Erollment Number : 20114079

Rahul Kurkure

CSE O3

Problem1 : 1.1

1.2

ModelSim - INTEL FPGA STARTER EDITION 2020.1
Transcript
Reading pref.tcl
project open C:/intelFPGA/20.1/ECN104Assignment2
Loading project ECN104Assignment2
ModelSim> vsim -gui work.tb_1_4
vsim -gui work.tb_1_4
Start time: 17:24:55 on Jul 04,2021
Loading work.tb_1_4
Loading work.problem_1_4
add wave -position end sim:/tb_1_4/*
VSIM 4> run -all
CLOCK WORKS PROPERLY
** Note: $finish : C:/intelFPGA/20.1/tb_1.v(133)
Time: 1 us Iteration: 0 Instance: /tb_1_4
1
Break in Module tb_1_4 at C:/intelFPGA/20.1/tb_1.v line 133
VSIM 5>

solution_1.v - 20.1 - Visual Studio Code
solution_2.v
solution_1.v
tb_2.v
tb_3.v
solution_3.v
tb_1.v
// non-inverting buffer
module problem_1_2(result , input1);
input input1;
output result ;
wire temp0 ;
problem_1_1 g1(temp0 , input1);
problem_1_1 g2(result , temp0);
endmodule

1.4

solution_1.v - 20.1 - Visual Studio Code
solution_2.v
solution_1.v
tb_2.v
tb_3.v
solution_3.v
tb_1.v
problem_1_1 g1(temp0 , input1);
problem_1_1 g2(result , temp0);
endmodule
// ring oscillator
`timescale 1ns/1ps
module problem_1_4(result , input1);
input input1 ;
output reg result ;
reg rg1, rg2, rg3 , rg4 ;
initial begin
result = input1 ;
end
always @ (result) begin
#5 rg1 <= ~ result ;
#5 rg2 <= ~ rg1;
#5 rg3 <= ~ rg2;
#5 rg4 <= ~ rg3;
#5 result <= ~ rg4;
end
endmodule

ModelSim - INTEL FPGA STARTER EDITION 2020.1
Transcript
Load canceled
Compile of solution_1.v was successful.
Compile of solution_2.v was successful.
Compile of tb_1.v was successful.
3 compiles, 0 failed with no errors.
ModelSim> vsim -gui work.tb_1_2
vsim -gui work.tb_1_2
Start time: 17:12:40 on Jul 04,2021
Loading work.tb_1_2
Loading work.problem_1_2
Loading work.problem_1_1
add wave -position insertpoint sim:/tb_1_2/*
VSIM 4> run -all
Test case passed for : Input : 0 Output : 0
ALL TEST CASES PASSED
** Note: $finish : C:/intelFPGA/20.1/tb_1.v(86)
Time: 25 ns Iteration: 0 Instance: /tb_1_2
1
Break in Module tb_1_2 at C:/intelFPGA/20.1/tb_1.v line 86
VSIM 5>

TESTBENCHES

Here we are first going to make a nand gate by calculating the and of A and B using & operator ,then inverting the result will give us NAND of A and B . NAND : ~ (A & B) According to question we have to use NAND gates to make AND gates so first we will take nand of both inputs then we will take another NAND to invert our result . Step 1 : A NAND B : (AB)’ → A’ + B’ Step 2 : (AB)’ NAND (AB)’ Result : ‘((AB)’ (AB)’)’ → AB

Problem 2 :

2.1

ModelSim - INTEL FPGA STARTER EDITION 2020.1
Transcript
Loading work.tb_2_1
Loading work.solution_2_1
add wave -position end sim:/tb_2_1/*
VSIM 7> run -all
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 1 Result : z
TEST CASES FAILED
** Note: $finish : C:/intelFPGA/20.1/tb_2.v(85)
Time: 100 ns Iteration: 0 Instance: /tb_2_1
1
Break in Module tb_2_1 at C:/intelFPGA/20.1/tb_2.v line 85
VSIM 8>

solution_2.v - 20.1 - Visual Studio Code
solution_2.v
solution_1.v
tb_2.v
tb_3.v
solution_3.v
tb_1.v
module solution_2_1(in1 , in2 , select , result);
input in1 , in2 , select ;
output result ;
assign result = select ? in2 : in1 ;
endmodule
`timescale 1ns/1ps
module multiplexer(in1 , in2 , select , result);
input in1 , in2 , select ;
output result ;
assign #2 result = select ? in2 : in1 ;
endmodule

