Lab 2: 4-bit Comparator

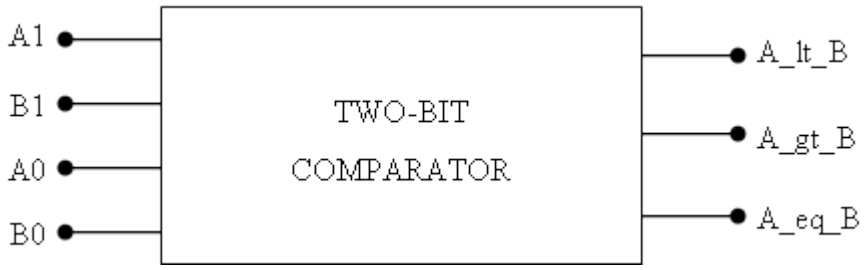
Due: March 12, 2015

Objective

- To be familiar with Verilog data flow modeling and testbench.

Action Items

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

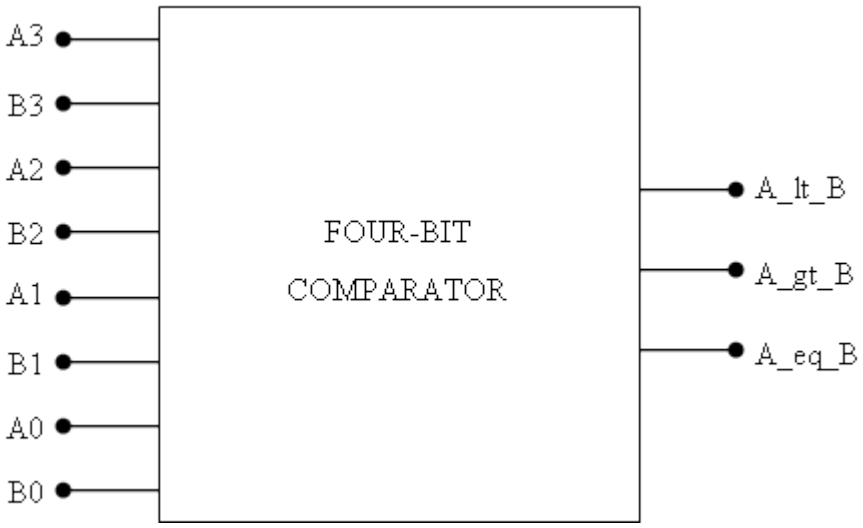


$$\begin{cases} A\_lt\_B = 1, \text{if } A1A0 < B1B0 \\ A\_lt\_B = 0, \text{otherwise} \end{cases}$$
$$\begin{cases} A\_gt\_B = 1, \text{if } A1A0 > B1B0 \\ A\_gt\_B = 0, \text{otherwise} \end{cases}$$
$$\begin{cases} A\_eq\_B = 1, \text{if } A1A0 = B1B0 \\ A\_eq\_B = 0, \text{otherwise} \end{cases}$$

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
```

2. Write a Verilog module that models a 4-bit comparator using continuous assignments. The 4-bit comparator is implemented by using two 2-bit comparators, i.e., module Lab2_1. The I/O signals and behavior of the 4-bit comparator are shown below.



$$\begin{cases} A\_lt\_B = 1, \text{if } A3A2A1A0 < B3B2B1B0 \\ A\_lt\_B = 0, \text{otherwise} \end{cases}$$
$$\begin{cases} A\_gt\_B = 1, \text{if } A3A2A1A0 > B3B2B1B0 \\ A\_gt\_B = 0, \text{otherwise} \end{cases}$$
$$\begin{cases} A\_eq\_B = 1, \text{if } A3A2A1A0 = B3B2B1B0 \\ A\_eq\_B = 0, \text{otherwise} \end{cases}$$

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
```

3. Write a testbench to test your design, module Lab2_2. The testbench should generate all possible input combinations of the 4-bit comparator.