4bit Comparator

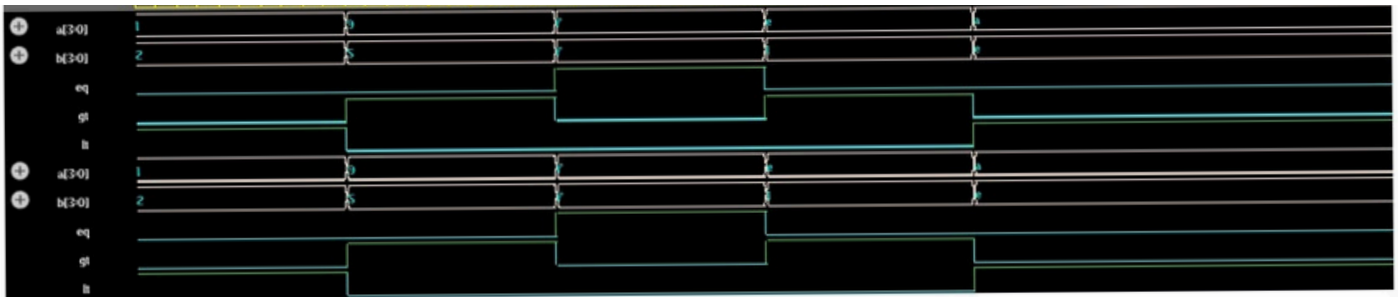
```verilog
// Code your testbench here
// or browse Examples
module tb_comparator_4bit;
  reg [3:0]a,b;
  wire lt,gt,eq;

  comparator_4bit uut(a,b,lt,gt,eq);

  initial begin
    $dumpfile("comparator.vcd");
    $dumpvars(0,tb_comparator_4bit);

    $monitor("Time=%0t | a=%b b=%b | lt=%b gt=%b eq=%b", $time,a,b,lt,gt,eq);

    a=4'b0001; b=4'b0010;#10;
    a=4'b1001; b=4'b0101;#10;
    a=4'b1111; b=4'b1111;#10;
a=4'b1110; b=4'b0001;#10;
a=4'b1010; b=4'b1110;#10;
#10;
    $finish;
  end
endmodule
```

```verilog
// Code your design here
module comparator_4bit(a, b, lt, gt, eq);
  input [3:0]a,b;
  output lt,gt,eq;

  assign lt= (a<b);
  assign gt= (a>b);
  assign eq= (a==b);
endmodule
```

Output:

Time=0 | a=0001 b=0010 | lt=1 gt=0 eq=0
Time=10 | a=1001 b=0101 | lt=0 gt=1 eq=0
Time=20 | a=1111 b=1111 | lt=0 gt=0 eq=1
Time=30 | a=1110 b=0001 | lt=0 gt=1 eq=0
Time=40 | a=1010 b=1110 | lt=1 gt=0 eq=0

Waveform:



D Flip flop

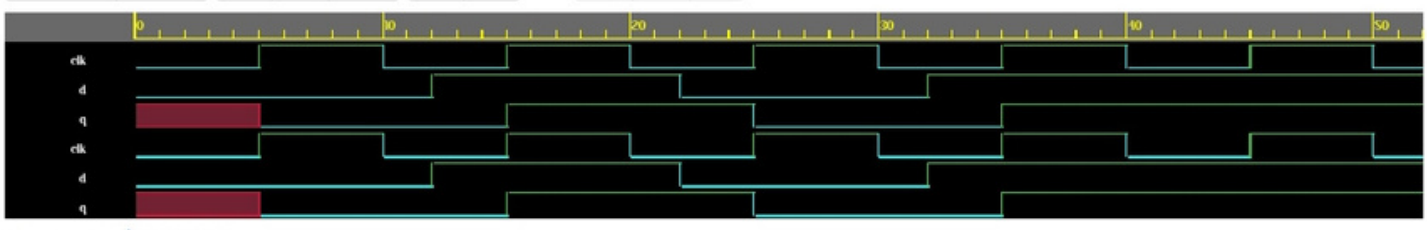
```
// Code your testbench here
// or browse Examples
module tb_d_ff;
  reg d,clk;
  wire q;

  d_ff uut(d,clk,q);

  initial begin
    $dumpfile("dff.vcd");
    $dumpvars(0,tb_d_ff);
    $monitor("Time=%0t | d=%b clk=%b
q=%b", $time,d,clk,q);

    clk = 0;forever #5 clk =~clk;
  end

  initial begin
    d=0; #12;
d=1; #10;
 d=0; #10;
d=1; #10;
    #10 $finish;
  end
endmodule
```

```
// Code your design here
module d_ff(input d,clk, output reg q);
  always @(posedge clk)
    q <=d;
endmodule
```

