
Day 10 Topic: Verilog codes for COMPARATOR Using Cadence (Xcelium)

◆ What is a Comparator?

A Comparator is a digital circuit that compares two binary numbers (say A and B) and determines their relative magnitudes:

- $A = B$
- $A > B$
- $A < B$

It produces logic outputs based on the comparison.

◆ Types of Comparators

1. 1-bit Comparator

- Compares single-bit inputs A and B.
- Outputs:
 - $A_{gt}B = A \& \sim B$
 - $A_{lt}B = \sim A \& B$
 - $A_{eq}B = \sim(A \wedge B)$

2. 2-bit Comparator

- Compares 2-bit numbers (A_1A_0 and B_1B_0).
- Comparison is done bit by bit (starting from MSB).

3. n-bit Comparator

- Generalized comparator for n-bit numbers.
- Usually implemented using cascading of 1-bit comparators or directly using arithmetic (subtraction).

◆ Truth Table (1-bit Comparator)

A B A=B A>B A<B

0 0 1 0 0

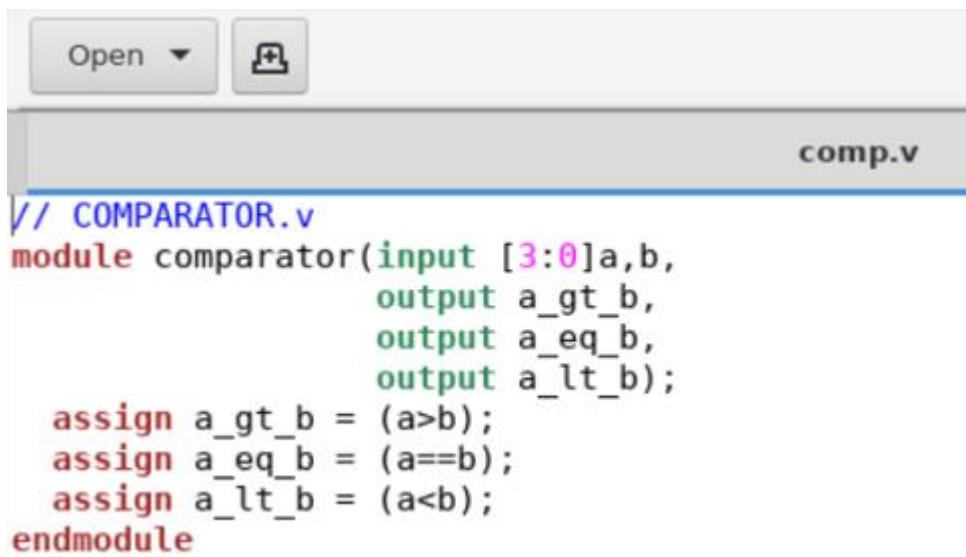
A B A=B A>B A<B

0 1 0 0 1

1 0 0 1 0

1 1 1 0 0

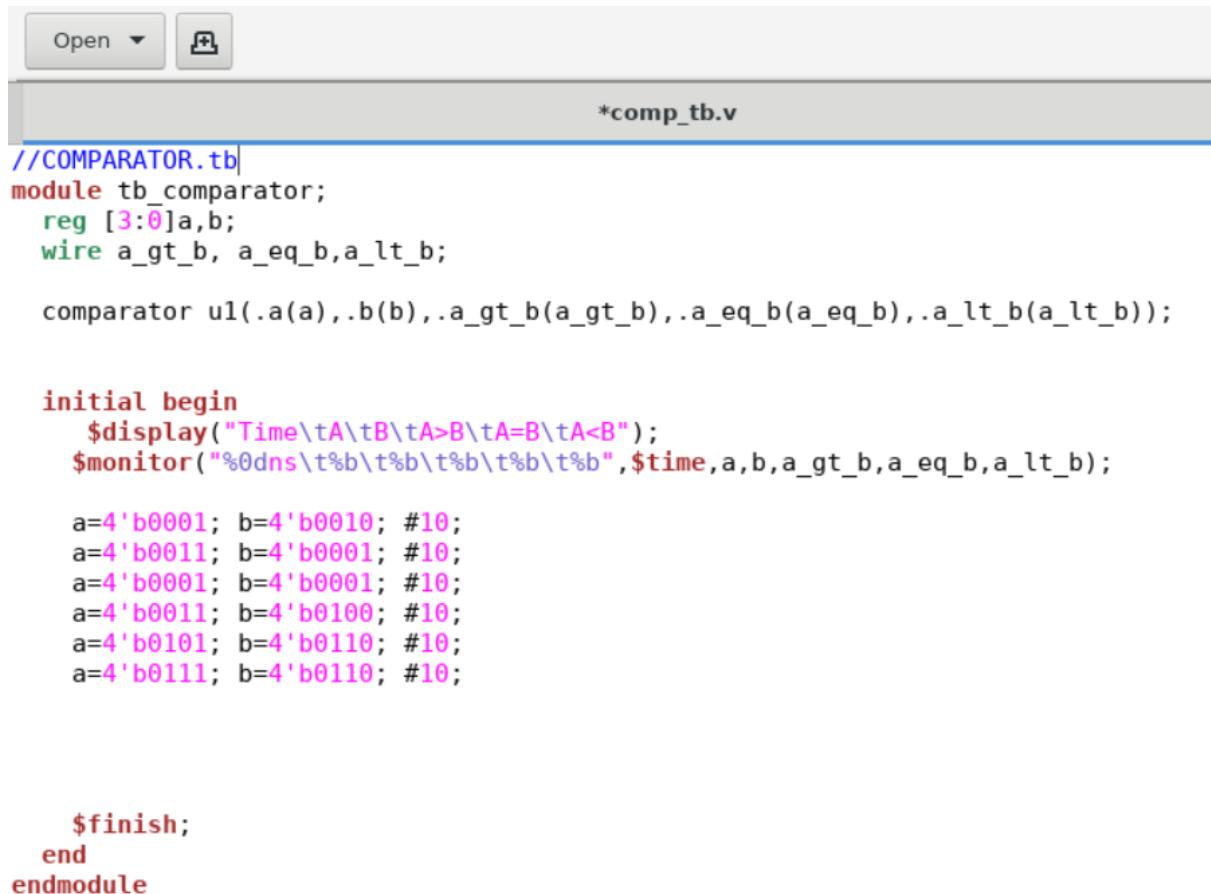
◆ 1.1) Design Code



The screenshot shows a text editor window with the file name "comp.v" at the top right. The code is a Verilog module named "comparator". It has three output ports: "a_gt_b", "a_eq_b", and "a_lt_b". The module takes two 4-bit inputs, "a" and "b", and uses assign statements to map the relational operators to the outputs.

```
// COMPARATOR.v
module comparator(input [3:0]a,b,
                     output a_gt_b,
                     output a_eq_b,
                     output a_lt_b);
    assign a_gt_b = (a>b);
    assign a_eq_b = (a==b);
    assign a_lt_b = (a<b);
endmodule
```

◆ 1.2) Test Bench Code



The screenshot shows a text editor window with the following content:

```
//COMPARATOR.tb
module tb_comparator;
reg [3:0]a,b;
wire a_gt_b, a_eq_b,a_lt_b;

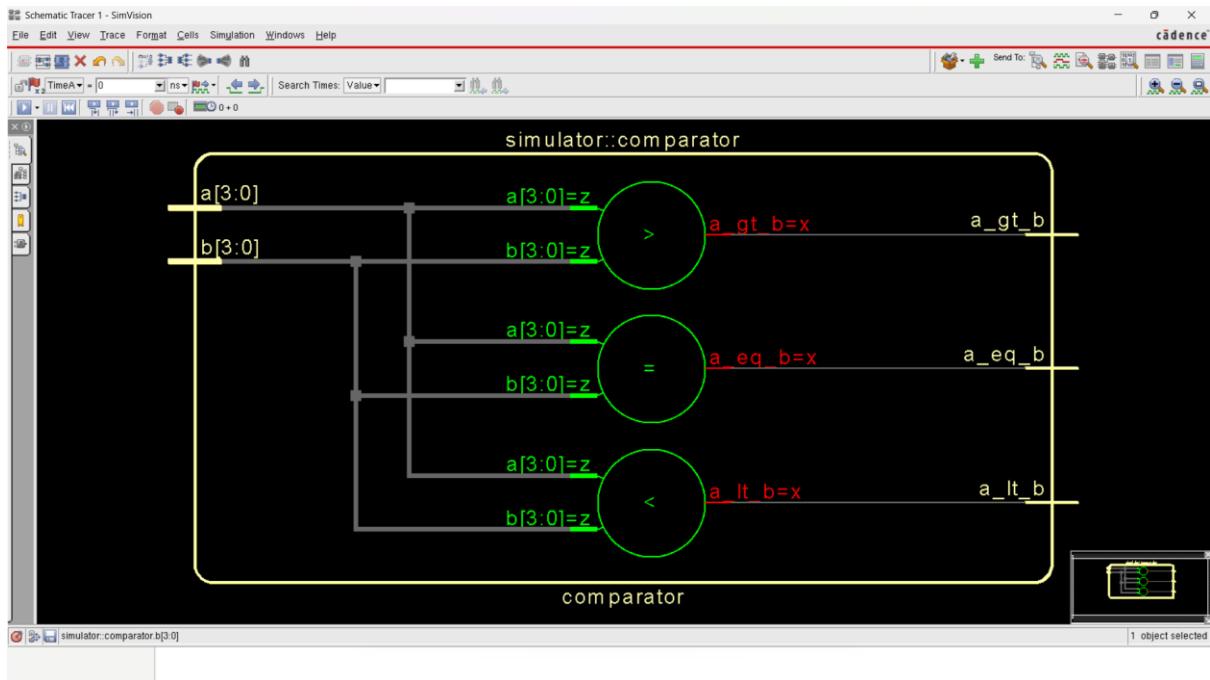
comparator u1(.a(a),.b(b),.a_gt_b(a_gt_b),.a_eq_b(a_eq_b),.a_lt_b(a_lt_b));

initial begin
$display("Time\tA\tB\tA>B\tA=B\tA<B");
$monitor("%0dns\t%b\t%b\t%b\t%b\t%b", $time,a,b,a_gt_b,a_eq_b,a_lt_b);

a=4'b0001; b=4'b0010; #10;
a=4'b0011; b=4'b0001; #10;
a=4'b0001; b=4'b0001; #10;
a=4'b0011; b=4'b0100; #10;
a=4'b0101; b=4'b0110; #10;
a=4'b0111; b=4'b0110; #10;

$finish;
end
endmodule
```

◆ 1.3) Schematic



◆ 1.4) Wave Forms

