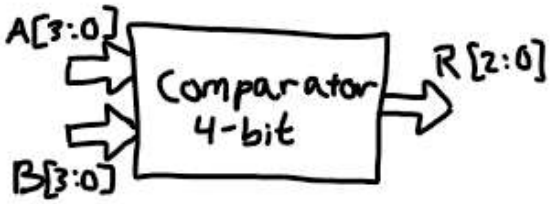
Laboratory Report #4

Name: Josh Ratificar **Date Completed:** 10-02-2023

Laboratory Exercise Title: Dataflow Modeling of Combinational Circuits

Block Diagrams:



Above includes the block diagrams for this Laboratory Exercise. Based on the diagrams, the following can be concluded:

4-Bit Comparator

8-Input Ports:

- **A[0], A[1], A[2], A[3], B[0], B[1], B[2], B[3]**

3-Output Ports:

- **R[2], R[1], R[0]**

NOTE

G ("greater than" is TRUE) | R [2] = G (1 if A > B, else 0)

E ("equal to" is TRUE) | R [1] = E (1 if A = B, else 0)

L ("less than" is TRUE) | R [0] = L (1 if A < B, else 0)

Exercise 4A:

```
1  /************************************
2  *  FILE:              FourBitComparator.v
3  *  AUTHOR:            Josh Ratificar
4  *  Class:             Gr.3 CpE 3101L Introduction to HDL
5  *  Group/Schedule     Friday, 7:30 AM to 10:30 AM
6  *  Description:       FourBitComparator.v module
7  ************************************/
8  module FourBitComparator(
9      input    [3:0]A,
10     input    [3:0]B,
11     output   [2:0]R
12 );
13     wire     [2:0]common;
14     // A > B @ R2 |R [2] = G (1 if A > B, else 0)
15     assign R[2] = (A > B)? 1'b1 : 1'b0;
16
17
18     // A = B @ R[1] | R[1] = E (1 if A = B, else 0)
19     assign R[1] = (A == B)? 1'b1 : 1'b0;
20
21     // A < B @ R[1] | R[1] = E (1 if A < B, else 0)
22     assign R[0] = (A < B)? 1'b1 : 1'b0;
23 endmodule
24
```

