



Laboratory Report # 4

Name: Christian Jay Gallardo

Date Completed: 9/19/2025

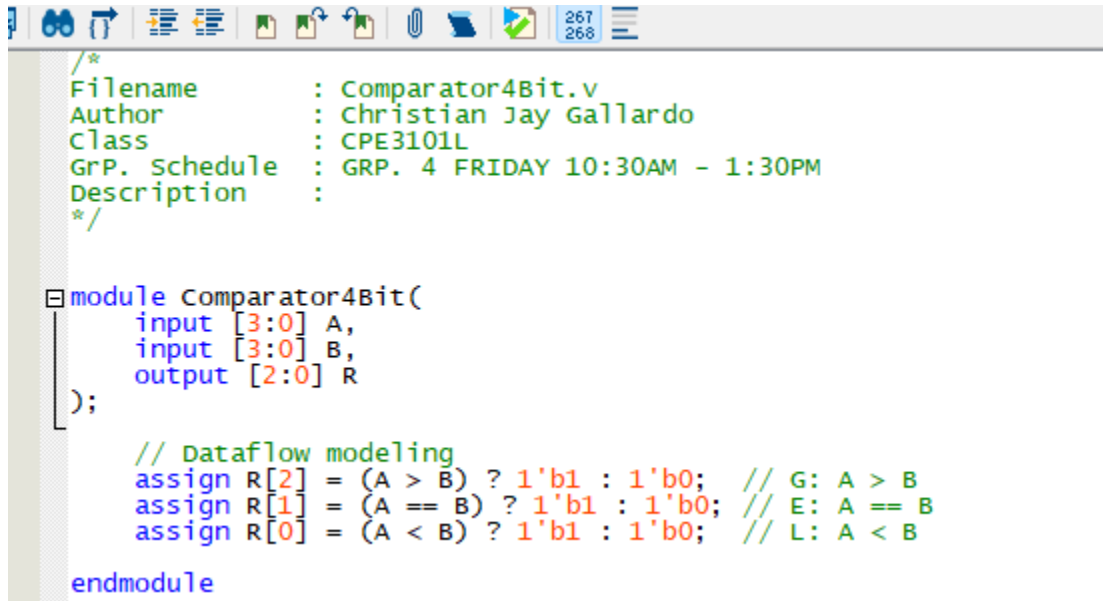
Laboratory Exercise Title: Dataflow Modeling of Combinational Circuits

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



```
7 //
8 module tb_comparator4bit;
9     reg [2:0] A, B;
10    wire [2:0] R;
11
12    // Instantiate the comparator
13    comparator4bit uut(
14        .A(A),
15        .B(B),
16        .R(R)
17    );
18
19    initial begin
20        // Test Case 1: A > B
21        A = 4'b0101; B = 4'b0101; #10;
22        $display("Test 1: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
23
24        // Test Case 2: A < B
25        A = 4'b0011; B = 4'b1100; #10;
26        $display("Test 2: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
27
28        // Test Case 3: A = B
29        A = 4'b1111; B = 4'b1111; #10;
30        $display("Test 3: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
31
32        // Test Case 4: A > B (edge case)
33        A = 4'b0001; B = 4'b0000; #10;
34        $display("Test 4: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
35
36        // Test Case 5: A < B (edge case)
37        A = 4'b0000; B = 4'b0001; #10;
38        $display("Test 5: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
39
40        // Test Case 6: A > B
41        A = 4'b1100; B = 4'b1011; #10;
42        $display("Test 6: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
43
44        // Test Case 7: A < B
45        A = 4'b0101; B = 4'b1010; #10;
46        $display("Test 7: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
47
48        // Test Case 8: A = B
49        A = 4'b1010; B = 4'b1010; #10;
50        $display("Test 8: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
51
52        // Test Case 9: A > B
53        A = 4'b1110; B = 4'b1101; #10;
54        $display("Test 9: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
55
56        // Test Case 10: A < B
57        A = 4'b0010; B = 4'b0011; #10;
58        $display("Test 10: A=%b, B=%b, R=%b (G=%b, E=%b, L=%b)", A, B, R, R[2], R[1], R[0]);
59
60    end
```

Messages

Type ID Message

22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/20.1/quartus/common/tcl/internal/nativeLink/qnativeLink.tcl" --rtl_sim "comparator4bit" "comparator4bit")

22036 For messages from NativeLink execution see the NativeLink log file c:/Users/CJ/Documents/HDL/Lab4-20250919/1427272-1-001/Lab4/comparator4bit_nativeLink_simulation.rpt

System (4) Processing (128)

26°C Mostly cloudy

Search

ENG INTL 11:01 pm 19/09/2023

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

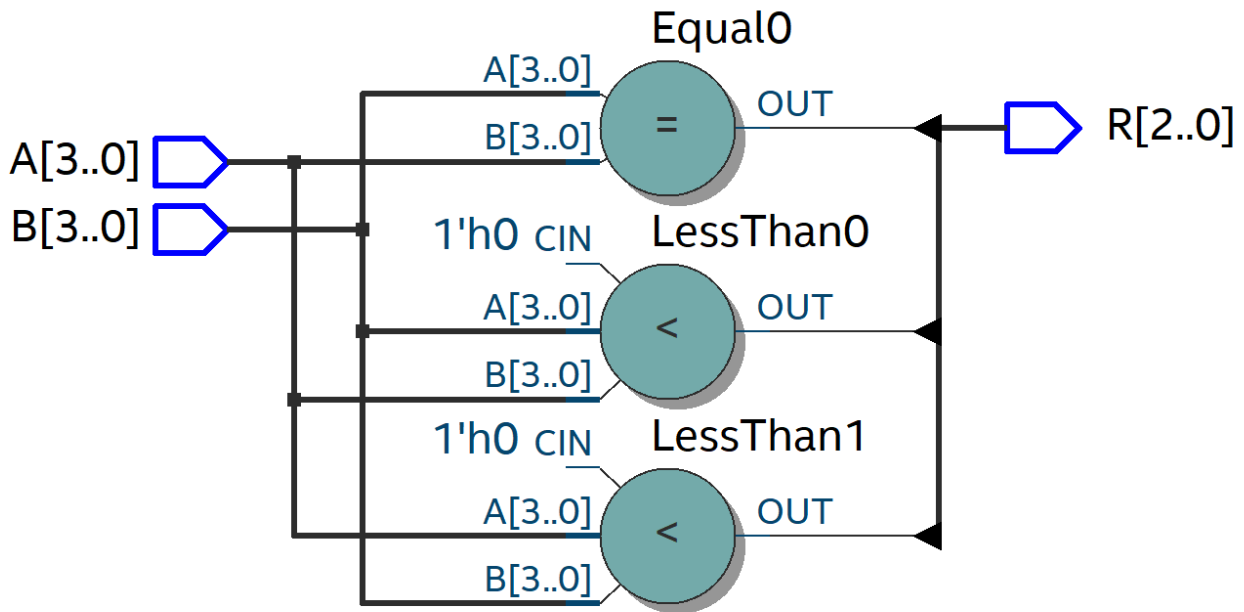


Figure 3 RTL Viewer of the 4-bit Comparator

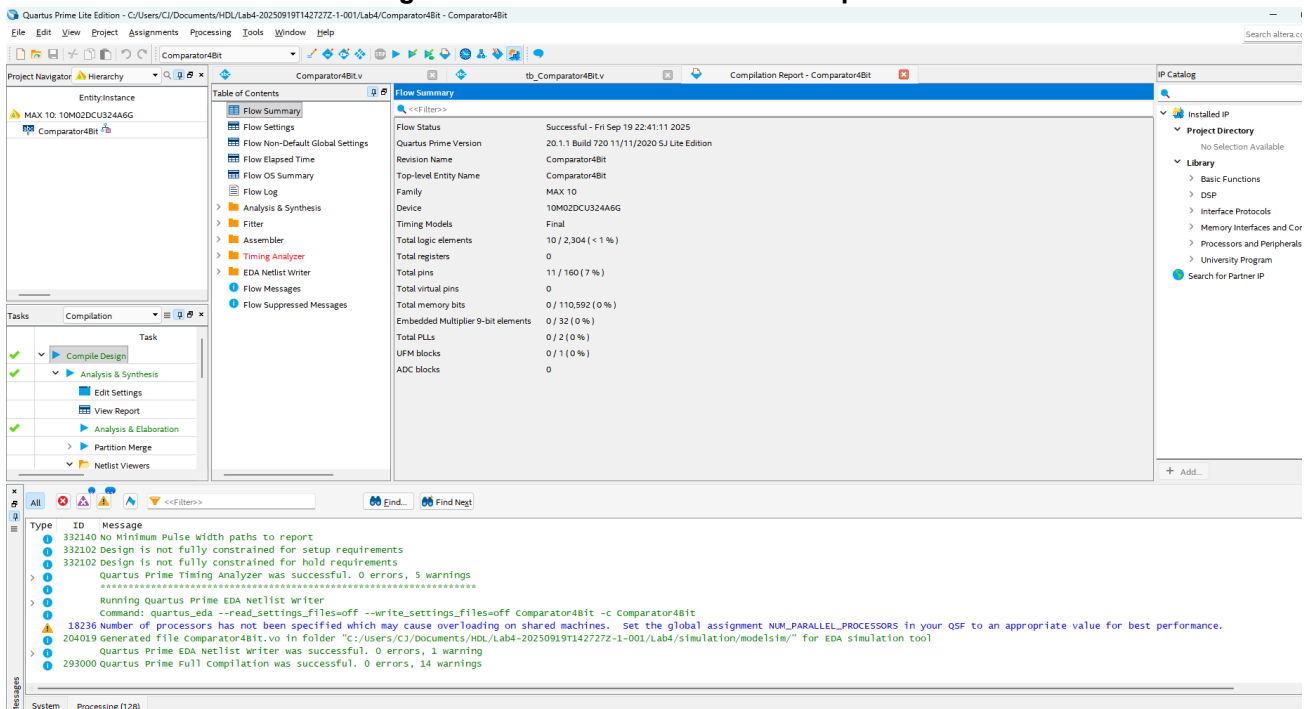


Figure 4 Summary of Design Synthesis for the 4-bit comparator



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

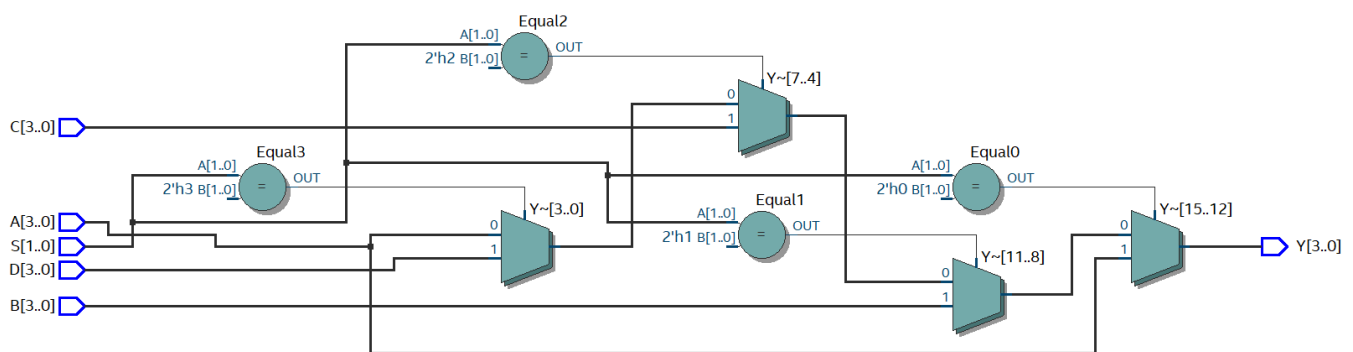


Figure 8 RTL Viewer of the n-bit 4-to-1 Multiplexer



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

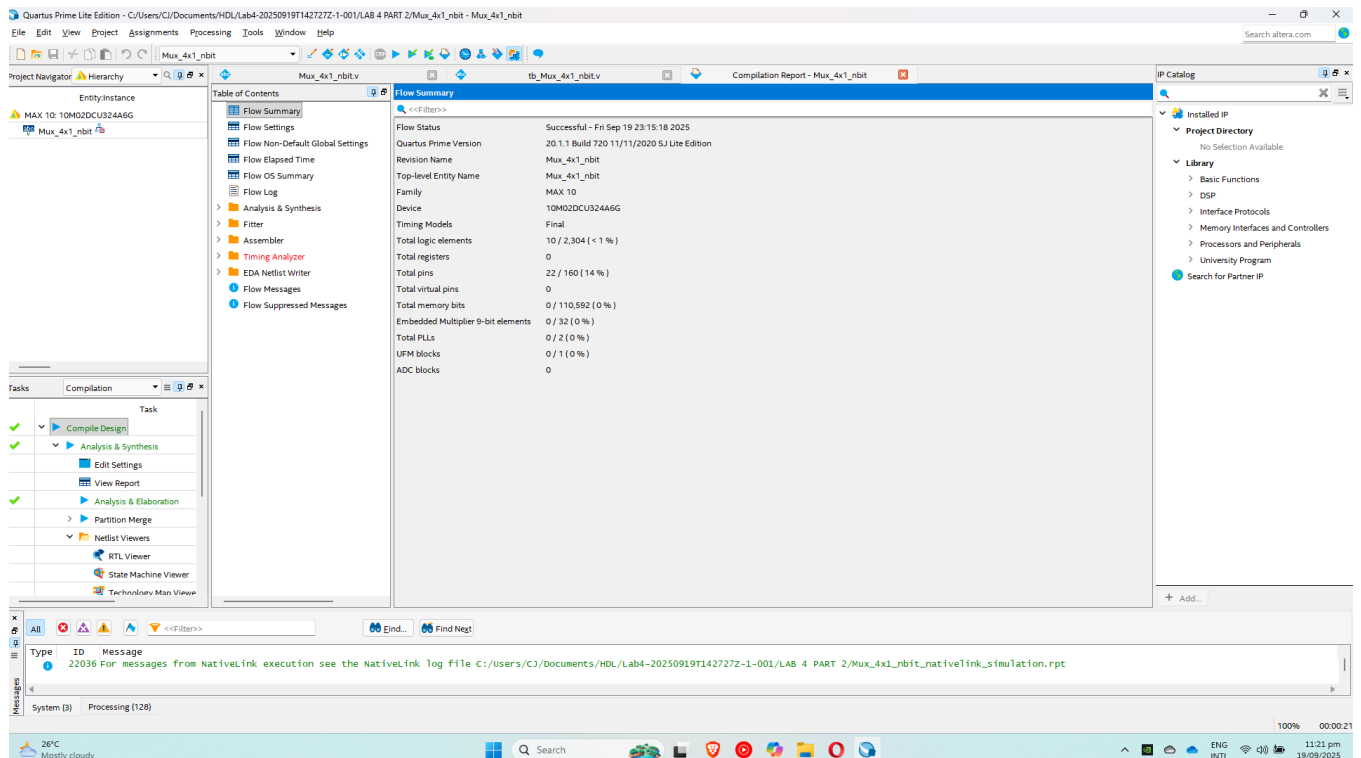


Figure 10 Summary of Design Synthesis of the 4-bit 4-to-1 Multiplexer with annotations



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

```
10 module tb_mux_4x1_nbit2;
11
12 // Parameters and signals for the testbench
13 reg [3:0] A, B, C, D; // 4-bit input buses
14 reg [1:0] S; // Selector input (2 bits)
15 wire [3:0] Y; // output wire
16
17 // Instantiate the Mux_4x1_nbit module
18 Mux_4x1_nbit #0 uut (
19     .A(A), .B(B), .C(C), .D(D),
20     .S(S), .Y(Y)
21 );
22
23 // Test procedure
24 initial begin
25     // Apply test vectors
26     A = 4'b0001; B = 4'b0010; C = 4'b0100; D = 4'b1000;
27
28     // Test case 1: S = 00, output should be A
29     S = 2'b00; #10;
30     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
31
32     // Test case 2: S = 01, output should be B
33     S = 2'b01; #10;
34     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
35
36     // Test case 3: S = 10, output should be C
37     S = 2'b10; #10;
38     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
39
40     // Test case 4: S = 11, output should be D
41     S = 2'b11; #10;
42     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
43
44     // Test case 5: Change inputs to check for different combinations
45     A = 4'b1111; B = 4'b0000; C = 4'b0101; D = 4'b0101;
46
47     // Test case 6: S = 00 with new inputs, output should be A
48     S = 2'b00; #10;
49     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
50
51     // Test case 7: S = 01 with new inputs, output should be B
52     S = 2'b01; #10;
53     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
54
55     // Test case 8: S = 10 with new inputs, output should be C
56     S = 2'b10; #10;
57     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
58
59     // Test case 9: S = 11 with new inputs, output should be D
60     S = 2'b11; #10;
61     $display("S = %b, A = %b, B = %b, C = %b, D = %b, Y = %b", S, A, B, C, D, Y);
62
63 // End of simulation
```

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

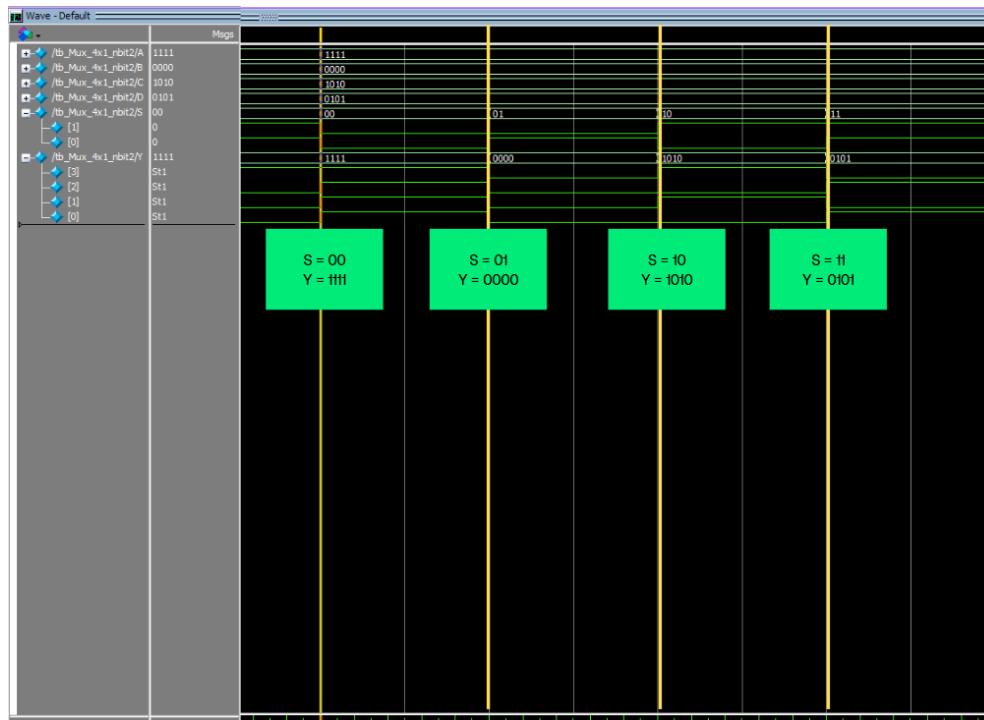


Figure 12 Successful simulation results of the 8-bit 4-to-1 Multiplexer with annotations

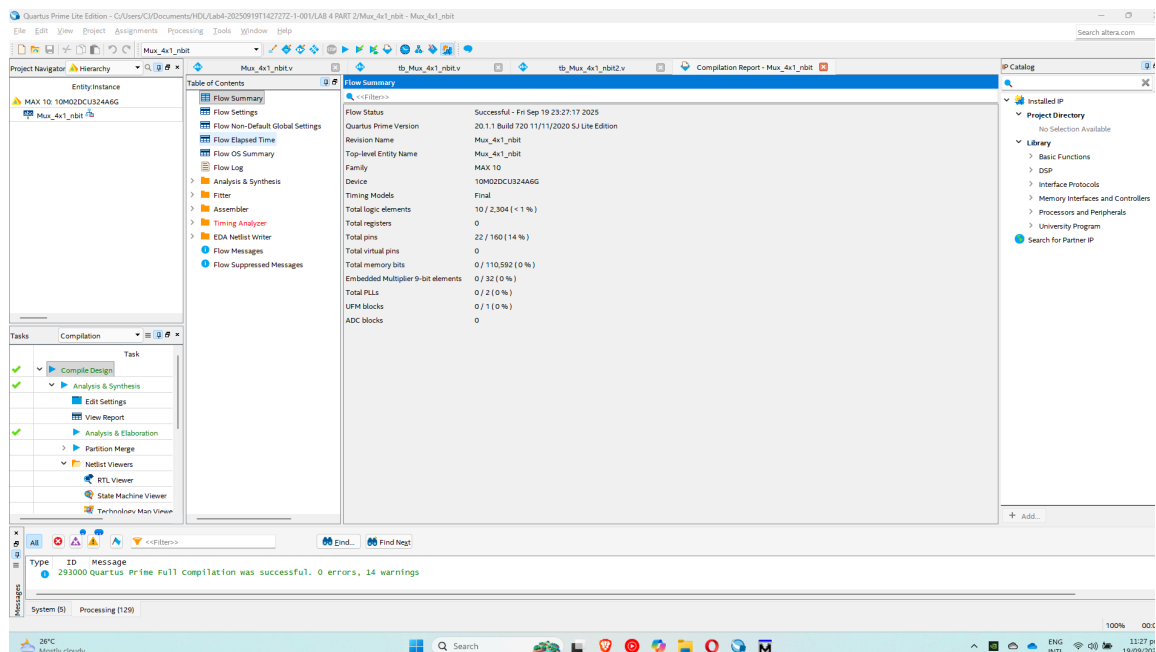


Figure 13 Summary of Design Synthesis of the 8-bit 4-to-1 Multiplexer with annotations