Laboratory Report #_4_

Name: Christian Jay Gallardo **Date Completed:** 9/19/2025

Laboratory Exercise Title: Dataflow Modeling of Combinational Circuits

Target Course Outcomes:

CO1: Create descriptions of digital hardware components such as in combinational and sequential circuits using synthesizable Verilog HDL constructs.

CO2: Verify the functionality of HDL-based components through design verification tools.

Exercise 4A:

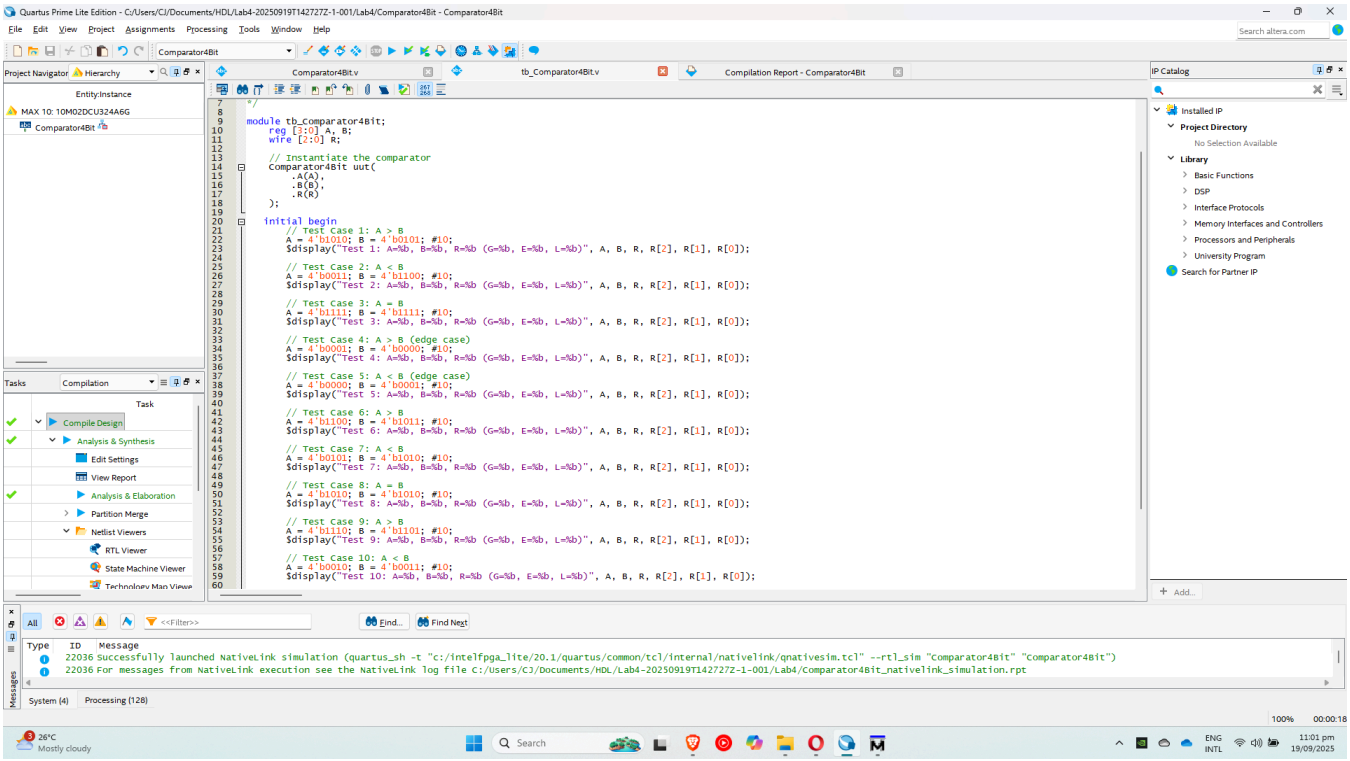
```
/*
Filename        : Comparator4Bit.v
Author          : Christian Jay Gallardo
Class           : CPE3101L
GrP. Schedule   : GRP. 4 FRIDAY 10:30AM - 1:30PM
Description     :
*/


module Comparator4Bit(
    input [3:0] A,
    input [3:0] B,
    output [2:0] R
);

    // Dataflow modeling
    assign R[2] = (A > B) ? 1'b1 : 1'b0;  // G: A > B
    assign R[1] = (A == B) ? 1'b1 : 1'b0; // E: A == B
    assign R[0] = (A < B) ? 1'b1 : 1'b0;  // L: A < B

endmodule
```

Figure 1 Verilog HDL Code for the 4-bit Comparator




```
*/
module tb_comparator4Bit;
    reg [3:0] A, B;
    wire [2:0] R;

    // Instantiate the comparator
    Comparator4Bit uut(
        .A(A),
        .B(B),
        .R(R)
    );

    initial begin
        // Test Case 1: A > B
        A = 4'b1010; B = 4'b0101; #10;
        $display("Test 1: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 2: A < B
        A = 4'b0011; B = 4'b1100; #10;
        $display("Test 2: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 3: A = B
        A = 4'b1111; B = 4'b1111; #10;
        $display("Test 3: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 4: A > B (edge case)
        A = 4'b0001; B = 4'b0000; #10;
        $display("Test 4: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 5: A < B (edge case)
        A = 4'b0000; B = 4'b0001; #10;
        $display("Test 5: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 6: A > B
        A = 4'b1100; B = 4'b1011; #10;
        $display("Test 6: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 7: A < B
        A = 4'b0101; B = 4'b1010; #10;
        $display("Test 7: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 8: A = B
        A = 4'b1010; B = 4'b1010; #10;
        $display("Test 8: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 9: A > B
        A = 4'b1110; B = 4'b1101; #10;
        $display("Test 9: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);

        // Test Case 10: A < B
        A = 4'b0010; B = 4'b0011; #10;
        $display("Test 10: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
```

Figure 2 Verilog HDL testbench code for the 4-bit Comparator

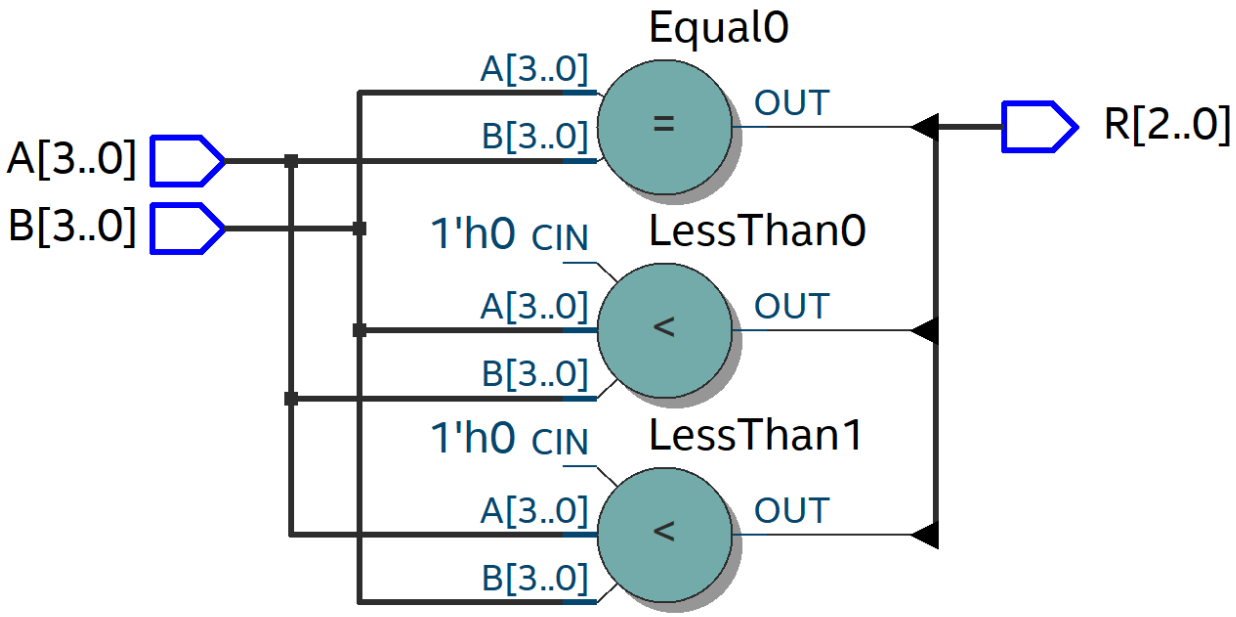


Figure 3 RTL Viewer of the 4-bit Comparator

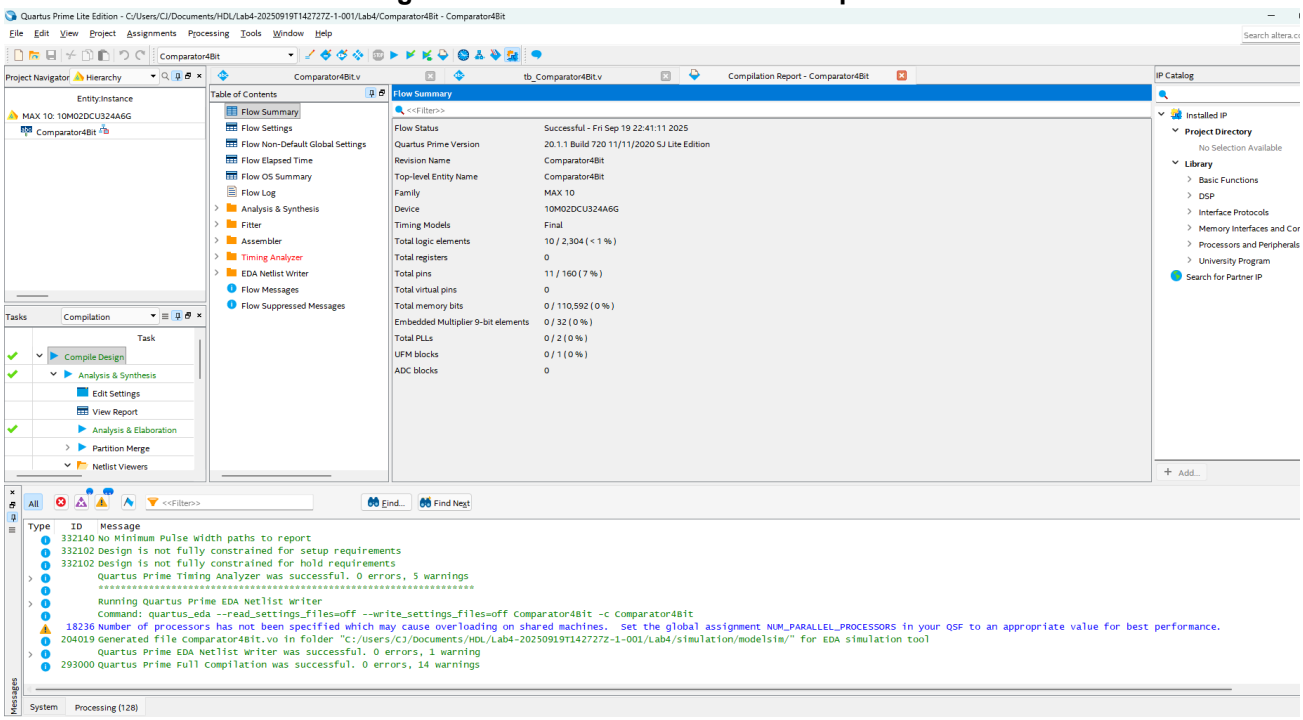


Figure 4 Summary of Design Synthesis for the 4-bit comparator

Figure 5 Successful simulation results with 10 test cases with annotations

Exercise 4B: Part 1

Table 1 Truth Table

S (Selector)	A (Input)	B (Input)	C (Input)	D (Input)	Y (Output)
00	0001	0010	0100	1000	0001
01	0001	0010	0100	1000	0010
10	0001	0010	0100	1000	0100
11	0001	0010	0100	1000	1000

```
/*
Filename        : The Mux_4x1_nbit.v
Author          : Christian Jay Gallardo
Class           : CPE3101L
Grp. Schedule   : GRP. 4 FRIDAY 10:30AM - 1:30PM
Description     : The Mux_4x1_nbit is a parameterized 4-to-1 multiplexer that selects one of four
                  n-bit inputs based on a 2-bit selector and outputs the selected value.
*/


module Mux_4x1_nbit #(parameter n = 4) (
    input [n-1:0] A, B, C, D,  // 4 input buses
    input [1:0] S,             // Selector input (2 bits)
    output [n-1:0] Y           // Output bus
);

    // Simple case statement to select the output based on S
    assign Y = (S == 2'b00) ? A :
               (S == 2'b01) ? B :
               (S == 2'b10) ? C :
               (S == 2'b11) ? D : A;  // Default case (for safety)

endmodule
```

Figure 6 Verilog HDL code for the n-bit 4-to-1 multiplexer

```
/*
Filename       : tb_Mux_4x1_nbit.v
Author         : Christian Jay Gallardo
Class          : CPE3101L
GrP. Schedule  : GRP. 4 FRIDAY 10:30AM - 1:30PM
Description    : The tb_Mux_4x1_nbit testbench tests the 4-bit multiplexer by applying all
                 selector values (S) and displaying the corresponding output (Y) for each set of input
*/

module tb_Mux_4x1_nbit;

    // Parameters and signals for the testbench
    reg [3:0] A, B, C, D;  // 4-bit input buses
    reg [1:0] S;           // Selector input (2 bits)
    wire [3:0] Y;          // Output wire

    // Instantiate the Mux_4x1_nbit module
    Mux_4x1_nbit #(4) uut (
        .A(A), .B(B), .C(C), .D(D),
        .S(S), .Y(Y)
    );

    // Test procedure
    initial begin
        // Apply test vectors
        A = 4'b0001; B = 4'b0010; C = 4'b0100; D = 4'b1000;

        // Test case 1: S = 00
        S = 2'b00; #10;
        $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);

        // Test case 2: S = 01
        S = 2'b01; #10;
        $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);

        // Test case 3: S = 10
        S = 2'b10; #10;
        $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);

        // Test case 4: S = 11
        S = 2'b11; #10;
        $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);

        // End of simulation
        $finish;
    end

endmodule
```

Figure 7 Verilog HDL testbench code for the 4-bit 4-to-1 multiplexer

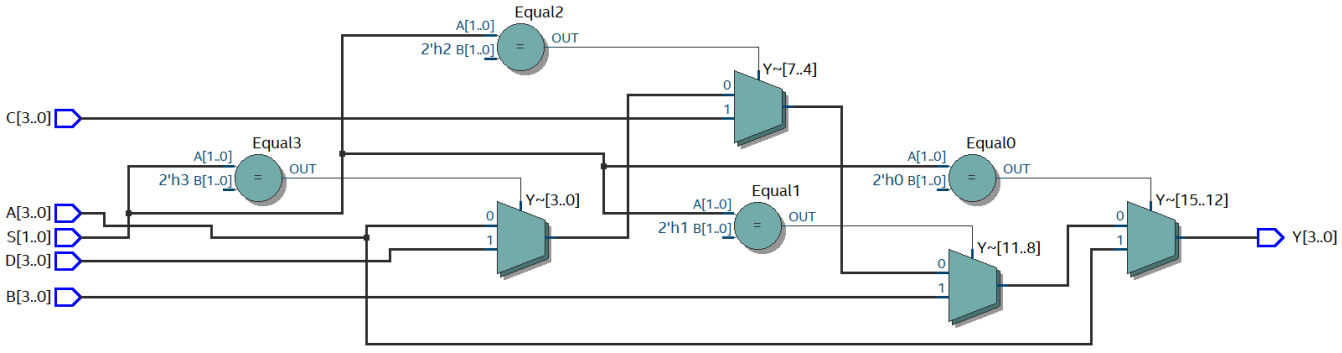


Figure 8 RTL Viewer of the n-bit 4-to-1 Multiplexer

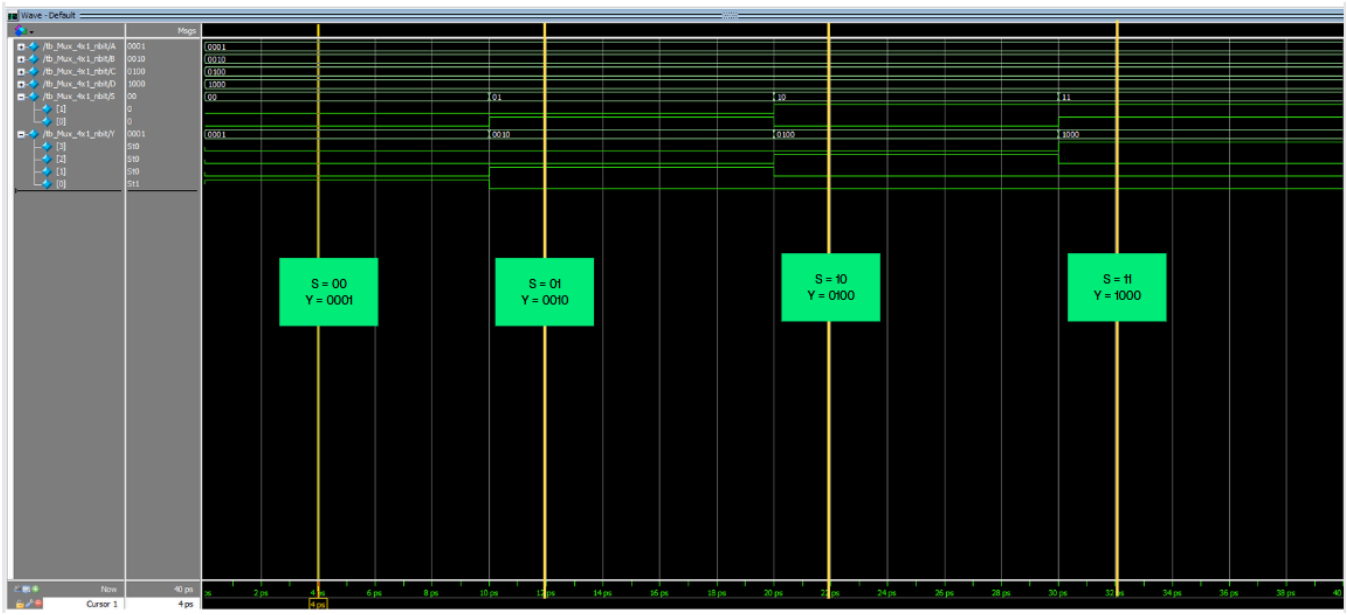


Figure 9 Successful simulation results of the 4-bit 4-to-1 Multiplexer with annotations

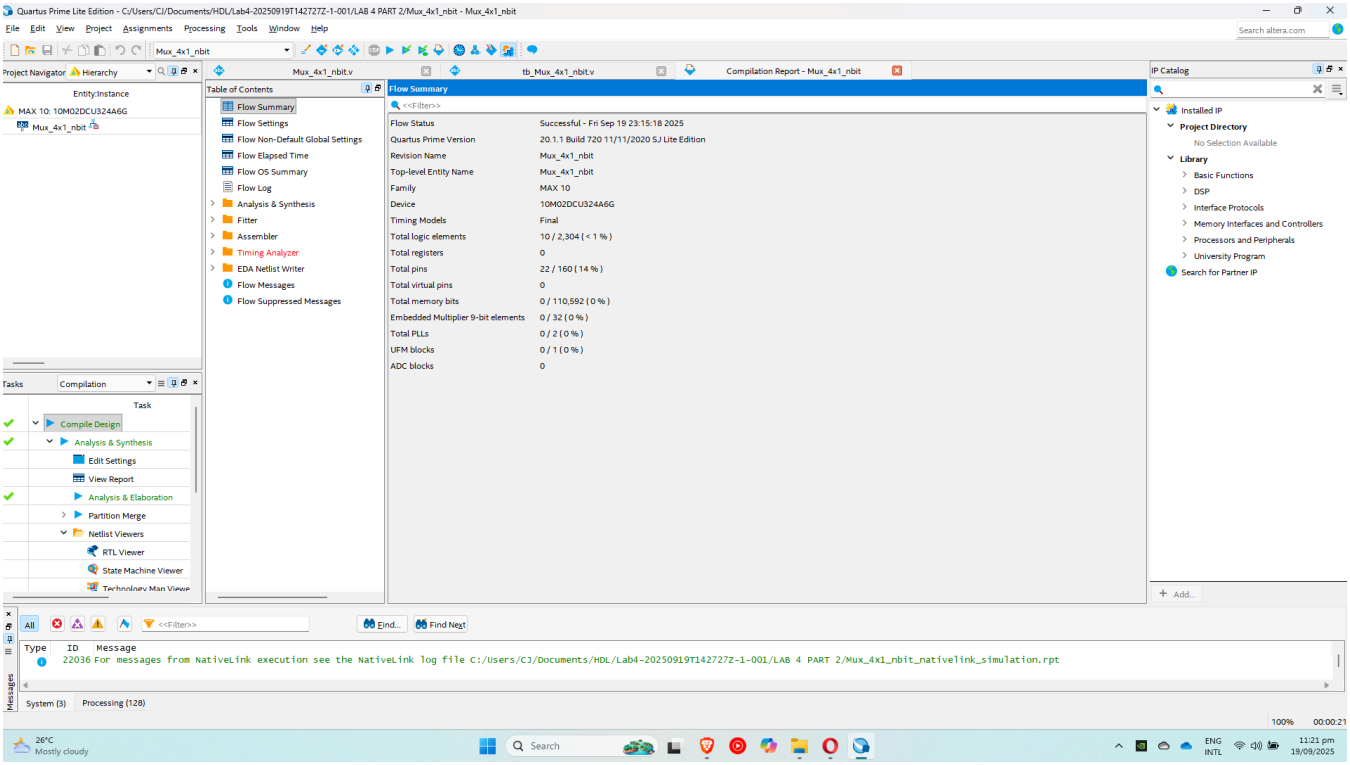


Figure 10 Summary of Design Synthesis of the 4-bit 4-to-1 Multiplexer with annotations

Exercise 4B: Part 2

Table 2 Truth Table

S (Selector)	A (Input)	B (Input)	C (Input)	D (Input)	Y (Output)
00	1111	0000	1010	0101	1111
01	1111	0000	1010	0101	0000
10	1111	0000	1010	0101	1010
11	1111	0000	1010	0101	0101

