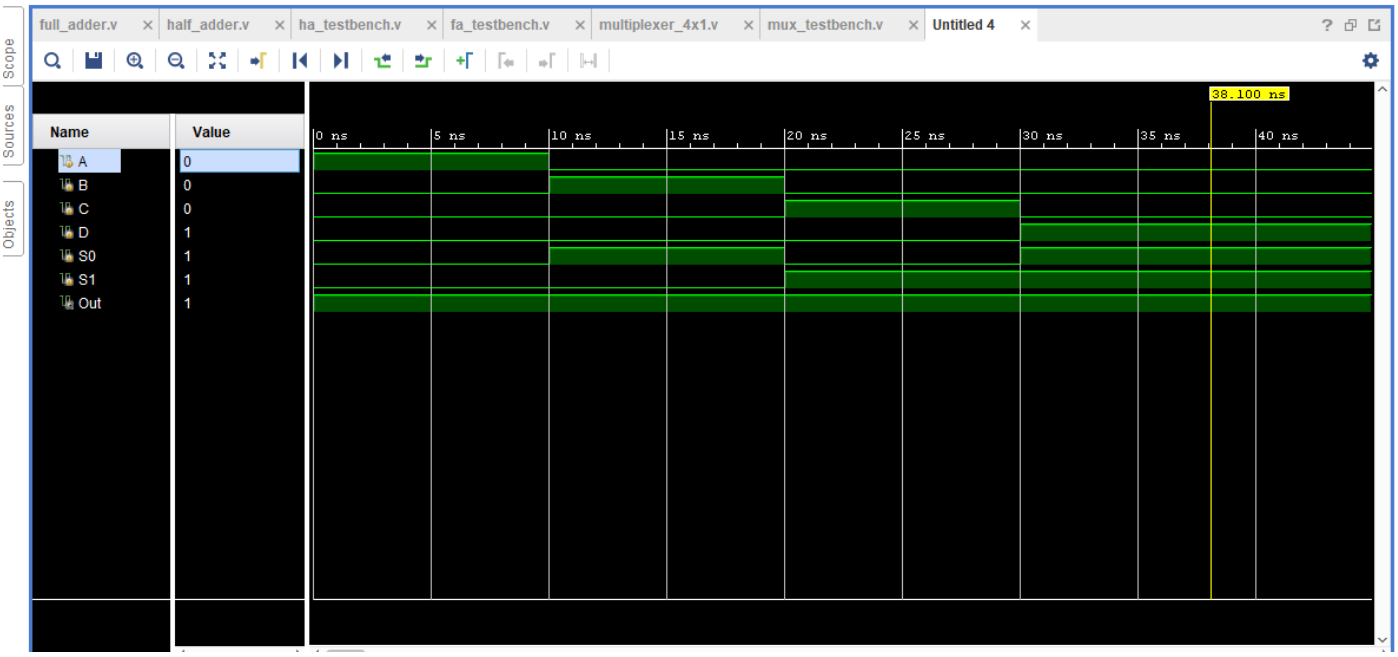



```
module mux4_1(
  input a,b,c,d,s0,s1,
  output out
  );

  //for 4 x 1 Multiplexer
  // S0 = 0 S1 = 0 Output = A
  // S0 = 1 S1 = 0 Output = B
  // S0 = 0 S1 = 1 Output = C
  // S0 = 1 S1 = 1 Output = D
  //Above outputs are given by below coding expression
  //data flow model is being used here.

  assign out = s1 ? (s0 ? d : c) : (s0 ? b : a);


  endmodule
```

Testbench:

