LAB 4 – Raja Aadhithan

Design – D Flip Flop:

Code:

```
module dfflop(input clock,reset,d_in, output Q_out,Qb_out);
reg x;
   always@(posedge clock) begin
    if(reset) x <= 1'b0;
    else x <= d_in;
   end
   assign Q_out = x;
   assign Qb_out = !x;
endmodule
```

Testbench:

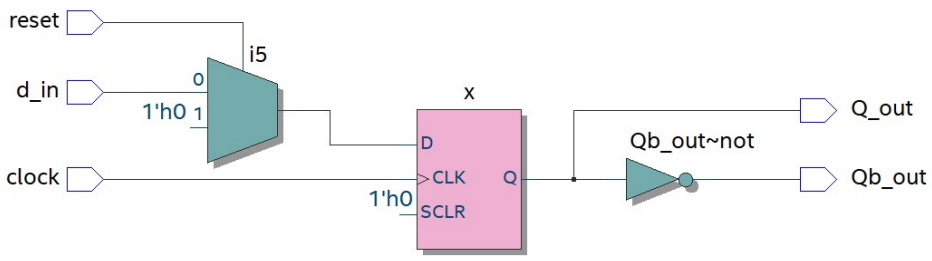
```
module count_tb();
reg clk,reset,load;
reg [3:0]i;
wire [3:0]q;
count4 dut(clk,reset,load,i,q);
initial begin
    clk = 1;
    forever #5 clk = ~clk;
end
initial begin
    $monitor("@time : %3d - for load = %b output is %b",$time,load,q);
    reset <= 1;
    #15;
    reset <= 0;
    load <= 1;
    i <= 4'b1101;
    #15;
    load <=0;
    #40;
    reset <= 1;
    #15;
    reset <= 0;
    #300;
    $finish;
end
endmodule
```

Wave:

Output:

```
VSIM 17> run -all
# @time:   0ps the input: x, reset:x, output is x,x
# @time:   5ps the input: x, reset:1, output is x,x
# @time:  10ps the input: x, reset:1, output is 0,1
# @time:  15ps the input: x, reset:0, output is 0,1
# @time:  20ps the input: x, reset:0, output is x,x
# @time:  25ps the input: 0, reset:0, output is x,x
# @time:  30ps the input: 0, reset:0, output is 0,1
# @time:  35ps the input: 1, reset:0, output is 0,1
# @time:  40ps the input: 1, reset:0, output is 1,0
# @time:  45ps the input: 0, reset:0, output is 1,0
# @time:  50ps the input: 0, reset:0, output is 0,1
# @time:  55ps the input: 1, reset:0, output is 0,1
# @time:  60ps the input: 1, reset:0, output is 1,0
# @time:  75ps the input: 1, reset:1, output is 1,0
# @time:  80ps the input: 1, reset:1, output is 0,1
# @time:  85ps the input: 1, reset:0, output is 0,1
# @time:  90ps the input: 1, reset:0, output is 1,0
# @time:  95ps the input: 0, reset:0, output is 1,0
# @time: 100ps the input: 0, reset:0, output is 0,1
# @time: 105ps the input: 1, reset:0, output is 0,1
# @time: 110ps the input: 1, reset:0, output is 1,0
# ** Note: $finish    : C:/Users/Aadhithan/Documents
#    Time: 115 ps  Iteration: 0  Instance: /dff_tb
```

RTL:



Exercises:

Design : SR latch:

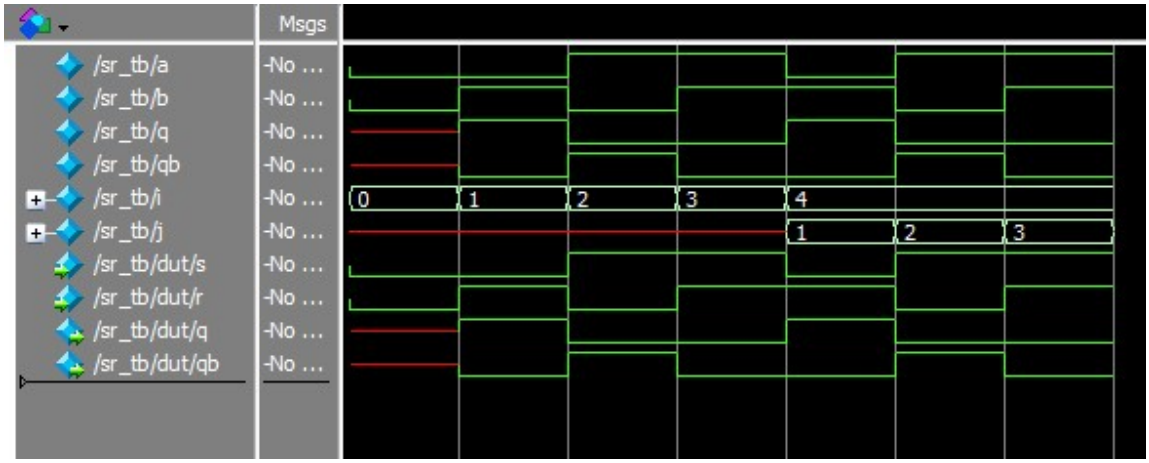
Code:

```
module sr(input s,r,output q,qb);
nor(q,s,qb);
nor(qb,r,q);
endmodule
```

Testbench:

```
module sr_tb();
reg a,b;
wire q,qb;
integer i,j;
sr dut(a,b,q,qb);
initial begin
    $monitor("@time %2d input is %b,%b  - output is %b,%b",$time,a,b,q,qb);
    for (i=0;i<4;i = i+1)
    begin
        {a,b}=i;
        #10;
    end
    for (j=1;j<4;j = j+1)
    begin
        {a,b}=j;
        #10;
    end
    $finish;
end
endmodule
```

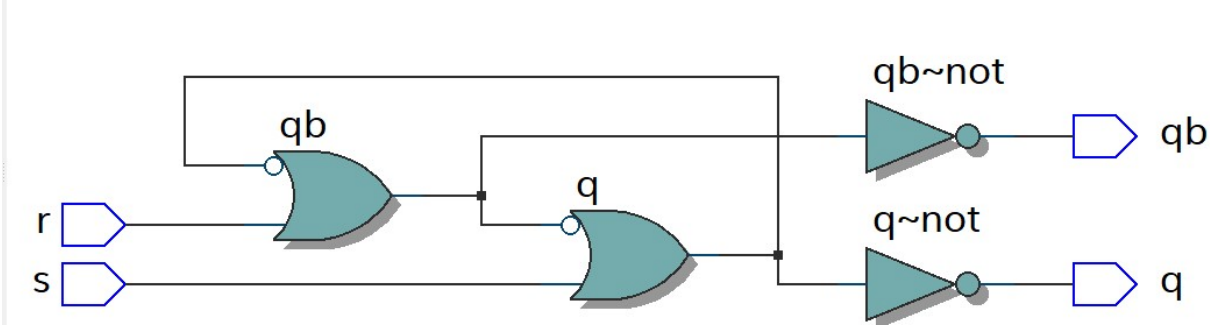
Wave:



Output:

```
VSIM 24> run -all
# @time  0 input is 0,0  - output is x,x
# @time 10 input is 0,1  - output is 1,0
# @time 20 input is 1,0  - output is 0,1
# @time 30 input is 1,1  - output is 0,0
# @time 40 input is 0,1  - output is 1,0
# @time 50 input is 1,0  - output is 0,1
# @time 60 input is 1,1  - output is 0,0
# ** Note: $finish    : C:/Users/Aadhithar
#    Time: 70 ps  Iteration: 0  Instance:
# 1
```

RTL:



Design: JK Flip Flop:

Code:

```
module jkfflop(input j,k,clk,output reg q,wire qb);
parameter HOLD=2'b00, RESET=2'b01, SET=2'b10, TOGGLE=2'b11;
always@(posedge clk)begin
    case({j,k})
    RESET : q <= 0;
    SET   : q <= 1;
    TOGGLE: q <= qb;
    endcase
end
assign qb = ~q;
endmodule
```

Testbench:

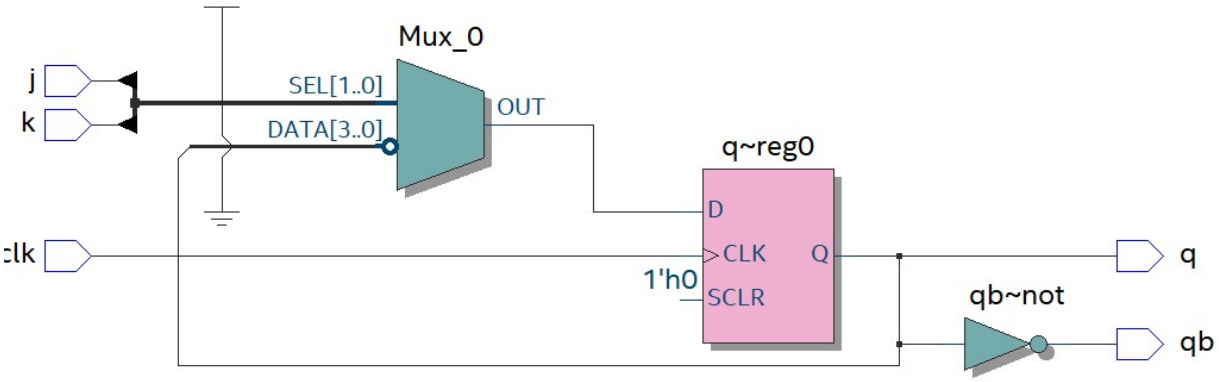
```
module jk_tb();
reg j,k,clk;
wire q,qb;
integer i;
jkfflop dut(j,k,clk,q,qb);
initial begin
    clk = 1'b1;
    forever #5 clk = ~clk;
end
initial begin
    $monitor("@time : %3d: j is %b, k is %b, output is %b-%b",$time,j,k,q,qb);
    for(i=0;i<8;i=i+1)
    begin
        {j,k}=i[1:0];
        #10;
    end
    $finish;
end
endmodule
```

Wave:

Output:

```
VSIM 28> run -all
# @time :   0: j is 0, k is 0, output is x-x
# @time :  10: j is 0, k is 1, output is 0-1
# @time :  20: j is 1, k is 0, output is 1-0
# @time :  30: j is 1, k is 1, output is 0-1
# @time :  40: j is 0, k is 0, output is 0-1
# @time :  50: j is 0, k is 1, output is 0-1
# @time :  60: j is 1, k is 0, output is 1-0
# @time :  70: j is 1, k is 1, output is 0-1
# ** Note: $finish    : C:/Users/Aadhithan/Docume
b.v(17)
#    Time: 80 ps  Iteration: 0  Instance: /jk_tb
```

RTL:



Design : T Flip Flop:

Code:

```
module tfflop(input t,clk,reset,output reg q, wire qb);
wire d;
assign d = t^q;
assign qb = !q;
always@(posedge clk)begin
    if (reset) q<=0;
    else q<=d;
end
endmodule
```

Testbench:

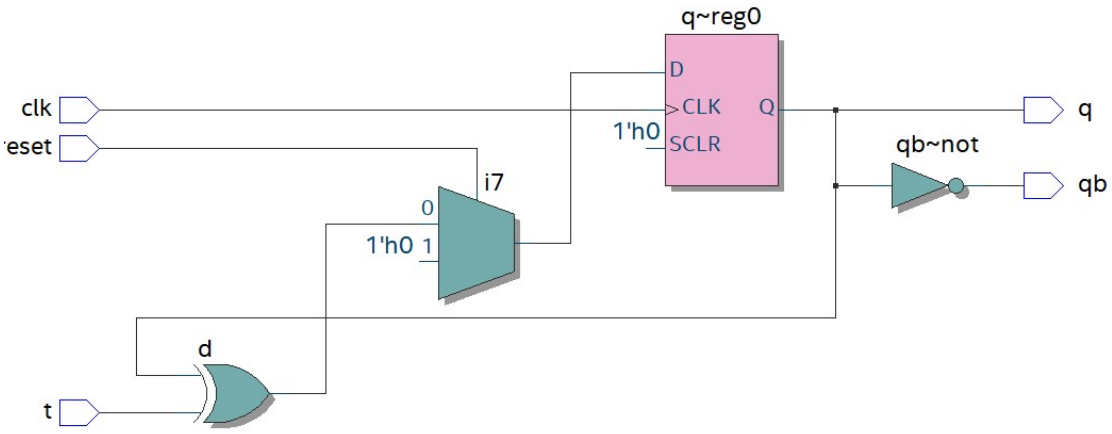
```
module tff_tb();
reg t,clk,reset;
wire q,qb;
integer i;
tfflop dut(t,clk,reset,q,qb);
initial begin
    clk = 1;
    forever #5 clk = ~clk;
end
initial begin
    $monitor("@time = %3d: T is %b and outputs are %b - %b",$time,t,q,qb);
    reset = 1'b1;
    #5;
    reset = 1'b0;
    for(i=0;i<8;i=i+1)
    begin
        t = i[0];
        #10;
    end
    $finish;
end
endmodule
```

Wave:

Output:

```
VSIM 32> run -all
# @time =    0: T is x and outputs are 0 - 1
# @time =    5: T is 0 and outputs are 0 - 1
# @time =   15: T is 1 and outputs are 0 - 1
# @time =   20: T is 1 and outputs are 1 - 0
# @time =   25: T is 0 and outputs are 1 - 0
# @time =   35: T is 1 and outputs are 1 - 0
# @time =   40: T is 1 and outputs are 0 - 1
# @time =   45: T is 0 and outputs are 0 - 1
# @time =   55: T is 1 and outputs are 0 - 1
# @time =   60: T is 1 and outputs are 1 - 0
# @time =   65: T is 0 and outputs are 1 - 0
# @time =   75: T is 1 and outputs are 1 - 0
# @time =   80: T is 1 and outputs are 0 - 1
# ** Note: $finish    : C:/Users/Aadhithan/Docume
_tb.v(20)
#    Time: 85 ps  Iteration: 0  Instance: /tff_tb
```

RTL:



Design: 4 bit synchronous up counter:

Code:

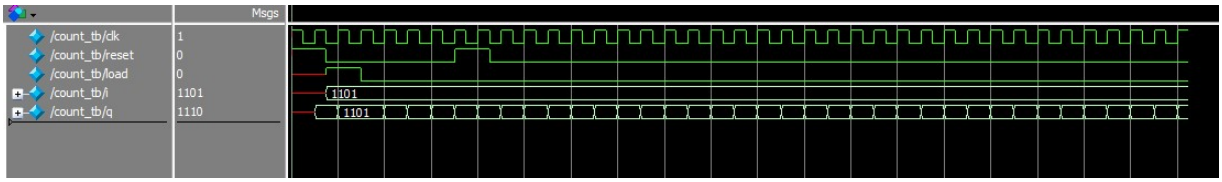
```
module count4(input clk,reset,load,[3:0]i,output reg [3:0]q);
always@(posedge clk)begin
    if(reset) q <= 4'd0;
    else if (load) q <= i;
    else q <= q+1;
end
endmodule
```