ModelSim - INTEL FPGA STARTER EDITION 2020.1
Transcript
ModelSim> vsim -gui work.tb_2_2
vsim -gui work.tb_2_2
Start time: 18:34:36 on Jul 04,2021
Loading work.tb_2_2
Loading work.solution_2_2
add wave -position end sim:/tb_2_2/*
VSIM 4> run -all
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 0 Result : z
Test Case Failed with : Select : 1 Result : z
Test Case Failed with : Select : 1 Result : z
TEST CASES FAILED
** Note: $finish : C:/intelFPGA/20.1/tb_2.v(179)
Time: 100 ns Iteration: 0 Instance: /tb_2_2
Type here to search
ENG
18:35
04-07-2021

File Edit Selection View Go Run Terminal Help • tb_2.v - 20.1 - Visual Studio Code

solution_2.v | solution_1.v | tb_2.v | tb_3.v | solution_3.v | tb_1.v

tb_2.v

```
    end


    end


    end


    initial begin
        FailCount = 0 ;
        select    = 0 ;
        in1     = 0 ;
        in2     = 0 ;

#100

if(FailCount != 0 )
begin
    $display( "   TEST CASES FAILED   " );
end

    else begin
        $display ("   ALL TEST CASES PASSED    ");
    end

    $finish ;

    end


    always #50  in1 = ~in1 ;
    always #25  in2 = ~in2 ;
    always #12  select = ~select ;



endmodule
```

Ln 6, Col 13 Spaces: 4 UTF-8 CRLF Verilog Go Live

Type here to search ENG 23:45 04-07-2021

File Edit Selection View Go Run Terminal Help • tb_2.v - 20.1 - Visual Studio Code

solution_2.v | solution_1.v | tb_2.v | tb_3.v | solution_3.v | tb_1.v

tb_2.v

```
    forever begin #10

        if( ( (select == 0) && result == in1 ) )begin
            $display(" ALL OK ","Select : %b ",select ,
            "Input : %b ",in1,
            "Result : %b ",result);
    end

        else if( ( (select == 1) && result == in2 ) )begin
            $display("ALL OK","Select : %b ",select ,
            "Input : %b ",in2,
            "Result : %b ",result);
    end


    else
    begin
        FailCount = FailCount + 1 ;
        $display("Test Case Failed with : ",
        "Select : %b ",select,
        "Result : %b ", result );
    end


    end


    end


    initial begin
        FailCount = 0 ;
        select    = 0 ;
        in1     = 0 ;
        in2     = 0 ;

#100

if(FailCount != 0 )
```

Ln 6, Col 13 Spaces: 4 UTF-8 CRLF Verilog Go Live

Type here to search ENG 23:45 04-07-2021

File Edit Selection View Go Run Terminal Help • tb_2.v - 20.1 - Visual Studio Code

solution_2.v | solution_1.v | tb_2.v | tb_3.v | solution_3.v | tb_1.v

tb_2.v

```verilog
`timescale 1ns/1ps

module tb_2_1;

reg in1 , in2 ;
reg select ;

wire result ;
integer  FailCount;

    solution_2_1 S1(select ,result ,input2 ,input1);


    initial begin #5

        if( ( (select == 0) && result == in1 ) )begin
            $display(" ALL OK ","Select : %b ",select ,
            "Input : %b ",in1,
            "Result : %b ",result);
    end

        else if( ( (select == 1) && result == in2 ) )begin
            $display("ALL OK","Select : %b ",select ,
            "Input : %b ",in2,
            "Result : %b ",result);
    end


    else
    begin
        FailCount = FailCount + 1 ;
        $display("Test Case Failed with : ",
        "Select : %b ",select ,
        "Result : %b ", result );
    end


    forever begin #10
```

Live Share | Ln 6, Col 13 | Spaces: 4 | UTF-8 | CRLF | Verilog | Go Live

Type here to search | ENG | 23:45 04-07-2021

File Edit Selection View Go Run Terminal Help • tb_2.v - 20.1 - Visual Studio Code

solution_2.v | solution_1.v | tb_2.v | tb_3.v | solution_3.v | tb_1.v

tb_2.v

```verilog
        if( ( (select == 0) && result == in1 ) )begin
            $display(" ALL OK ","Select : %b ",select ,
            "Input : %b ",in1,
            "Result : %b ",result);
    end

        else if( ( (select == 1) && result == in2 ) )begin
            $display("ALL OK","Select : %b ",select ,
            "Input : %b ",in2,
            "Result : %b ",result);
    end


    else
    begin
        FailCount = FailCount + 1 ;
        $display("Test Case Failed with : ",
        "Select : %b ",select,
        "Result : %b ", result );
    end


    end


    end


    initial begin
        FailCount = 0 ;
        select    = 0 ;
        in1       = 0 ;
        in2       = 0 ;

#100

if(FailCount != 0 )
begin
$display( "   TEST CASES FAILED    " );
end

    else begin
        $display ("   ALL TEST CASES PASSED     ");
    end

    $finish ;

    end


    always #50  in1 = ~in1 ;
    always #25  in2 = ~in2 ;
    always #12  select = ~select ;



endmodule
```