Output:

Time=0 | d=0 clk=0 q=x
Time=5 | d=0 clk=1 q=0
Time=10 | d=0 clk=0 q=0
Time=12 | d=1 clk=0 q=0
Time=15 | d=1 clk=1 q=1
Time=20 | d=1 clk=0 q=1
Time=22 | d=0 clk=0 q=1
Time=25 | d=0 clk=1 q=0
Time=30 | d=0 clk=0 q=0
Time=32 | d=1 clk=0 q=0
Time=35 | d=1 clk=1 q=1
Time=40 | d=1 clk=0 q=1
Time=45 | d=1 clk=1 q=1
Time=50 | d=1 clk=0 q=1

Waveform:



T flip flop

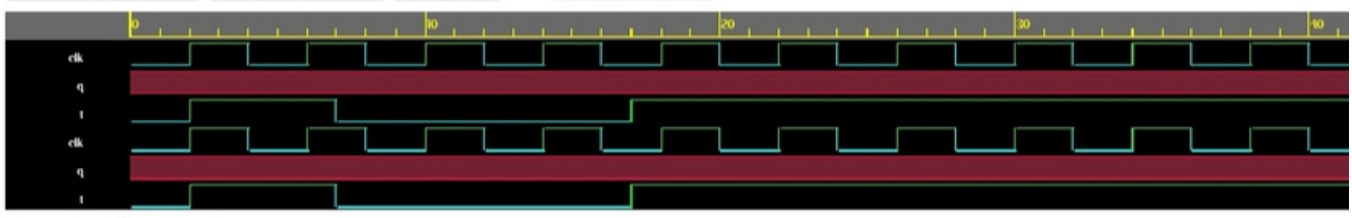
```
// or browse Examples

module tb_t_ff;
  reg t,clk;
  wire q;

  t_ff uut(t,clk, q);

  initial begin
    $dumpfile("tff.vcd");
    $dumpvars(0,tb_t_ff);
    $monitor("Time=%0t | t=%b clk=%b
q=%b", $time,t,clk,q);

    clk = 0; forever #2 clk =~clk;
  end

  initial begin
    t=0; #2;
t=1;#5;
t=0; #10;
t=1; #15;
    #10 $finish;
  end
endmodule
```

```
// Code your design here
module t_ff(input t,clk,output reg q);
  always @(posedge clk)
    if (t)q <=~q;
endmodule
```

Output:

Time=0 | t=0 clk=0 q=x
Time=2 | t=1 clk=1 q=x
Time=4 | t=1 clk=0 q=x
Time=6 | t=1 clk=1 q=x
Time=7 | t=0 clk=1 q=x
Time=8 | t=0 clk=0 q=x
Time=10 | t=0 clk=1 q=x
Time=12 | t=0 clk=0 q=x
Time=14 | t=0 clk=1 q=x
Time=16 | t=0 clk=0 q=x
Time=17 | t=1 clk=0 q=x
Time=18 | t=1 clk=1 q=x
Time=20 | t=1 clk=0 q=x

Waveform:



SR flip flop

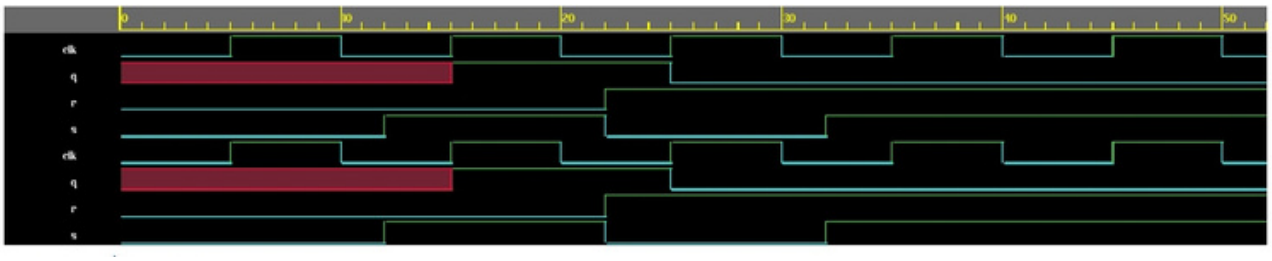
```
// or browse Examples

module tb_sr_ff;
  reg s,r,clk;
  wire q;

  sr_ff uut(s,r,clk, q);

  initial begin
    $dumpfile("srff.vcd");
    $dumpvars(0,tb_sr_ff);
    $monitor("Time=%0t | s=%b r=%b clk=%b q=%b", $time,s,r,clk, q);

    clk=0; forever #5 clk=~clk;
  end

  initial begin
    s=0;r=0;#12;
    s=1;r=0;#10;
    s=0;r=1;#10;
    s=1;r=1;#10;  // invalid
    #10 $finish;
  end
endmodule
```

```
// Code your design here
module sr_ff(input s, r,input clk, output reg q);
  always @(posedge clk) begin
    if(s & ~r) q<=1;
    else if(~s & r) q<=0;
  end
endmodule
```

Output:

Time=0 | s=0 r=0 clk=0 q=x
Time=5 | s=0 r=0 clk=1 q=x
Time=10 | s=0 r=0 clk=0 q=x
Time=12 | s=1 r=0 clk=0 q=x
Time=15 | s=1 r=0 clk=1 q=1
Time=20 | s=1 r=0 clk=0 q=1
Time=22 | s=0 r=1 clk=0 q=1
Time=25 | s=0 r=1 clk=1 q=0
Time=30 | s=0 r=1 clk=0 q=0
Time=32 | s=1 r=1 clk=0 q=0
Time=35 | s=1 r=1 clk=1 q=0
Time=40 | s=1 r=1 clk=0 q=0
Time=45 | s=1 r=1 clk=1 q=0
Time=50 | s=1 r=1 clk=0 q=0

Waveform:



JK flip flop

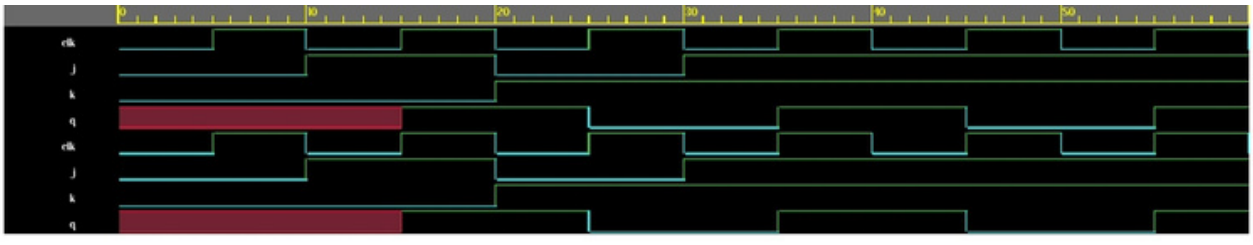
```
// Code your testbench here
// or browse Examples

module tb_jk_ff;
  reg j,k,clk;
  wire q;

  jk_ff uut(j,k,clk, q);

  initial begin
    $dumpfile("jkff.vcd");
    $dumpvars(0,tb_jk_ff);
    $monitor("Time=%0t | j=%b k=%b clk=%b q=%b", $time ,j,k,clk,q);

    clk=0; forever #5 clk=~clk;
  end

  initial begin
    j=0;k=0;#10;
    j=1;k=0;#10;
    j=0;k=1;#10;
    j=1;k=1;#20;

    #10 $finish;
  end
endmodule