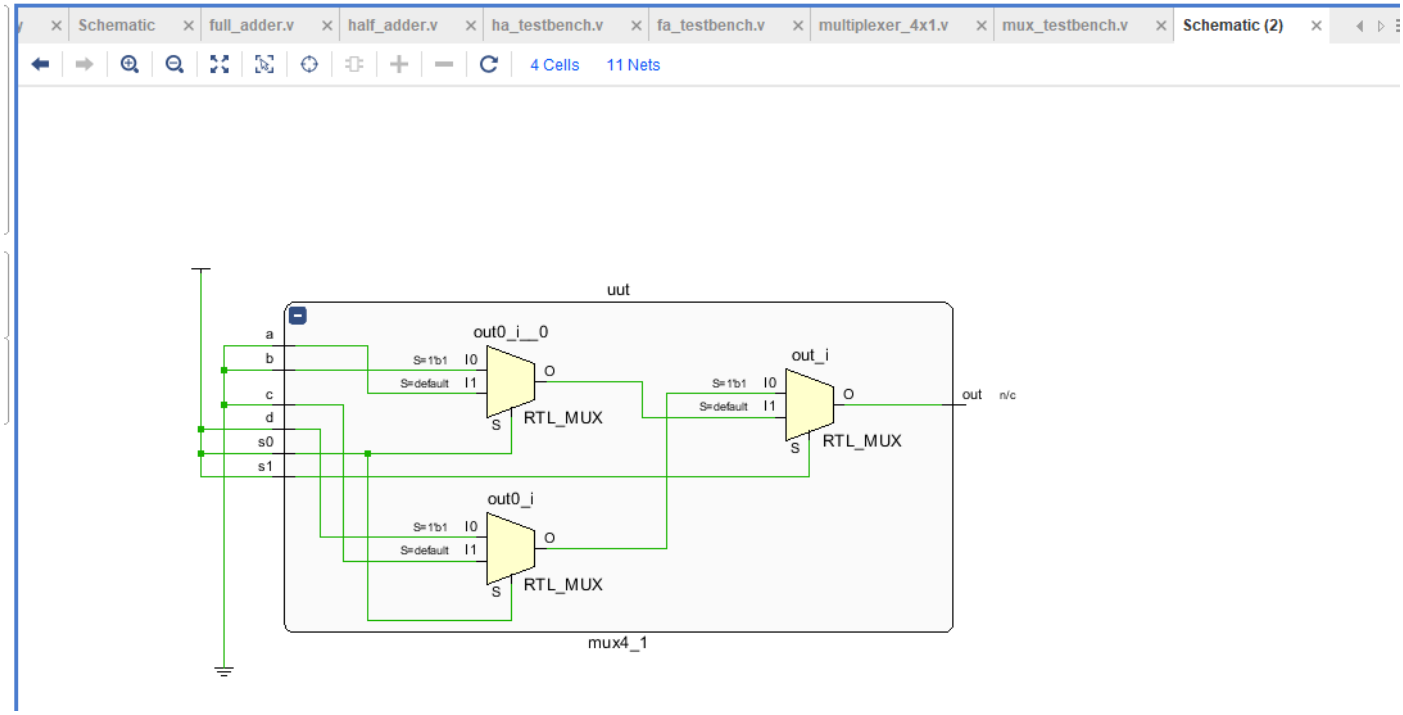
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...

module test_mux;
 reg A,B,C,D,S0,S1;  //inputs
 wire Out;    //outputs


 //multiplexer under test
 mux4_1 uut(
  .a(A), .b(B),.c(C), .d(D), .s0(S0), .s1(S1), .out(Out)
  );

  initial begin
  A = 1'b1; B = 1'b0;  C = 1'b0; D = 1'b0;  S0 = 1'b0; S1 = 1'b0;  //initializing values
  #10 S0 = 1; S1 = 0;A = 0; B = 1; C = 0; D = 0;  //after 10 ns input is updated
  #10 S0 = 0; S1 = 1;A = 0; B = 0; C = 1; D = 0;
  #10 S0 = 1; S1 = 1; A = 0; B = 0; C = 0; D = 1;

