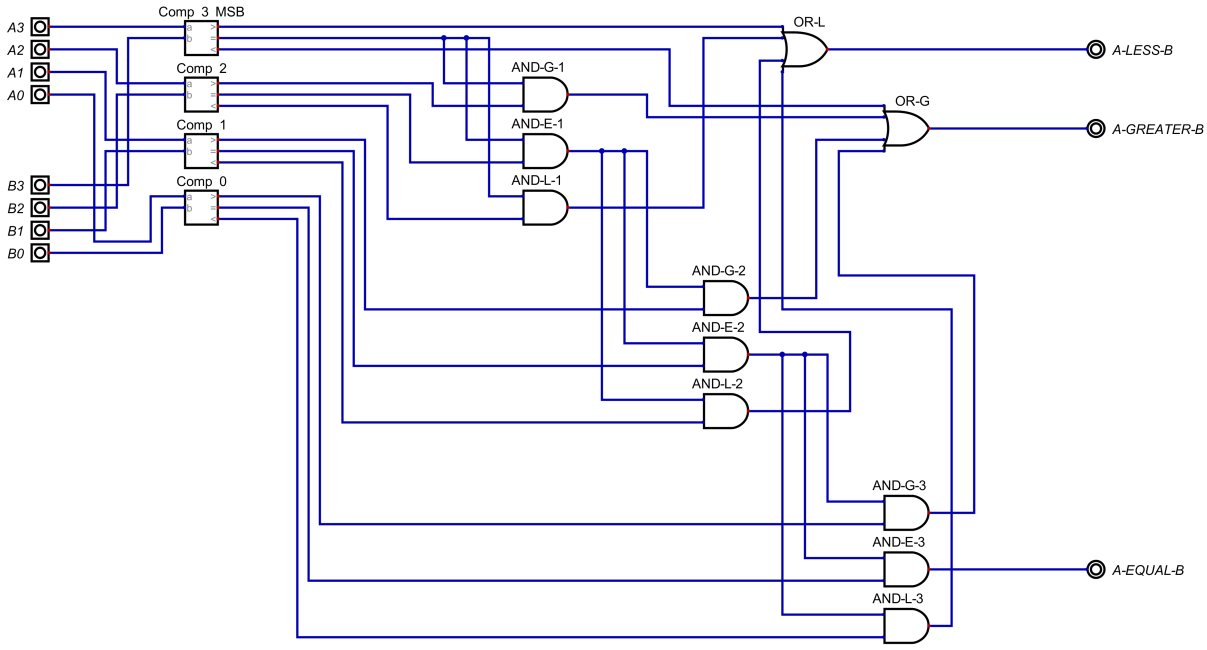
PRE-LAB #3 Report

Berna Yıldıran 26431

Circuit Design

As the initial step of this lab project, the “4-Bit Signed Comparator Circuit” is designed using Digital software. The circuit design can be seen in Figure 1.



Circuit Flow

Step 1: MSBs (A3 and B3) of the input A and B is compared using 1-bit comparator (Comp 3 MSB).

- If A3 > B3, the output will be sent to OR-L GATE before the A-LESS-B output. (w1)
- If A3 = B3, the output will be sent to AND-G-1, AND-E-1, AND-L-1 GATES, in order to add the result of A2 - B2 comparison to the circuit to decide the final output. (w2)
- If A3 < B3, the output will be sent to OR-G GATE before the A-GREATER-B output. (w3)
- NOTE: Since A and B are signed integers, and 1 represent negative numbers and 0 represent positive numbers, while A (1) > B (0) the wire connects to the OR-L GATE before the A-LESS-B output. In the same way, when A (0) < B (1) the wire connects to the OR-G GATE before the A-GREATER-B output.

Step 2: When MSBs of A and B are equal, the circuit compares A2 and B2 bits of the inputs using 1-bit comparator (Comp 2).

- If A2 > B2, the output will be sent to AND-G-1 GATE together with w2 and the result of this operation will be connected to the OR-G GATE before the A-GREATER-B output. (w4)
- If A2 = B2, the output will be sent to AND-E-1 GATE together with w2 and the result of this operation will sent to AND-G-2, AND-E-2, AND-L-2 GATES, in order to add the result of A1 - B1 comparison to the circuit to decide the final output. (w5)
- If A2 < B2, the output will be sent to AND-L-1 GATE together with w2 and the result of this operation will be connected to the OR-L GATE before the A-LESS-B output. (w6)

Step 3: If first two bits of A and B are equal, the circuit compares A1 and B1 bits of the inputs using 1-bit comparator (Comp 1).