Figure 1.0 – *FourBitComparator.v Script*

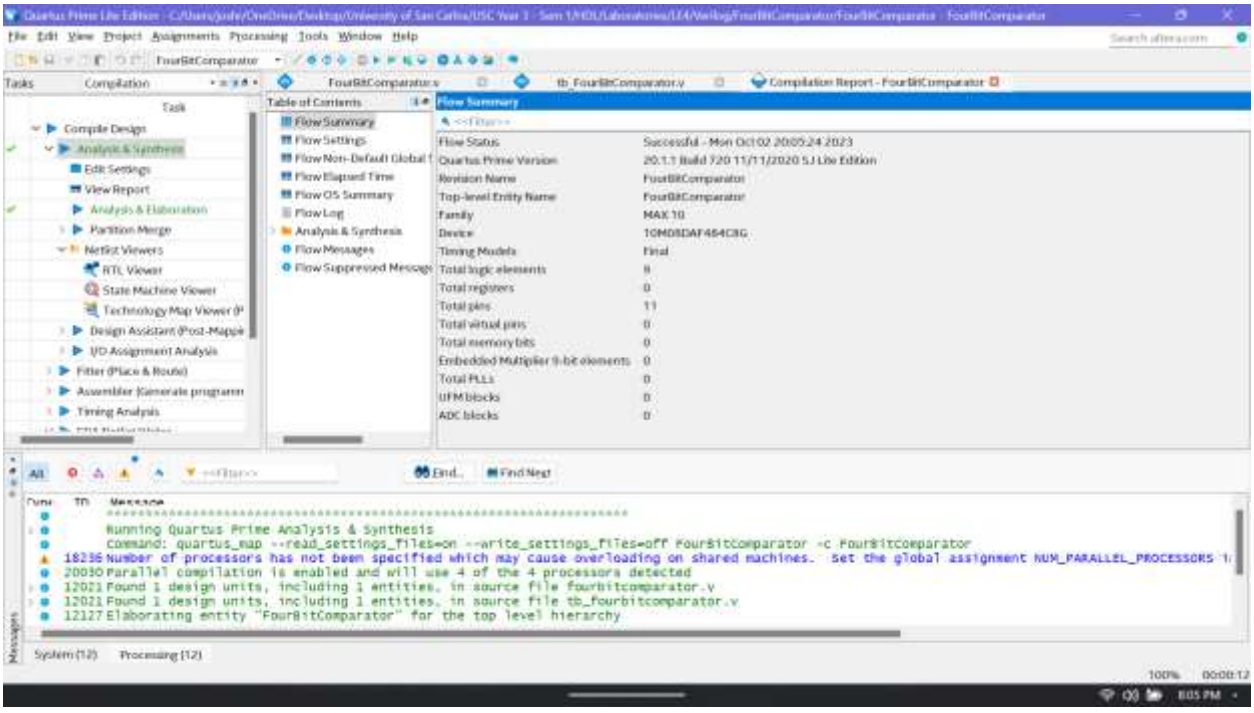


Figure 1.1 – *FourBitComparator.v Analysis and Elaboration Test Results*

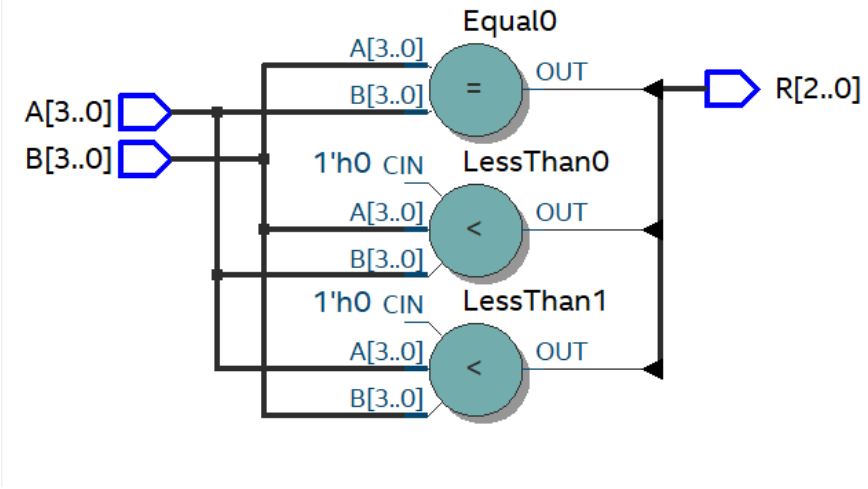


Figure 1.2 – *FourBitComparator RTL View Output*

```
/*************************************
*  FILE:              tb_FourBitComparator.v
*  AUTHOR:            Josh Ratificar
*  Class:             Gr.3 CpE 3101L Introduction to HDL
*  Group/Schedule     Friday, 7:30 AM to 10:30 AM
*  Description:       tb_FourBitComparator.v module
*************************************/
`timescale 1 ns/ 1 ps
module tb_FourBitComparator();
   reg   [3:0] A, B;
   wire  [2:0] R;
   FourBitComparator UUT(
      .A(A),
      .B(B),
      .R(R)
   );
   initial begin
      $display("Test Bench | tb_FourBitComparator...");
      // Test cases for A < B
      $display("A < B");
      A = 4'b0100; B = 4'b1010; #10; // A < B
      A = 4'b1100; B = 4'b1111; #10; // A < B
      A = 4'b0010; B = 4'b1101; #10; // A < B
      A = 4'b0011; B = 4'b1101; #10; // A < B

      // Test cases for A = B
      $display("A = B");
      A = 4'b1110; B = 4'b1110; #10; // A = B
      A = 4'b1111; B = 4'b1111; #10; // A = B
      A = 4'b1000; B = 4'b1000; #10; // A = B
      A = 4'b0000; B = 4'b0000; #10; // A = B

      // Test cases for A > B
      $display("A > B");
      A = 4'b0001; B = 4'b0000; #10; // A > B
      A = 4'b1111; B = 4'b1110; #10; // A > B
      A = 4'b0101; B = 4'b0011; #10; // A > B
      A = 4'b0010; B = 4'b0001; #10; // A > B
      $stop;
   end
   initial begin
      $monitor("Time = %2d ns\t A = %d | [%b]\t B = %d [%b]\t R = %b", $time, A, A, B, B, R);
   end
endmodule
```