  end
  endmodule
```

Output:

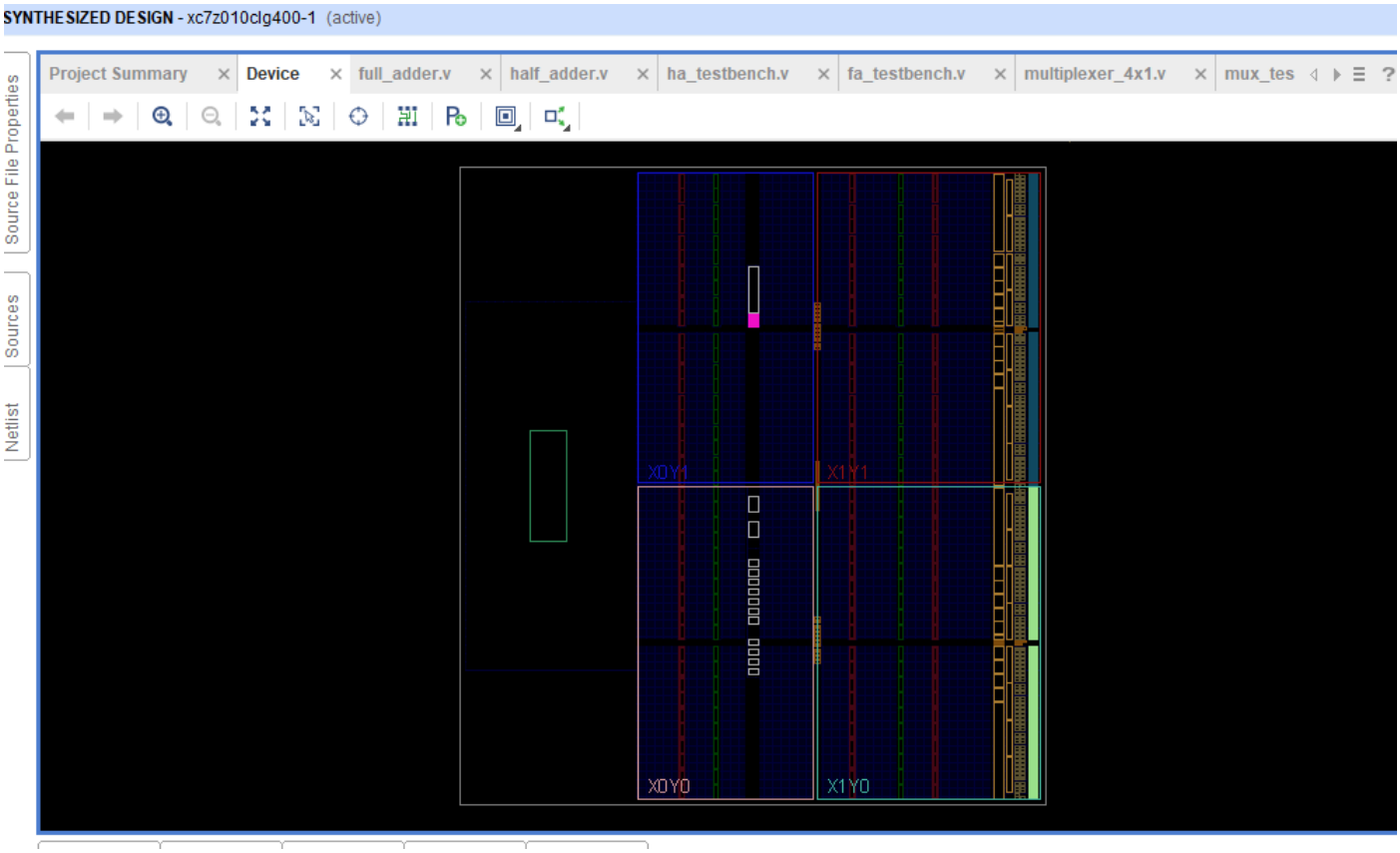


5th Question

RTL Schematic of MUX



Synthesized Design


SYNTHESIZED DESIGN - xc7z010clg400-1 (active)
Source File Properties
Sources
Netlist
Project Summary
Device
full_adder.v
half_adder.v
ha_testbench.v
fa_testbench.v
multiplexer_4x1.v
mux_tes
X0Y1
X1Y1
X0Y0
X1Y0