- If A1 > B1, the output will be sent to AND-G-2 GATE together with w6 and the result of this operation will be connected to the OR-G GATE before the A-GREATER-B output. (w7)
- If A1 = B1, the output will be sent to AND-E-2 GATE together with w6 and the result of this operation will sent to AND-G-3, AND-E-3, AND-L-3 GATES in order to add the result of A0 - B0 comparison to the circuit to decide the final output. (w8)
- If A1 < B1, the output will be sent to AND-L-2 GATE together with w6 and the result of this operation will be connected to the OR-L GATE before the A-LESS-B output. (w9)

Step 4: If first three bits of A and B are equal, the circuit compares A0 and B0 bits of the inputs using 1-bit comparator (Comp 0).

- If A0 > B0, the output will be sent to AND-G-3 GATE together with w8 and the result of this operation will be connected to the OR-G GATE before the A-GREATER-B output. (w10)
- If A0 = B0, the output will be sent to AND-E-3 GATE together with w8 and the result of this operation will sent to output A-EQUAL-B. (w11)
- If A0 < B0, the output will be sent to AND-L-3 GATE together with w8 and the result of this operation will be connected to the OR-L GATE before the A-LESS-B output. (w12)

Step 5: Depending on the input wires of OR-G and OR-L GATES, the output of the circuit based on the input A and B will be determined.

- Based on the values of w1, w6, w9, w12, which are the inputs of the OR-L GATE, if any of the inputs are 1, the circuit will output the result of A - B comparison as A-LESS-B.
- Based on the values of w3, w4, w7, w10, which are the inputs of the OR-G GATE, if any of the inputs are 1, the circuit will output the result of A - B comparison as A-GREATER-B.

Verilog: Design Sources

The source code of the above circuit is written in Verilog as the following:

```
`timescale 1ns / 1ps
module CompUnsigned #(
    parameter Bits = 1
)
(
    input [(Bits -1):0] a,
    input [(Bits -1):0] b,
    output \> ,
    output \= ,
    output \<
);
    assign \> = a > b;
    assign \= = a == b;
    assign \< = a < b;
endmodule

module Comparator_Signed_4Bit(
      input A3,
      input A2,
      input A1,
      input A0,
      input B3,
      input B2,
      input B1,
      input B0,
      output A_GREATER_B,
      output A_LESS_B,
      output A_EQUAL_B
      );

      wire s0;
      wire s1;
      wire s2;
      wire s3;
      wire s4;
      wire s5;
      wire s6;
      wire s7;
      wire s8;
      wire s9;
      wire s10;
      wire s11;
      wire s12;
      wire s13;

      // ----------------------------
      // Comp 3 MSB
      CompUnsigned #(
        .Bits(1)
      )
      CompUnsigned_i0 (
        .a( A3 ),
        .b( B3 ),
        .\> ( s0 ),
        .\= ( s1 ),
        .\< ( s2 )
      );

      // ----------------------------
      // Comp 2
      CompUnsigned #(
        .Bits(1)
      )
      CompUnsigned_i1 (
        .a( A2 ),
        .b( B2 ),
        .\> ( s3 ),
```

```
        .\= ( s4 ),
        .\< ( s5 )
      );

      // ----------------------------
      // Comp 1
      CompUnsigned #(
        .Bits(1)
      )
      CompUnsigned_i2 (
        .a( A1 ),
        .b( B1 ),
        .\> ( s6 ),
        .\= ( s7 ),
        .\< ( s8 )
      );

      // ----------------------------
      // Comp 0
      CompUnsigned #(
        .Bits(1)
      )
      CompUnsigned_i3 (
        .a( A0 ),
        .b( B0 ),
        .\> ( s9 ),
        .\= ( s10 ),
        .\< ( s11 )
      );

      // ----------------------------
      // OUTPUTS
      assign s12 = (s1 & s4);
      assign s13 = (s12 & s7);
      assign A_EQUAL_B  = (s13 & s10);
      assign A_LESS_B  = (s0 | (s1 & s5) | (s12 & s8) | (s13 & s11));
      assign A_GREATER_B  = (s2 | (s1 & s3) | (s12 & s6) | (s13 & s9));
    endmodule
```

Verilog: Simulation Sources

The simulation of the circuit is written as the following. The test cases for the input A and B are:

- **Test Case #1**
  A → 0000 (0)
  B → 1000 (-8)
- **Test Case #2**
  A → 1000 (-1)
  B → 1111 (-8)
- **Test Case #3**
  A → 0111 (+7)
  B → 0011 (+3)
- **Test Case #4**
  A → 1001 (-7)
  B → 0001 (+1)
- **Test Case #5**
  A → 0100 (+4)
  B → 1100 (-4)
- **Test Case #6**
  A → 1110 (-2)
  B → 1110 (-2)
- **Test Case #7**
  A → 0100 (+5)
  B → 0010 (+2)
- **Test Case #8**
  A → 1111 (-1)
  B → 1111 (-1)

```
`timescale 1ns / 1
module Sim_Comp_Signed_4Bit(
      );

      // INPUTS
```

```
        reg A3;
        reg A2;
        reg A1;
        reg A0;
        reg B3;
        reg B2;
        reg B1;
        reg B0;

        // OUTPUTS
        wire A_GREATER_B;
        wire A_LESS_B;
        wire A_EQUAL_B;


        Comparator_Signed_4Bit UUT (
        .A3(A3),
        .A2(A2),
        .A1(A1),
        .A0(A0),

        .B3(B3),
        .B2(B2),
        .B1(B1),
        .B0(B0),

        .A_GREATER_B(A_GREATER_B),
        .A_LESS_B(A_LESS_B),
        .A_EQUAL_B(A_EQUAL_B)
        );

        // INITIALIZE INPUTS
        initial begin
            // ---------------------------------------
            // TEST #1
            // A = 0 // B = -8
            // ---------------------------------------
            A3 = 0;
            A2 = 0;
            A1 = 0;
            A0 = 0;

            B3 = 1;
            B2 = 0;
            B1 = 0;
            B0 = 0;
            #10;

            // ---------------------------------------
            // TEST #2
            // A = -8 // B = -1
            // ---------------------------------------
            A3 = 1;
            A2 = 0;
            A1 = 0;
            A0 = 0;

            B3 = 1;
            B2 = 1;
            B1 = 1;
            B0 = 1;
            #10;

            // ---------------------------------------
            // TEST #3
            // A = +7 // B = +3
            // ---------------------------------------
            A3 = 0;
```

```
        A2 = 1;
        A1 = 1;
        A0 = 1;

        B3 = 0;
        B2 = 0;
        B1 = 1;
        B0 = 1;
        #10;

        // -------------------------------------
        // TEST #4
        // A = -7 // B = +1
        // -------------------------------------
        A3 = 1;
        A2 = 0;
        A1 = 0;
        A0 = 1;

        B3 = 0;
        B2 = 0;
        B1 = 0;
        B0 = 1;
        #10;

        // -------------------------------------
        // TEST #5
        // A = +4 // B = -4
        // -------------------------------------
        A3 = 0;
        A2 = 1;
        A1 = 0;
        A0 = 0;

        B3 = 1;
        B2 = 1;
        B1 = 0;
        B0 = 0;
        #10;

        // -------------------------------------
        // TEST #6
        // A = -2 // B = -2
        // -------------------------------------
        A3 = 1;
        A2 = 1;
        A1 = 1;
        A0 = 0;

        B3 = 1;
        B2 = 1;
        B1 = 1;
        B0 = 0;
        #10;

        // -------------------------------------
        // TEST #7
        // A = 5 // B = 2
        // -------------------------------------
        A3 = 0;
        A2 = 1;
        A1 = 0;
        A0 = 0;

        B3 = 0;
        B2 = 0;
        B1 = 1;
        B0 = 0;
```

```
          #10;

          // -------------------------------------
          // TEST #8
          // A = -1 // B = -1
          // -------------------------------------
          A3 = 1;
          A2 = 1;
          A1 = 1;
          A0 = 1;

          B3 = 1;
          B2 = 1;
          B1 = 1;
          B0 = 1;
          #10;

      end
endmodule
```

Verilog: Constraints

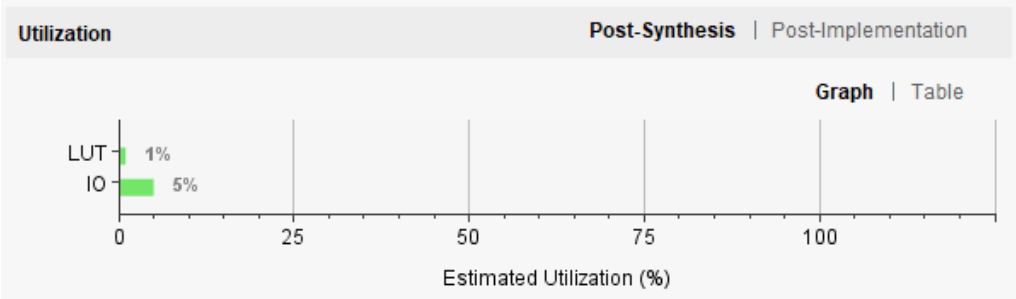
The constraints file for exporting circuit to the FPGA board are the following. Comments about the code are written in order to better explaining the code.

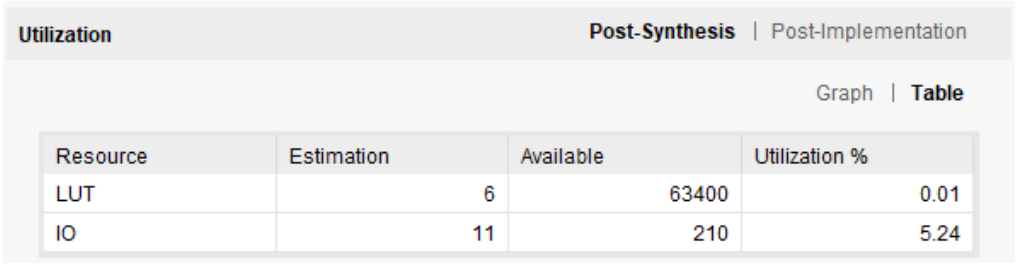
```
# For the constraint file, 8 switches and 3 LEDs will be needed.
# Switches for the first 4-Bit Number --> R15, M13, L16, J15
# Switches for the second 4-Bit Number --> T8, R13, U18, T18
# LED for A>B: H17
# LED for A<B: K15
# LED for A=B: J13

# ---------------------------------
# INPUT SWITCHES A
# ---------------------------------
# SW A3
set_property -dict { PACKAGE_PIN R15 IOSTANDARD LVCMOS33 } [get_ports { A3 }];
# SW A2
set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMOS33 } [get_ports { A2 }];
# SW A1
set_property -dict { PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports { A1 }];
# SW A0
set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [get_ports { A0 }];


# ---------------------------------
# INPUT SWITCHES B
# ---------------------------------
# SW B3
set_property -dict { PACKAGE_PIN T8 IOSTANDARD LVCMOS33 } [get_ports { B3 }];
# SW B2
set_property -dict { PACKAGE_PIN R13 IOSTANDARD LVCMOS33 } [get_ports { B2 }];
# SW B1
set_property -dict { PACKAGE_PIN U18 IOSTANDARD LVCMOS33 } [get_ports { B1 }];
# SW B0
set_property -dict { PACKAGE_PIN T18 IOSTANDARD LVCMOS33 } [get_ports { B0 }];


# ---------------------------------
# OUTPUT LEDS
# ---------------------------------
# LED A_GREATER_B
```

```
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { A_GREATER_B }];
# LED A_LESS_B
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { A_LESS_B }];
# LED A_EQUAL_B
set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { A_EQUAL_B }];
```

Verilog: Synthesis

Synthesis results of the 4-Bit Signed Comparator circuit.

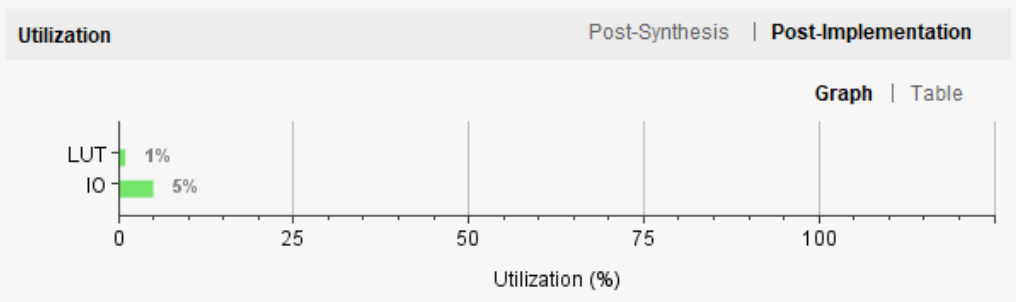




Resource	Estimation	Available	Utilization %
LUT	6	63400	0.01
IO	11	210	5.24

Verilog: Implementation

Implementation results of the 4-Bit Signed Comparator circuit.



Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	6	63400	0.01
IO	11	210	5.24

Verilog: Project Summary

Project summary of the 4-Bit Signed Comparator circuit.