```

```
// Code your design here
module jk_ff(input j,k, clk, output reg q);
  always @(posedge clk)begin
    if(j &~k) q <= 1;
    else if(~j &k)q <=0;
    else if(j &k)q <=~q;
  end
endmodule

```

Output:

Time=0 | j=0 k=0 clk=0 q=x
Time=5 | j=0 k=0 clk=1 q=x
Time=10 | j=1 k=0 clk=0 q=x
Time=15 | j=1 k=0 clk=1 q=1
Time=20 | j=0 k=1 clk=0 q=1
Time=25 | j=0 k=1 clk=1 q=0
Time=30 | j=1 k=1 clk=0 q=0
Time=35 | j=1 k=1 clk=1 q=1
Time=40 | j=1 k=1 clk=0 q=1
Time=45 | j=1 k=1 clk=1 q=0
Time=50 | j=1 k=1 clk=0 q=0
Time=55 | j=1 k=1 clk=1 q=1

Waveform:



D latch

```
// Code your testbench here
// or browse Examples
module tb_d_latch;
  reg d,en;
  wire q;

  d_latch uut(d,en ,q);

  initial begin
    $dumpfile("dlatch.vcd");
    $dumpvars(0,tb_d_latch);
    $monitor("Time=%0t | d=%b en=%b q=%b",
$time, d,en,q);

    d=0; en=0;#10;
    d=1; en=1;#10;
    d=0; en=1;#10;
    d=1; en=0;#10;
    #10 $finish;
  end
endmodule
```

```
// Code your design here
module d_latch(input d,en,output reg q);
  always @(*) begin
    if(en)q=d;
  end
endmodule
```

Output:

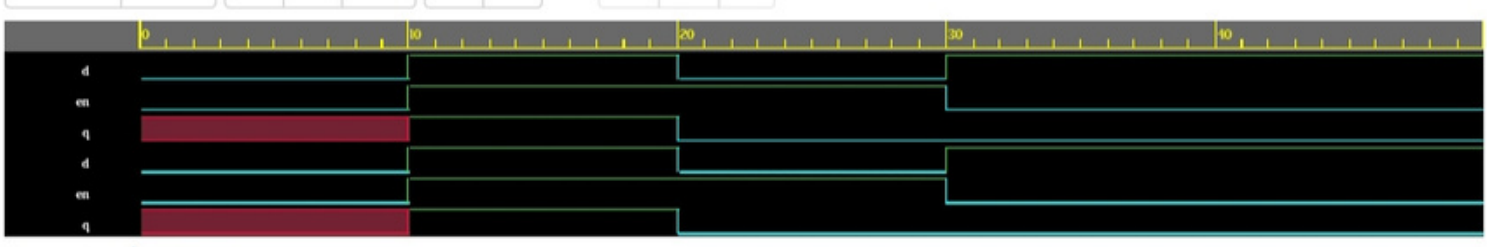
Time=0 | d=0 en=0 q=x

Time=10 | d=1 en=1 q=1

Time=20 | d=0 en=1 q=0

Time=30 | d=1 en=0 q=0

Waveform:



JK latch

```
// Code your testbench here
// or browse Examples
module tb_jk_latch;
  reg j,k,en;
  wire q;

  jk_latch uut(j,k,en,q);

  initial begin
    $dumpfile("jklatch.vcd");
    $dumpvars(0,tb_jk_latch);
    $monitor("Time=%0t | j=%b k=%b en=%b q=%b", $time, j,k,en, q);

    j=0;k=0;en=0;#10;
    j=1;k=0;en=1;#10;
    j=0;k=1;en=1;#10;
    j=1;k=1;en=1;#20;
    #10 $finish;
  end
endmodule
```

```
// Code your design here
module jk_latch(input j, k, en,output reg q);
  always @(*) begin
    if(en) begin
      if(j & ~k) q=1;
      else if(~j & k)q=0;
      else if(j & k)q=~q;
    end
  end
endmodule
```

Output:

Time=0 | j=0 k=0 en=0 q=x
Time=10 | j=1 k=0 en=1 q=1
Time=20 | j=0 k=1 en=1 q=0
Time=30 | j=1 k=1 en=1 q=1

Waveform: