

Lab 2: 4-bit Comparator

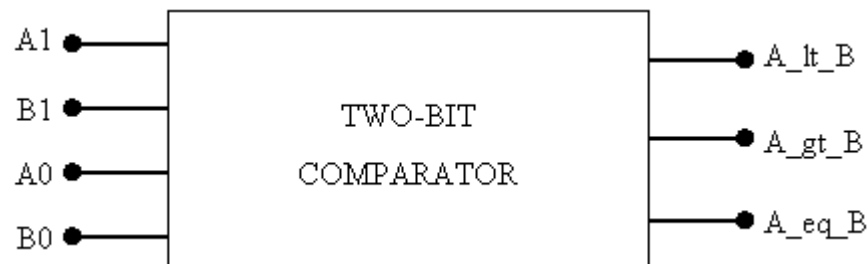
Due: March 12, 2015

Objective

- To be familiar with Verilog modeling and testbench.

Action Items

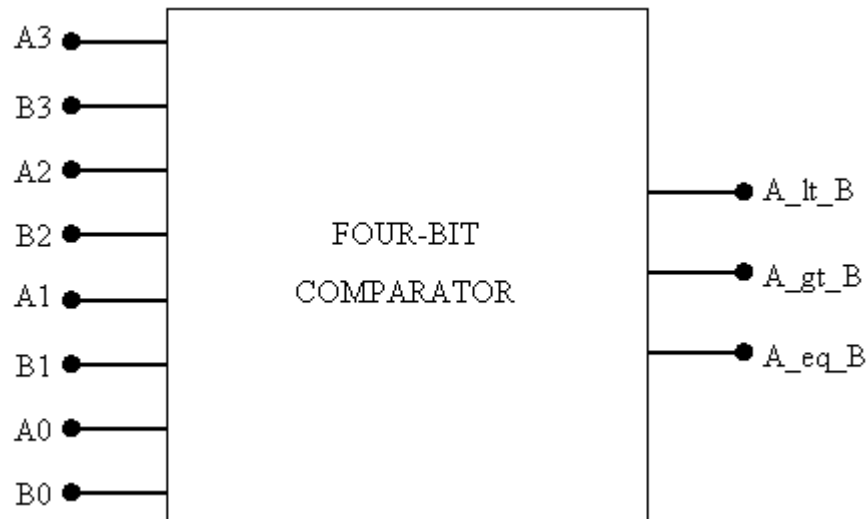
1. Write a Verilog module that models a 2-bit comparator and test your module using the testbench file Lab2_1_t.v. The I/O signals and behavior of the comparator are shown below.


$$\begin{cases} A_lt_B = 1, \text{ if } A1A0 < B1B0 \\ A_lt_B = 0, \text{ otherwise} \\ A_gt_B = 1, \text{ if } A1A0 > B1B0 \\ A_gt_B = 0, \text{ otherwise} \\ A_eq_B = 1, \text{ if } A1A0 = B1B0 \\ A_eq_B = 0, \text{ otherwise} \end{cases}$$

You have to use the following template for your design.

```
module Lab2_1(A_lt_B, A_gt_B, A_eq_B, A1, A0, B1, B0);  
  input A1, A0, B1, B0;  
  output A_lt_B, A_gt_B, A_eq_B;  
  // add your design here  
endmodule
```

- Write a Verilog module that models a 4-bit comparator. The 4-bit comparator must be implemented by using two 2-bit comparators (i.e., instances of module Lab2_1) and necessary logic gates. The I/O signals and behavior of the 4-bit comparator are shown below.



$$\begin{cases}
 A_lt_B = 1, & \text{if } A_3A_2A_1A_0 < B_3B_2B_1B_0 \\
 A_lt_B = 0, & \text{otherwise} \\
 A_gt_B = 1, & \text{if } A_3A_2A_1A_0 > B_3B_2B_1B_0 \\
 A_gt_B = 0, & \text{otherwise} \\
 A_eq_B = 1, & \text{if } A_3A_2A_1A_0 = B_3B_2B_1B_0 \\
 A_eq_B = 0, & \text{otherwise}
 \end{cases}$$

You have to use the following template for your design.

```

module Lab2_2(A_lt_B, A_gt_B, A_eq_B, A3, A2, A1, A0, B3,
B2, B1, B0);
input A3, A2, A1, A0, B3, B2, B1, B0;
output A_lt_B, A_gt_B, A_eq_B;
    // add your design here
endmodule

```

- Write a testbench to test your design, module Lab2_2. The testbench should generate all possible input combinations of the 4-bit comparator. One possible way to design the testbench is to modify Lab2_1_t.v.