

RTL CHALLENGE

Day-23:- Verilog Code for Implementation of One Bit Comparator

Software used:- Xilinx Vivado(2023.1)

Theory:-

One-bit comparator:-

A one-bit comparator is a digital circuit that compares two binary numbers consisting of a single bit each and determines their relationship, typically whether they are equal, greater than, or less than each other. The outputs of the comparator represent these relationships.

Here's a basic description of how a one-bit comparator works:

1. **Inputs:** The two binary numbers to be compared are inputted to the comparator. Let's call them A and B.
2. **Comparison:** The comparator compares the corresponding bits of A and B.
3. **Outputs:** Based on the comparison, the comparator produces three outputs:
 - A > B (Output "A>B")
 - A < B (Output "A<B")
 - A = B (Output "A=B")
4. **Logic Design:** The internal logic of the comparator consists of logic gates such as AND, OR, and NOT gates, which generate the outputs based on the comparison of the input bits.
5. **Truth Table:** The truth table for a one-bit comparator would list all possible combinations of A and B and the corresponding outputs for A>B, A<B, and A=B.

Here's a simple truth table for a one-bit comparator:

| A | B | A>B | A<B | A=B |
|---|---|-----|-----|-----|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |

In this truth table:

- If A > B, the output "A>B" is 1 and the other outputs are 0.
- If A < B, the output "A<B" is 1 and the other outputs are 0.
- If A = B, the output "A=B" is 1 and the other outputs are 0.

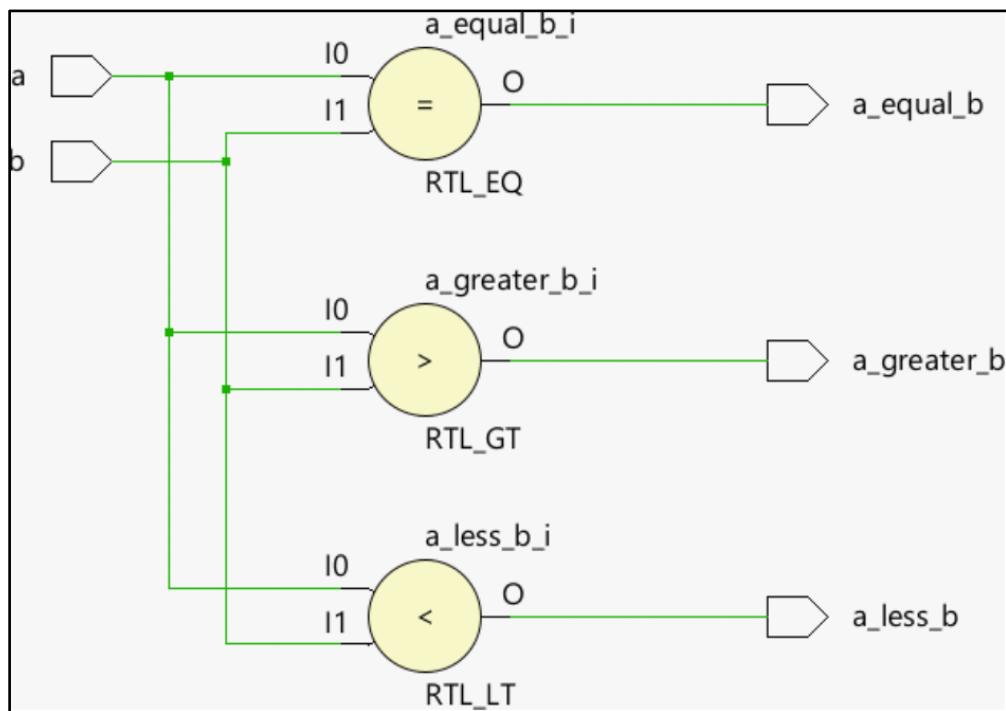
Design Code:-

```
module OneBitComparator(
    input a,b,
    output a_equal_b,a_less_b,a_greater_b
);
    assign a_equal_b = (a == b);
    assign a_less_b = (a<b);
    assign a_greater_b = (a>b);
endmodule
```

Test Bench:-

```
module OneBitComparator_tb;
reg a,b;
wire a_equal_b,a_less_b,a_greater_b;
OneBitComparator uut (.a(a),.b(b),.a_equal_b(a_equal_b),.a_less_b(a_less_b),.a_greater_b(a_greater_b));
initial
begin
    a=0;b=0:#100;
    a=0;b=1:#100;
    a=1;b=0:#100;
    a=1;b=1:#100;
end
endmodule
```

Schematic:-



Output Waveforms:-