Figure 1.3 – *tb_ FourBitComparator.v Script for Testing Module*

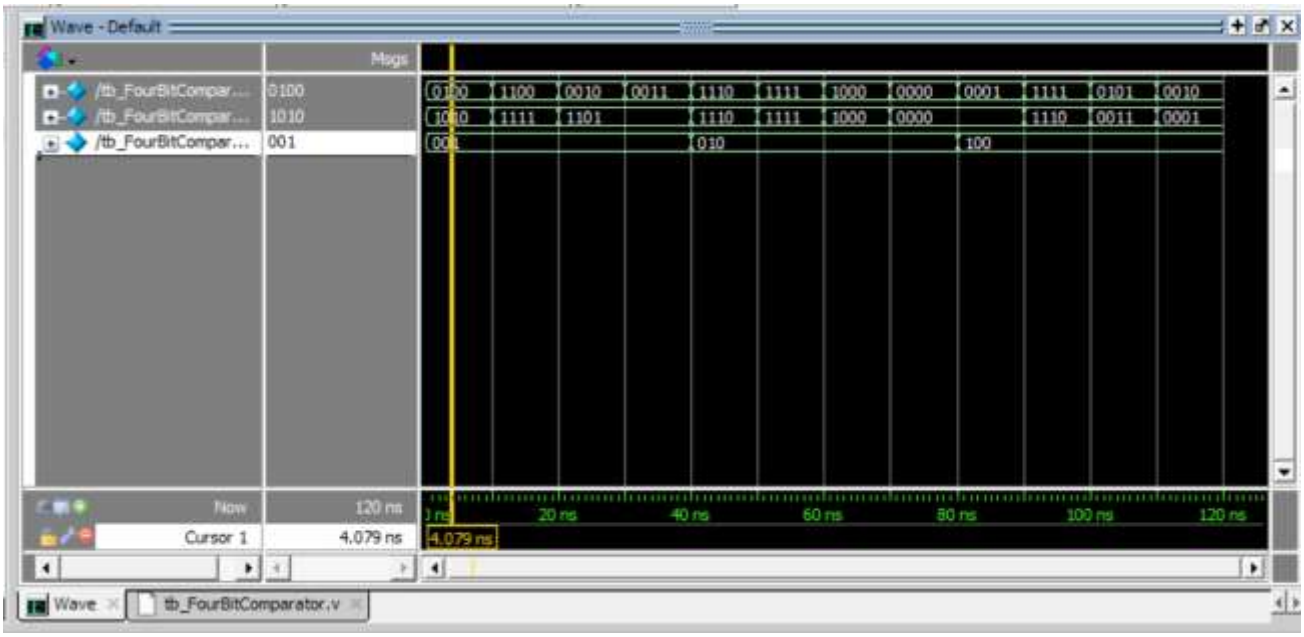


Figure 1.4 – *Four Bit Comparator RTL Simulation Output*

```
# Test Bench : tb_FourBitComparator...
# A < B
# Time =  0 ns  A =  4 : [0100]  B = 10 [1010]  R = 001
# Time = 10 ns  A = 12 : [1100]  B = 15 [1111]  R = 001
# Time = 20 ns  A =  2 : [0010]  B = 13 [1101]  R = 001
# Time = 30 ns  A =  3 : [0011]  B = 13 [1101]  R = 001
# A = B
# Time = 40 ns  A = 14 : [1110]  B = 14 [1110]  R = 010
# Time = 50 ns  A = 15 : [1111]  B = 15 [1111]  R = 010
# Time = 60 ns  A =  8 : [1000]  B =  8 [1000]  R = 010
# Time = 70 ns  A =  0 : [0000]  B =  0 [0000]  R = 010
# A > B
# Time = 80 ns  A =  1 : [0001]  B =  0 [0000]  R = 100
# Time = 90 ns  A = 15 : [1111]  B = 14 [1110]  R = 100
# Time = 100 ns  A =  5 : [0101]  B =  3 [0011]  R = 100
# Time = 110 ns  A =  2 : [0010]  B =  1 [0001]  R = 100
# ** Note: $stop    : C:/Users/joshr/OneDrive/Desktop/University of San Carlos/USC Year 3 - Sem 1/HDL/Laboratories/LE4/Verilog/FourBitComparator/tb_FourBitComparator.v(39
#    Time: 120 ns  Iteration: 0  Instance: /tb_FourBitComparator
# Break in Module tb_FourBitComparator at C:/Users/joshr/OneDrive/Desktop/University of San Carlos/USC Year 3 - Sem 1/HDL/Laboratories/LE4/Verilog/FourBitComparator/tb_Fo
parator.v line 39

VSIM 2>
```

Figure 1.5 – *Four Bit Comparator Test Bench Monitor Output (Annotations to **Figure 1.4**)*

Discussion of Results (Exercise 3A)

By observing the Test bench script (**Figure 1.3**), we expect that the respective R bit reflect the comparison between two four-bit inputs. This is the best-case desired behaviour expected which after simulating through RTL simulation (**Figure 1.4**), R appropriately and respectively changes every 40 ns. This is how the test bench was setup, and we can furthermore observe this behaviour in **Figure 1.5**. In conclusion, our design for our comparator works even when we applied Data Flow design in HDL Verilog.