30 Days of RTL Coding

-By T Tushar Shenoy

Day 18

Problem Statement: Implementing Comparator in Behavioural style of Implementation.

Theory:

1-Bit Magnitude Comparator

A comparator used to compare two bits is called a single-bit comparator. It consists of two inputs each for two single-bit numbers and three outputs to generate less than, equal to, and greater than between two binary numbers.

Truth Table

A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

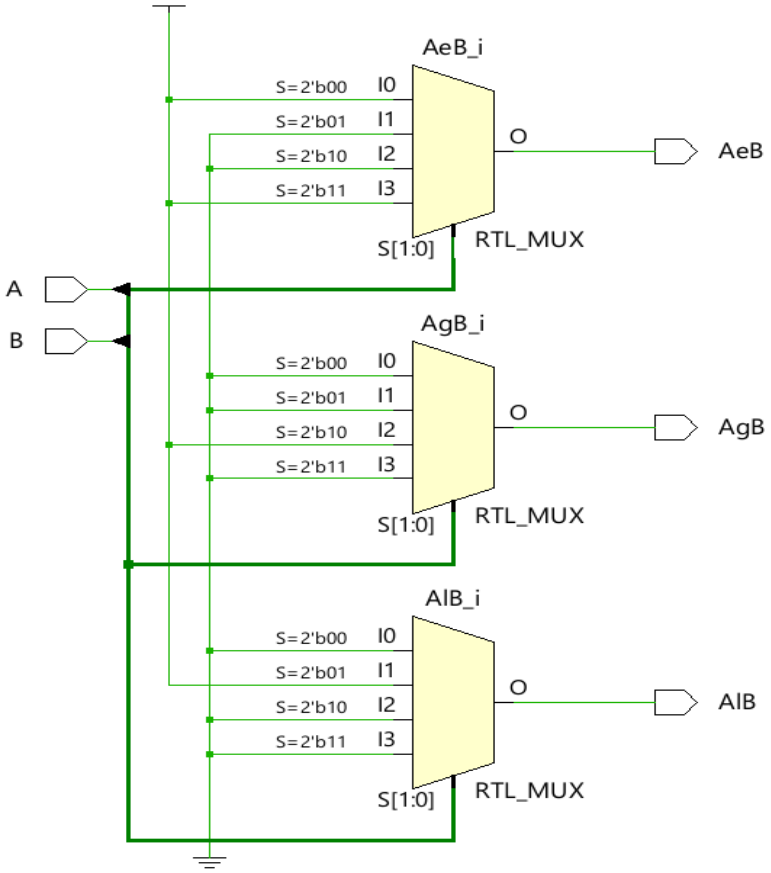
Verilog Code:

```
//Verilog Code for 1 Bit Comparator
module comparator1bit(A,B,AeB,AgB,AlB);

input A,B;
output reg AeB,AgB,AlB;

always@(A,B)
begin

AeB=1'b0;AgB=1'b0;AlB=1'b0;

case({A,B})
2'b00:AeB<=1'b1;
2'b01:AlB=1'b1;
2'b10:AgB=1'b1;
2'b11:AeB<=1'b1;
endcase

end

endmodule
```

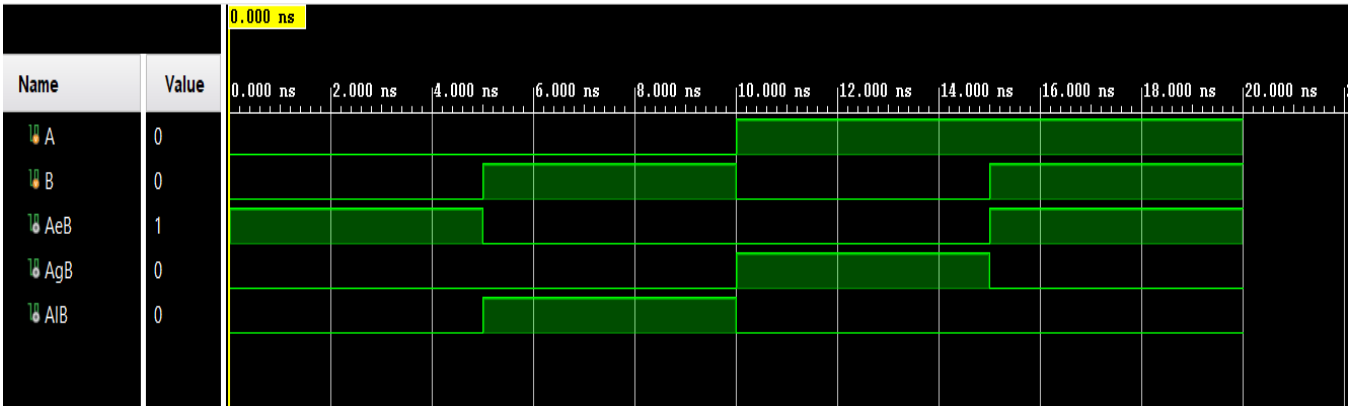
Testbench Code:

```
//Testbench code for 1 Bit Comparator
module comparator1bit_tb();

reg A,B;
wire AeB,AgB,AlB;

comparator1bit dut(.A(A),.B(B),.AeB(AeB),.AgB(AgB),.AlB(AlB));

initial begin
   A=1'b0;B=1'b0;
#5 A=1'b0;B=1'b1;
#5 A=1'b1;B=1'b0;
#5 A=1'b1;B=1'b1;
#5 $finish;
end

endmodule
```

Schematic:



Simulation Output:



2-Bit Magnitude Comparator

A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to, and greater than between two binary numbers.

Truth Table

Inputs				Outputs		
A		B				
$a1$	$a0$	$b1$	$b0$	$A = B$	$A > B$	$A < B$
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

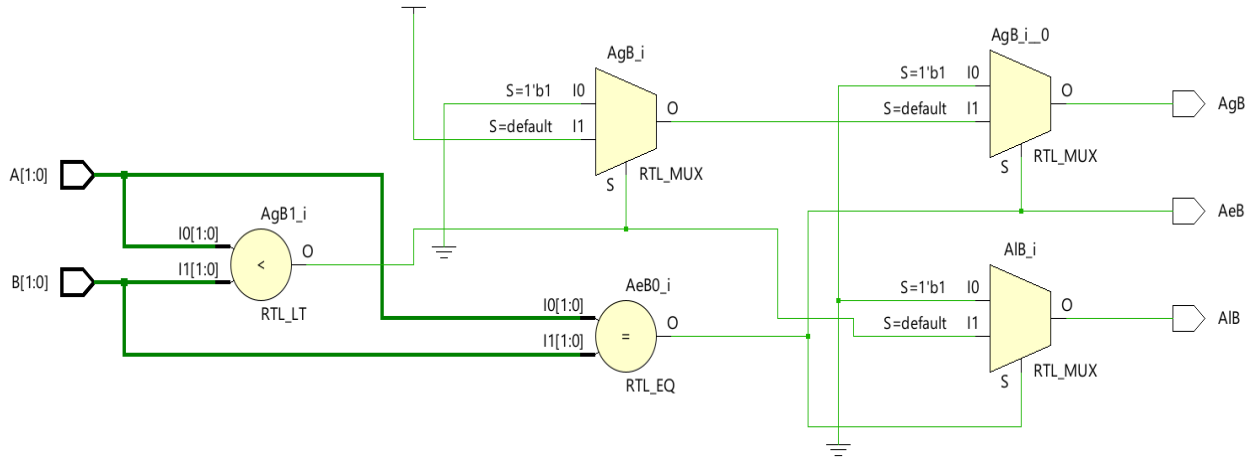
Verilog Code:

```
//Verilog Code for 2 Bit Comparator
module comparator2bit(A,B,AeB,AgB,AlB);

input [1:0]A,B;
output reg AeB,AgB,AlB;

always@(A,B)
begin
AeB=1'b0;AgB=1'b0;AlB=1'b0;
if(A==B)
  AeB=1'b1;
else if(A<B)
  AlB=1'b1;
else
  AgB=1'b1;
end

endmodule
```

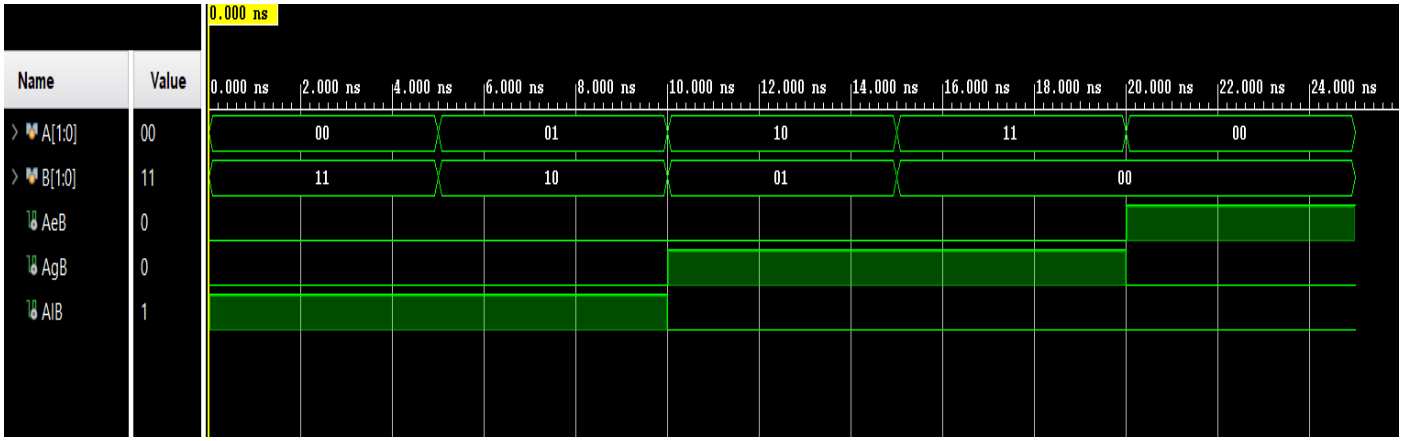
Testbench Code:

```
//Testbench code for 2 Bit Comparator
module comparator2bit_tb();

reg [1:0]A,B;
wire AeB,AgB,AlB;

comparator2bit dut(.A(A),.B(B),.AeB(AeB),.AgB(AgB),.AlB(AlB));

initial begin
   A=2'b00;B=2'b11;
#5 A=2'b01;B=2'b10;
#5 A=2'b10;B=2'b01;
#5 A=2'b11;B=2'b00;
#5 A=2'b00;B=2'b00;
//Add More Test Cases Here
#5 $finish;
end

endmodule
```

Schematic:



Simulation Output:



4-Bit Magnitude Comparator

A comparator used to compare two binary numbers each of four bits is called a 4-bit magnitude comparator. It consists of eight inputs each for two four-bit numbers and three outputs to generate less than, equal to, and greater than between two binary numbers.

Verilog Code:

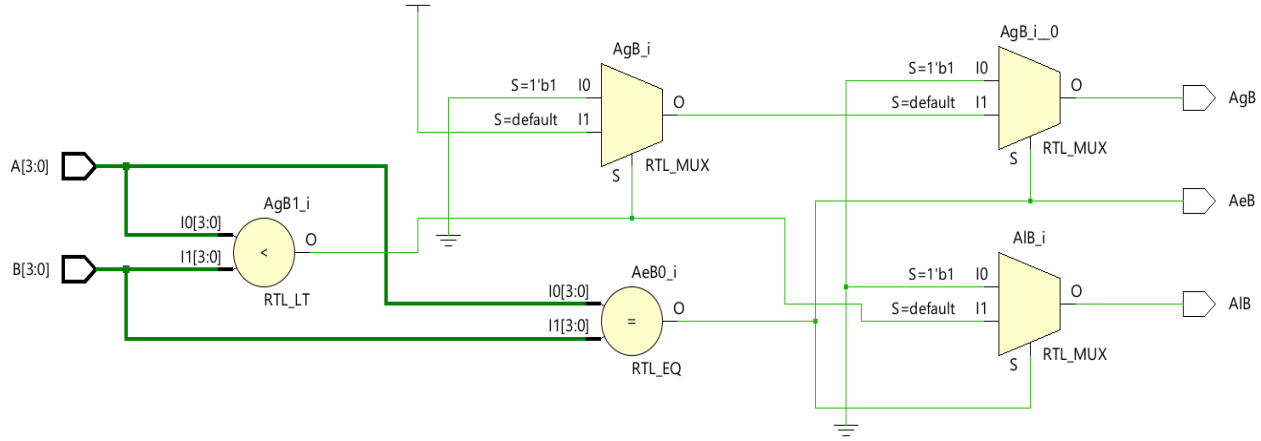
```
//Verilog Code for 4 Bit Comparator
module comparator4bit(A,B,AeB,AgB,AlB);

input [3:0]A,B;
output reg AeB,AgB,AlB;

always@(A,B)
begin
AeB=1'b0;AgB=1'b0;AlB=1'b0;
if(A==B)
  AeB=1'b1;
else if(A<B)
  AlB=1'b1;
else
  AgB=1'b1;
end

endmodule
```

Testbench Code:

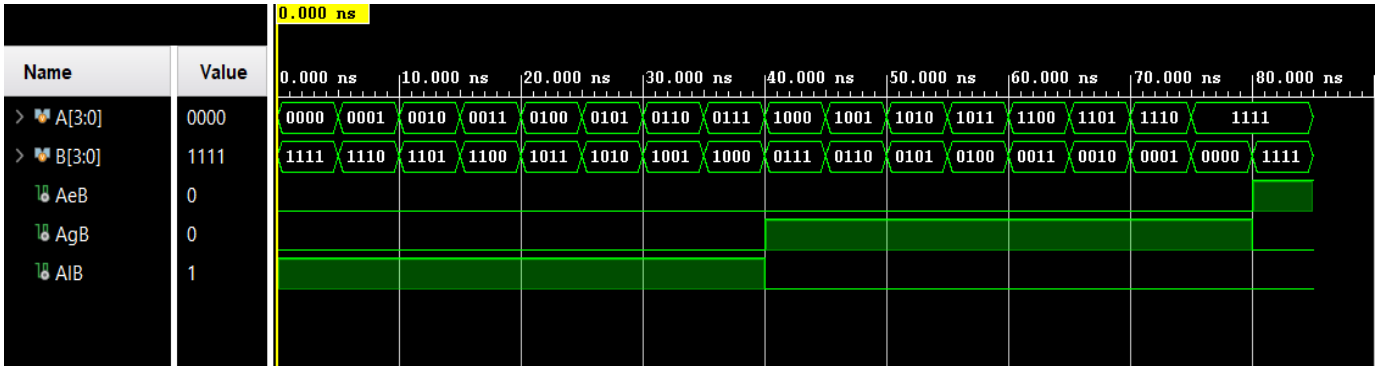
```
//Testbench code for 4 Bit Comparator
module comparator4bit_tb();

reg [3:0]A,B;
wire AeB,AgB,AlB;

comparator4bit dut(.A(A),.B(B),.AeB(AeB),.AgB(AgB),.AlB(AlB));

initial begin
   A=4'b0000;B=4'b1111;
#5 A=4'b0001;B=4'b1110;
#5 A=4'b0010;B=4'b1101;
#5 A=4'b0011;B=4'b1100;
#5 A=4'b0100;B=4'b1011;
#5 A=4'b0101;B=4'b1010;
#5 A=4'b0110;B=4'b1001;
#5 A=4'b0111;B=4'b1000;
#5 A=4'b1000;B=4'b0111;
#5 A=4'b1001;B=4'b0110;
#5 A=4'b1010;B=4'b0101;
#5 A=4'b1011;B=4'b0100;
#5 A=4'b1100;B=4'b0011;
#5 A=4'b1101;B=4'b0010;
#5 A=4'b1110;B=4'b0001;
#5 A=4'b1111;B=4'b0000;
#5 A=4'b1111;B=4'b1111;
```

```
//Add More Test Cases Here
#5 $finish;
end

endmodule
```

Schematic:



Simulation Output:



GitHub Repository URL: https://github.com/tusharshenoy/RTL-Day-18-Comparator-Behavioral-I