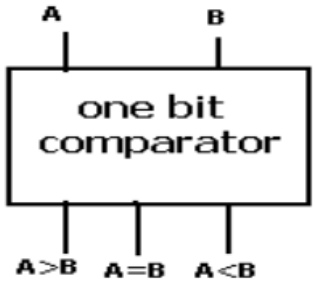
Experiment-3

Design Verilog HDL to implement simple circuits using structural, Data flow and Behavioral Model.

Aim: To design Verilog HDL to implement one-bit comparator using Data flow, behavioral and structural model.

Symbol



Truth table

Input		Output		
A	B	A==B	A>B	A<B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

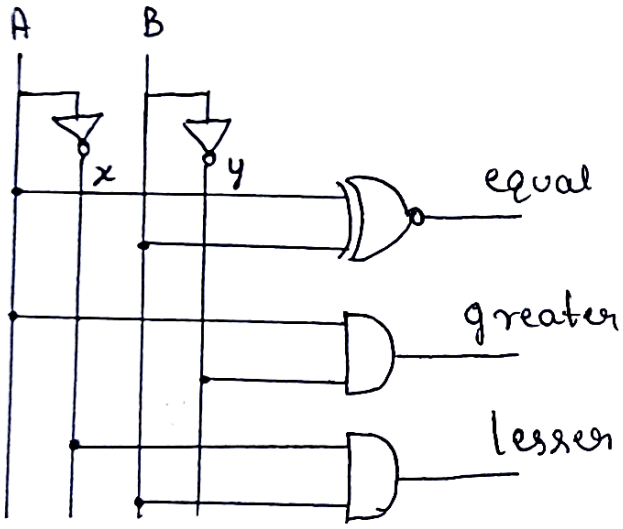
Expressions

Equal=A'B' + AB=A xnor B

Greater=AB’

Lesser=A’B

Logic Diagram



Verilog Code

Dataflow Description

```
module comparator_1 (a, b, equal, greater, lesser);
input a, b;
output equal, greater, lesser;
assign equal = ~(a^b);
assign greater = a&(~b);
assign lesser = (~a)&b;
endmodule
```

Behavioral Description

```
module comparator_2 (a, b, equal, greater, lesser);
input a, b;
output reg equal, greater, lesser;
always @ (a , b)
   begin
     if (a == b)
          begin
          equal = 1; greater = 0; lesser = 0;
          end
      else if (a > b)
          begin
          equal = 0; greater = 1; lesser = 0;
          end
       else
          begin
          equal = 0; greater = 0; lesser = 1;
          end
     end
endmodule
```

Structural Description

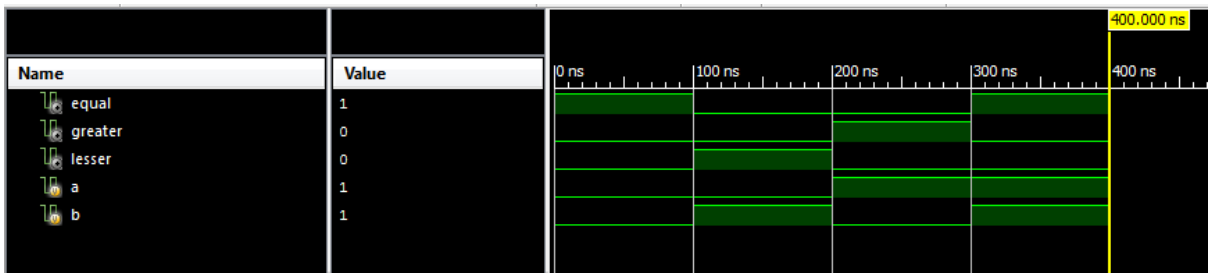
```
module comparator_3 (a, b, equal, greater, lesser);
input a, b;
output equal, greater, lower;
wire x, y;
not u1(x, a);
not u2(y, b);
xnor u4(equal, a, b);
and u3(greater, a, y);
and u5(lesser, x, b);
endmodule
```

Test bench code

```
module comparatortb;
reg a, b;
wire equal, greater, lesser;
```

```
comparator_1 UUT (.a(a), .b(b), .equal(equal), .greater(greater), .lesser(lesser));
initial begin
a=0; b=0; #100;
a=0; b=1; #100;
a=1; b=0; #100;
a=1; b=1; #100;
end
endmodule
```

Timing Diagram



Result

Verilog HDL code for one-bit comparator is designed and implemented using Data flow, behavioral and structural model.