**Circuit Diagram & Truth Table:**

**Diagram

Description automatically generated**

**Methodology**

There are 13 Possible Cases:

1. A>B>C
2. A>C>B
3. A>B=C
4. B>A>C
5. B>C>A
6. B>A=C
7. C>A>B
8. C>B>A
9. C>B=A
10. B=C>A
11. A=C>B
12. B=A>C
13. A=B=C

**Code:**

**//DESIGN MODULE**

module comparator(Data\_in\_A,Data\_in\_B,Data\_in\_C,A\_Greater\_B\_Greater\_C,A\_Greater\_C\_Greater\_B,A\_Greater\_B\_Equal\_C,B\_Greater\_A\_Greater\_C,B\_Greater\_C\_Greater\_A,

B\_Greater\_A\_Equal\_C,C\_Greater\_A\_Greater\_B,C\_Greater\_B\_Greater\_A,C\_Greater\_A\_Equal\_B,A\_Equal\_B\_Greater\_C,A\_Equal\_C\_Greater\_B,B\_Equal\_C\_Greater\_A,A\_Equal\_B\_Equal\_C);

// input ports.

input [7:0] Data\_in\_A;

input [7:0] Data\_in\_B;

input [7:0] Data\_in\_C

// output ports.

output A\_Greater\_B\_Greater\_C;

output A\_Greater\_C\_Greater\_B;

output A\_Greater\_B\_Equal\_C;

output B\_Greater\_A\_Greater\_C;

output B\_Greater\_C\_Greater\_A;

output B\_Greater\_A\_Equal\_C;

output C\_Greater\_A\_Greater\_B;

output C\_Greater\_B\_Greater\_A;

output C\_Greater\_A\_Equal\_B;

output A\_Equal\_B\_Greater\_C;

output A\_Equal\_C\_Greater\_B;

output B\_Equal\_C\_Greater\_A;

output A\_Equal\_B\_Equal\_C;

//Internal variables

reg A\_Greater\_B\_Greater\_C;

reg A\_Greater\_C\_Greater\_B;

reg A\_Greater\_B\_Equal\_C;

reg B\_Greater\_A\_Greater\_C;

reg B\_Greater\_C\_Greater\_A;

reg B\_Greater\_A\_Equal\_C;

reg C\_Greater\_A\_Greater\_B;

reg C\_Greater\_B\_Greater\_A;

reg C\_Greater\_A\_Equal\_B;

reg A\_Equal\_B\_Greater\_C;

reg A\_Equal\_C\_Greater\_B;

reg B\_Equal\_C\_Greater\_A;

reg A\_Equal\_B\_Equal\_C;

//When the inputs and A or B or C are changed execute this block

always @(Data\_in\_A or Data\_in\_B or Data\_in\_C)

begin

if((Data\_in\_A>Data\_in\_B) & (Data\_in\_B>Data\_in\_C))

begin

A\_Greater\_B\_Greater\_C=1;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_A>Data\_in\_C) & (Data\_in\_C>Data\_in\_B))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=1;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_A>Data\_in\_B) & (Data\_in\_B == Data\_in\_C))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=1;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_B>Data\_in\_A) & (Data\_in\_A>Data\_in\_C))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=1;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_B>Data\_in\_C) & (Data\_in\_C>Data\_in\_A))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=1;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_B>Data\_in\_A) & (Data\_in\_A == Data\_in\_C))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=1;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_C>Data\_in\_A) & (Data\_in\_A>Data\_in\_B))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=1;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_C>Data\_in\_B) & (Data\_in\_B>Data\_in\_A))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=1;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_C>Data\_in\_A) & (Data\_in\_A == Data\_in\_B))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=1;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_A == Data\_in\_B) & (Data\_in\_A>Data\_in\_C))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=1;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_A == Data\_in\_C) & (Data\_in\_A>Data\_in\_B))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=1;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=0;

end

else if((Data\_in\_B == Data\_in\_C) & (Data\_in\_B>Data\_in\_A))

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=1;

A\_Equal\_B\_Equal\_C=0;

end

else

begin

A\_Greater\_B\_Greater\_C=0;

A\_Greater\_C\_Greater\_B=0;

A\_Greater\_B\_Equal\_C=0;

B\_Greater\_A\_Greater\_C=0;

B\_Greater\_C\_Greater\_A=0;

B\_Greater\_A\_Equal\_C=0;

C\_Greater\_A\_Greater\_B=0;

C\_Greater\_B\_Greater\_A=0;

C\_Greater\_A\_Equal\_B=0;

A\_Equal\_B\_Greater\_C=0;

A\_Equal\_C\_Greater\_B=0;

B\_Equal\_C\_Greater\_A=0;

A\_Equal\_B\_Equal\_C=1;

end

end

endmodule

**//TestBench Module**

module PROJECT;

// Inputs

reg [7:0] Data\_in\_A;

reg [7:0] Data\_in\_B;

reg [7:0] Data\_in\_C;

// Outputs

wire A\_Greater\_B\_Greater\_C;

wire A\_Greater\_C\_Greater\_B;

wire A\_Greater\_B\_Equal\_C;

wire B\_Greater\_A\_Greater\_C;

wire B\_Greater\_C\_Greater\_A;

wire B\_Greater\_A\_Equal\_C;

wire C\_Greater\_A\_Greater\_B;

wire C\_Greater\_B\_Greater\_A;

wire C\_Greater\_A\_Equal\_B;

wire A\_Equal\_B\_Greater\_C;

wire A\_Equal\_C\_Greater\_B;

wire B\_Equal\_C\_Greater\_A;

wire A\_Equal\_B\_Equal\_C;

comparator comp(Data\_in\_A,Data\_in\_B,Data\_in\_C,A\_Greater\_B\_Greater\_C,A\_Greater\_C\_Greater\_B,A\_Greater\_B\_Equal\_C,B\_Greater\_A\_Greater\_C,B\_Greater\_C\_Greater\_A,

B\_Greater\_A\_Equal\_C,C\_Greater\_A\_Greater\_B,C\_Greater\_B\_Greater\_A,C\_Greater\_A\_Equal\_B,A\_Equal\_B\_Greater\_C,A\_Equal\_C\_Greater\_B,B\_Equal\_C\_Greater\_A,A\_Equal\_B\_Equal\_C);

initial

begin

//Apply inputs

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd100;

Data\_in\_C = 8'd50;

#40;

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd50;

Data\_in\_C = 8'd100;

#40;

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd100;

Data\_in\_C = 8'd100;

#40;

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd200;

Data\_in\_C = 8'd50;

#40;

Data\_in\_A = 8'd50;

Data\_in\_B = 8'd200;

Data\_in\_C = 8'd150;

#40;

Data\_in\_A = 8'd50;

Data\_in\_B = 8'd100;

Data\_in\_C = 8'd50;

#40;

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd100;

Data\_in\_C = 8'd250;

#40;

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd200;

Data\_in\_C = 8'd250;

#40;

Data\_in\_A = 8'd100;

Data\_in\_B = 8'd100;

Data\_in\_C = 8'd150;

#40;

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd150;

Data\_in\_C = 8'd50;

#40;

Data\_in\_A = 8'd150;

Data\_in\_B = 8'd100;

Data\_in\_C = 8'd150;

#40;

Data\_in\_A = 8'd50;

Data\_in\_B = 8'd100;

Data\_in\_C = 8'd100;

#40;

Data\_in\_A = 8'd50;

Data\_in\_B = 8'd50;

Data\_in\_C = 8'd50;

#40;

end

endmodule

**Output:**

Graphical user interface, application

Description automatically generated