4-BIT COMPARATOR

A=B

Recall the 1-bit comparator circuit we saw above. The equation for the A=B condition was $A \oplus B$. Let’s call this x. Taking a look at the truth table above, A=B is true only when (A3=B3 and A2=B2 and A1=B1 and A0=B0). Hence

$Z\ (A=B) = A3B3\ .\ A2B2\ .\ A1.B1\ .\ A0.B0 = x3x2x1x0$

A>B

Since there are multiple occasions where this particular condition is high, we will OR (add) each of those individual occasions. We find the first instance of A>B at the top of the table where A3>B3. For this to be possible in a binary system, A3 has to be equal to 1, and B3 has to be equal to 0. Since there are only 0s and 1s in a binary system. We can represent this as A3.B3′. Moving on to the next instance of A>B, we can see that it occurs at A3=B3 *and* A2>B2. From the equation for A=B above, A3=B3 can be represented as x3. And this entire instance can be written as x3A2B2′. Similarly, deriving equations for the remaining instances, we get the following equation

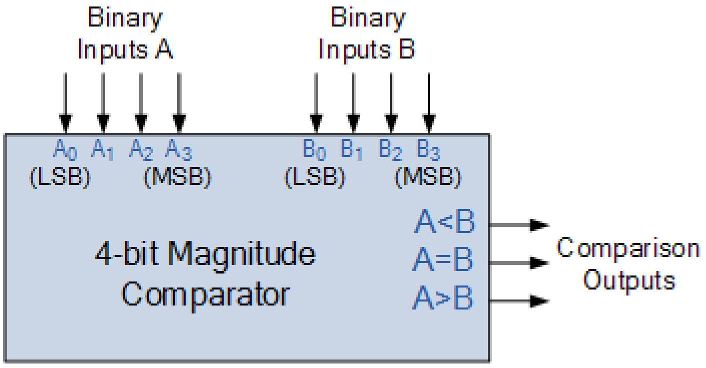
$X(A>B) = A3B3' + x3A2B2' + x3x2A1B1' + x3x2x1A0B0'$

A<B

Employing the same principles we used above, we get the following equation

$Y(A<B) = A3'B3 + X3A2'B2 + X3X2A1'B1 + X3X2X1A0'B0$

Designing the circuit based on the above equations, we get the following logic diagram for a 4-bit comparator.



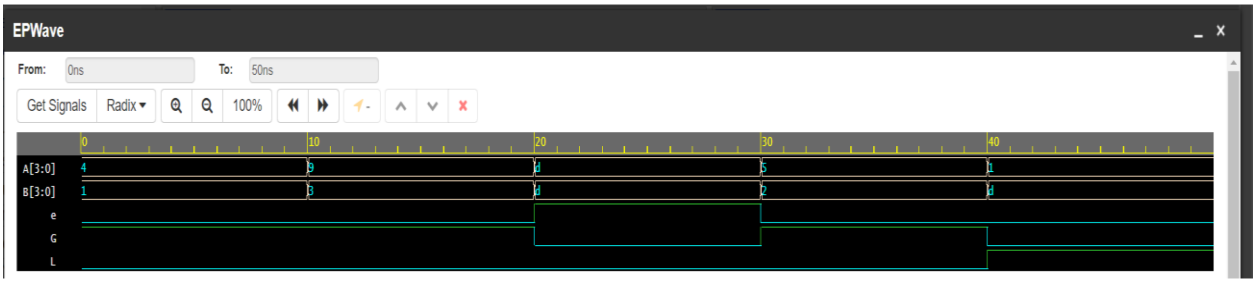
RTL CODE

```
module four_bit_comparator(input [3:0]A, [3:0]B,output G, e, L);
assign e = (A == B);
assign L = (A < B);
assign G = (A > B);
endmodule
```

TESTBENCH:

```
module testbench;
  reg [3:0]A;
  reg [3:0]B;
  wire G,e,L;
  four_bit_comparator C1 (A,B,G,e,L);
  initial
    begin
      $dumpfile(".vcd");
      $dumpvars(1);
    end
  initial
    begin
      repeat(20)
        begin
          A=$random;B=$random();
          #10;
        end
    end
  initial
    begin
```

```
     #50 $finish();
   end
endmodule
```



From: 0ns To: 50ns