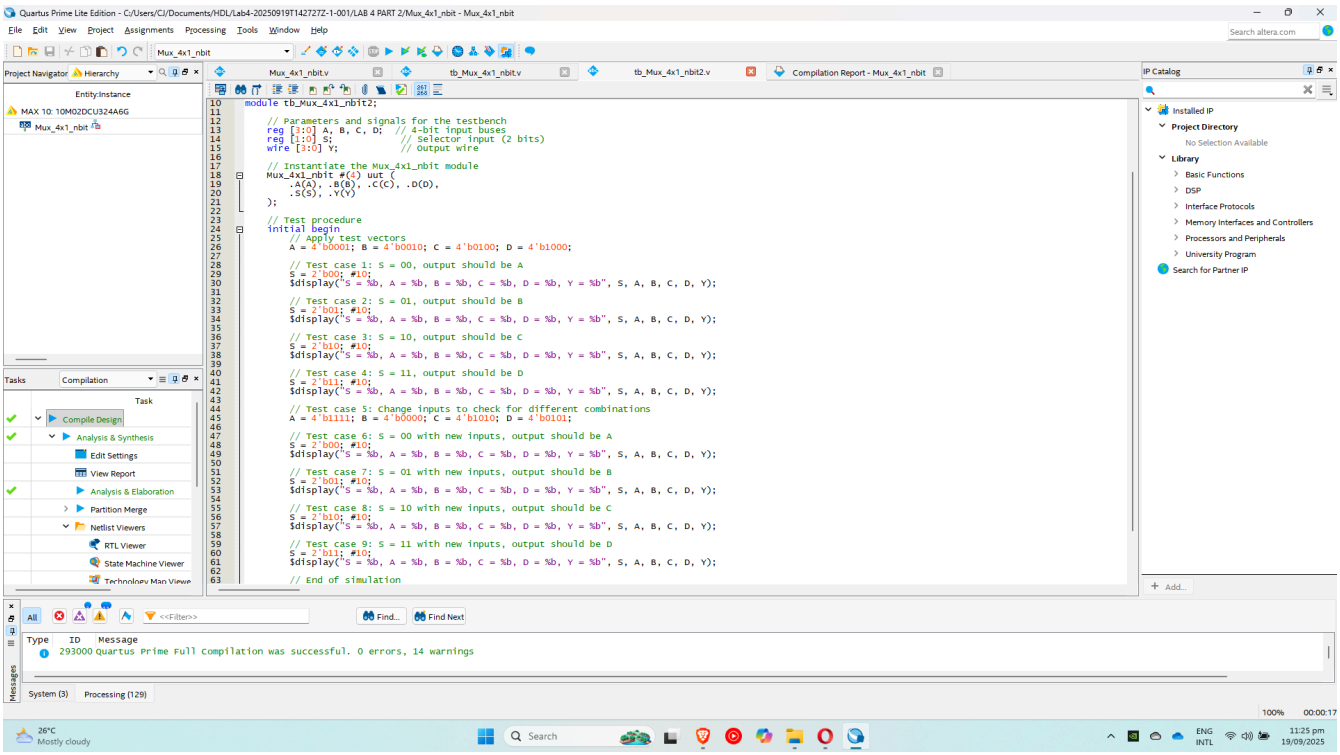


Figure 11 Verilog HDL testbench code for the 4-bit 4-to-1 multiplexer

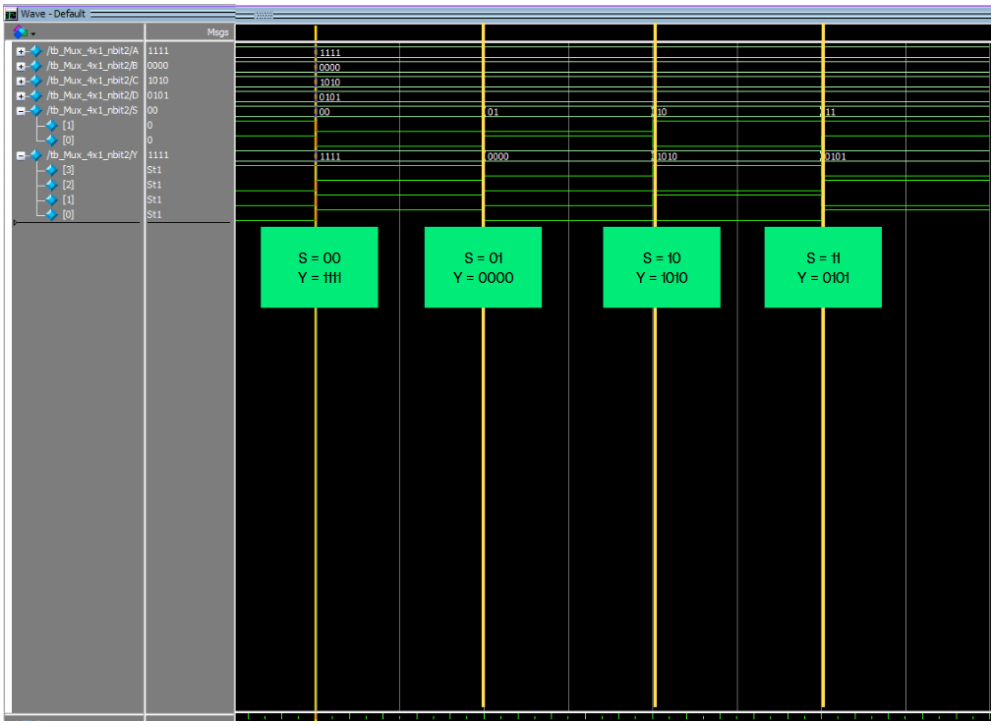


Figure 12 Successful simulation results of the 8-bit 4-to-1 Multiplexer with annotations

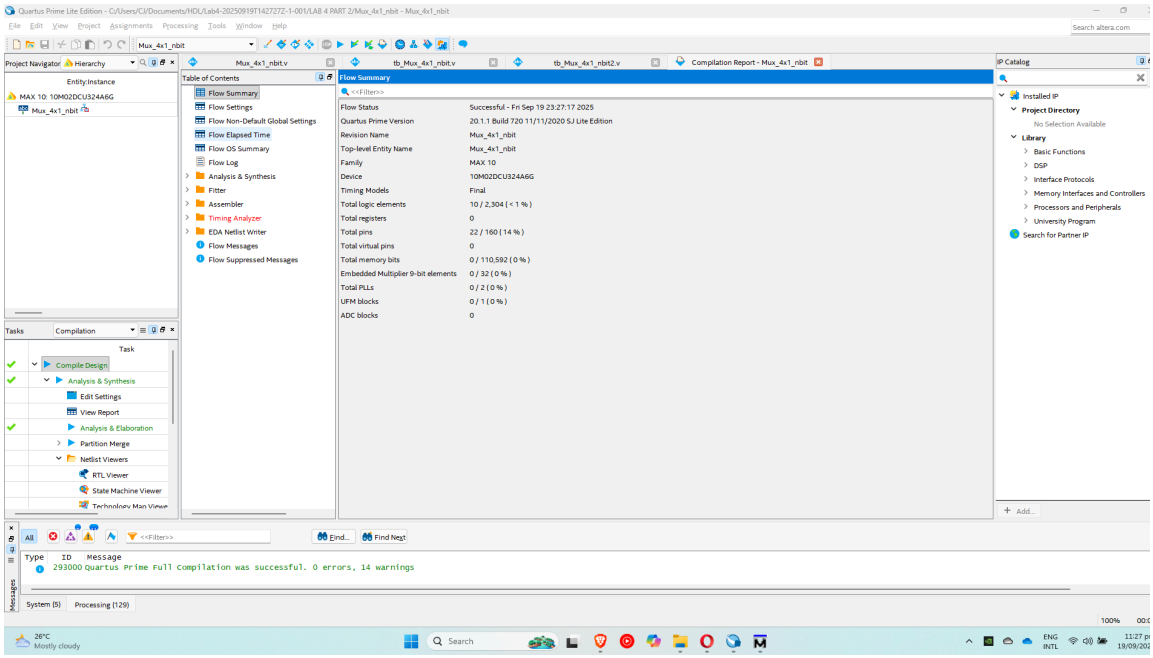


Figure 13 Summary of Design Synthesis of the 8-bit 4-to-1 Multiplexer with annotations