Live Share | Ln 6, Col 13 | Spaces: 4 | UTF-8 | CRLF | Verilog | Go Live

Type here to search | ENG | 23:46 04-07-2021

File Edit Selection View Go Run Terminal Help
• tb_2.v - 20.1 - Visual Studio Code
solution_2.v
solution_1.v
tb_2.v
tb_3.v
solution_3.v
tb_1.v
tb_2.v
always #12 select = ~select ;
endmodule
module multiplexer;
reg in1 , in2 ;
reg select ;
wire result ;
integer FailCount;
multiplexer S2(select ,result ,input2 ,input1);
initial begin #5
if(((select == 0) && result == in1))begin
$display(" ALL OK ","Select : %b ",select ,
"Input : %b ",in1,
"Result : %b ",result);
end
else if(((select == 1) && result == in2))begin
$display("ALL OK","Select : %b ",select ,
"Input : %b ",in2,
"Result : %b ",result);
end
else
begin
FailCount = FailCount + 1 ;
$display("Test Case Failed with : ",
"Select : %b ",select ,
"Result : %b ", result);
end
forever begin #10
if(((select == 0) && result == in1))begin
$display(" ALL OK ","Select : %b ",select ,
"Input : %b ",in1,
"Result : %b ",result);
end
else if(((select == 1) && result == in2))begin
$display("ALL OK","Select : %b ",select ,
"Input : %b ",in2,
"Result : %b ",result);
end
else
begin
FailCount = FailCount + 1 ;
$display("Test Case Failed with : ",
"Select : %b ",select,
"Result : %b ", result);
end
Live Share
Ln 6, Col 13
Spaces: 4
UTF-8
CRLF
Verilog
Go Live
Type here to search
ENG
23:46
04-07-2021

PROBLEM 3

solution_3.v - 20.1 - Visual Studio Code

solution_2.v | solution_1.v | tb_2.v | tb_3.v | solution_3.v | tb_1.v

solution_3.v

```
/* Enroll Number : 20114079 ;
 number 1 (Largest Digit) : 9 ;
 number 2 Largest + Least-Significant Digit   : 9+9 = 18 ;

    9 +18 = 27 ;
*/

module turing(clk , result ,done );

input clk ;
output reg done ;
output reg [31:0] result ;

`define SYMB_A     2'b00
`define SYMB_ADD   2'b01
`define SYMB_BLANK 2'b10

reg [1:0] tape [29:0];
reg back ;
integer n1 , n2 ;

initial begin

    result = 32'b0 ;
    back = 0 ;
    done =  0 ;

    tape[0] = `SYMB_BLANK ;
```

Ln 70, Col 8 | Spaces: 4 | UTF-8 | CRLF | Verilog | Go Live

solution_2.v | solution_1.v | tb_2.v | tb_3.v | solution_3.v | tb_1.v

solution_3.v

```
    #10
    $finish ;

    end

    if(tape[head] == `SYMB_ADD)
    begin
        tape[head] = `SYMB_A ;
        tape[28]   = `SYMB_BLANK ;
    end

if (tape[head == `SYMB_BLANK]) begin

    back = 1 ;
    head = head - 1 ;

end



end



endmodule
```

Ln 70, Col 8 | Spaces: 4 | UTF-8 | CRLF | Verilog | Go Live

solution_3.v - 20.1 - Visual Studio Code

solution_3.v

```
integer head = 1 ;

always @(posedge clk ) begin

    if (head !=0 && tape[head ] == `SYMB_A)
    begin
        if(back == 0 )
        begin
            head = head +1 ;
        end

        else begin
            head = head + 1 ;
            result = result + 1;
        end

    end

    if(head <= 0 )
    begin
        done = 1 ;

    #10
    $finish ;

    end

    if(tape[head] == `SYMB_ADD)
    begin
```

solution_3.v - 20.1 - Visual Studio Code

solution_3.v

```
    done =  0 ;

    tape[0] = `SYMB_BLANK ;


    for(n1 = 1 ; n1 <=9 ; n1 = n1 +  1)begin
        tape[n1] = `SYMB_A ;
    end

    tape[10] = `SYMB_ADD ;

    for(n2 = 11 ; n2 <= 28 ; n1 = n1 +  1)begin
        tape[n1] = `SYMB_A ;
    end

    tape[29] = `SYMB_BLANK ;


end


integer head = 1 ;

always @(posedge clk ) begin

    if (head !=0 && tape[head ] == `SYMB_A)
    begin
        if(back == 0 )
        begin
            head = head +1 ;
```

Problem 4 :

Explaination : Adding zeroes to the binary numbers to wont make any difference so we will completely ignore the multiplication by zero , for first one we will shift the bits to the left by 1 and for next time multiplication by 1 we will shift the bits by 3 units to the left and then finnaly add those two numbers.

solution_4.v - 20.1 - Visual Studio Code

solution_4.v

```verilog
`timescale 1ns/1ps

module equation_lut(a, b, c ,d ,result );

input a ,b , c ,d  ;
output result ;

reg LUT[1:0][1:0][1:0][1:0];
reg [1:0] address [1:0];

initial
begin


    LUT[0][0][0][0] = 1 ;
    LUT[0][0][0][1] = 0 ;
    LUT[0][0][1][0] = 0 ;
    LUT[0][0][1][1] = 1 ;
    LUT[0][1][0][0] = 1 ;
    LUT[0][1][0][1] = 0 ;
    LUT[0][1][1][0] = 0 ;
    LUT[0][1][1][1] = 1 ;
    LUT[1][0][0][0] = 0 ;
    LUT[1][0][0][1] = 1 ;
    LUT[1][0][1][0] = 1 ;
    LUT[1][0][1][1] = 0 ;
    LUT[1][1][0][0] = 0 ;
    LUT[1][1][0][1] = 1 ;
    LUT[1][1][1][0] = 1 ;
    LUT[1][1][1][1] = 1 ;


    address[0][0] = a ;
    address[0][1] = b ;
    address[1][0] = c ;
    address[1][1] = d ;

end


assign result = LUT[address[0][0]][address[0][1]][address[1][0]][address[1][1]];

endmodule
```

solution_4.v - 20.1 - Visual Studio Code

solution_4.v

```verilog
module and4_lut(address , result );

output result ;
input [3:0] address ;

reg[1:0] rg1 ;
reg[1:0] rg2 ;
reg[1:0] rg3 ;

initial
begin

rg1[0] = address [0];
rg1[1] = address [1];
rg2[0] = address [2];
rg2[1] = address [3];

end


and2_lut L1( rg1 , rg3[0]);
and2_lut L2( rg2 , rg3[1]);
and2_lut L3( rg3 , result);

endmodule

`timescale 1ns/1ps
```

solution_4.v - 20.1 - Visual Studio Code

solution_4.v

```verilog
module and2_lut(address ,result );

    input [1:0] address ;
    reg [3:0] LUT[1:0] ;
    output result ;

    initial begin
        LUT[0][0] = 0 ;
        LUT[0][1] = 0 ;
        LUT[1][0] = 0 ;
        LUT[1][1] = 1 ;
    end

    assign result = LUT[address[0]][address[1]];


endmodule


module and4_lut(address , result );

output result ;
input [3:0] address ;

reg[1:0] rg1 ;
reg[1:0] rg2 ;
reg[1:0] rg3 ;

initial
```

ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
sim - Default
Objects
Wave - Default
Transcript
Reading pref.tcl
project open C:/intelFPGA/20.1/ECN104Assignment2
Loading project ECN104Assignment2
Compile of tb_3.v was successful.
ModelSim> vsim -gui work.tb_3
vsim -gui work.tb_3
Start time: 21:06:57 on Jul 04,2021
Loading work.tb_3
Loading work.solution_3
add wave -position end sim:/tb_3/*
VSIM 4> run -all
Type here to search
ENG
21:09
04-07-2021

ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
sim - Default
Objects
Wave - Default
Transcript
Reading pref.tcl
project open C:/intelFPGA/20.1/ECN104Assignment2
Loading project ECN104Assignment2
Compile of tb_3.v was successful.
ModelSim> vsim -gui work.tb_3
vsim -gui work.tb_3
Start time: 21:06:57 on Jul 04,2021
Loading work.tb_3
Loading work.solution_3
add wave -position end sim:/tb_3/*
VSIM 4> run -all
Type here to search
ENG
21:09
04-07-2021

ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type Top Category Visibility Total coverage
tb_3 tb_3 Module DU Instance +acc=...
turing solution_3 Module DU Instance +acc=...
#INITIAL#8 tb_3 Process - +acc=...
#ALWAYS#12 tb_3 Process - +acc=...
#ALWAYS#18 tb_3 Process - +acc=...
#vsim_capacity# Capacity Statistics +acc=...
Wave - Default
/tb_3/clk
/tb_3/done
/tb_3/result
Now 0 ps
Cursor 1 0 ps
Transcript
Reading pref.tcl
project open C:/intelFPGA/20.1/ECN104Assignment2
Loading project ECN104Assignment2
Compile of solution_3.v was successful.
Compile of tb_3.v was successful.
ModelSim> vsim -gui work.tb_3
vsim -gui work.tb_3
Start time: 21:17:21 on Jul 04,2021
Loading work.tb_3
Loading work.solution_3
add wave -position end sim:/tb_3/*
VSIM 4> run -all
Project : ECN104Assignment2 Now: 0 ps Delta: 0 sim:/tb_3
Type here to search
ENG 21:20 04-07-2021