Testbench:

```
module count_tb();
reg clk,reset,load;
reg [3:0]i;
wire [3:0]q;
count4 dut(clk,reset,load,i,q);
initial begin
    clk = 1;
    forever #5 clk = ~clk;
end
initial begin
    $monitor("@time : %3d - for load = %b output is %b",$time,load,q);
    reset <= 1;
    #15;
    reset <= 0;
    load <= 1;
    i <= 4'b1101;
    #15;
    load <=0;
    #40;
    reset <= 1;
    #15;
    reset <= 0;
    #300;
    $finish;
end
endmodule
```

Wave:

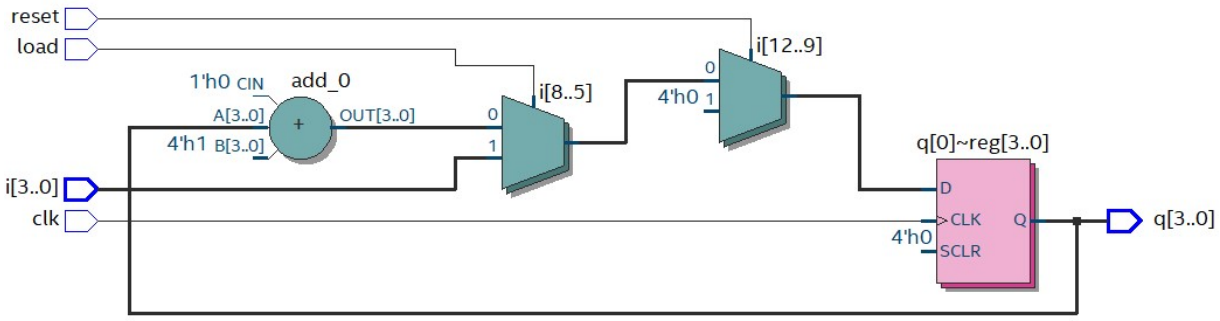
Zoomed out version:



Output:

```
VSIM 36> run -all
# @time :   0 - for load = x output is xxxx
# @time :  10 - for load = x output is 0000
# @time :  15 - for load = 1 output is 0000
# @time :  20 - for load = 1 output is 1101
# @time :  30 - for load = 0 output is 1101
# @time :  40 - for load = 0 output is 1110
# @time :  50 - for load = 0 output is 1111
# @time :  60 - for load = 0 output is 0000
# @time :  70 - for load = 0 output is 0001
# @time :  80 - for load = 0 output is 0000
# @time :  90 - for load = 0 output is 0001
# @time : 100 - for load = 0 output is 0010
# @time : 110 - for load = 0 output is 0011
# @time : 120 - for load = 0 output is 0100
# @time : 130 - for load = 0 output is 0101
# @time : 140 - for load = 0 output is 0110
# @time : 150 - for load = 0 output is 0111
# @time : 160 - for load = 0 output is 1000
# @time : 170 - for load = 0 output is 1001
# @time : 180 - for load = 0 output is 1010
# @time : 190 - for load = 0 output is 1011
# @time : 200 - for load = 0 output is 1100
# @time : 210 - for load = 0 output is 1101
# @time : 220 - for load = 0 output is 1110
# @time : 230 - for load = 0 output is 1111
# @time : 240 - for load = 0 output is 0000
# @time : 250 - for load = 0 output is 0001
# @time : 260 - for load = 0 output is 0010
# @time : 270 - for load = 0 output is 0011
# @time : 280 - for load = 0 output is 0100
# @time : 290 - for load = 0 output is 0101
# @time : 300 - for load = 0 output is 0110
# @time : 310 - for load = 0 output is 0111
# @time : 320 - for load = 0 output is 1000
# @time : 330 - for load = 0 output is 1001
# @time : 340 - for load = 0 output is 1010
# @time : 350 - for load = 0 output is 1011
# @time : 360 - for load = 0 output is 1100
# @time : 370 - for load = 0 output is 1101
# @time : 380 - for load = 0 output is 1110
# ** Note: $finish    : C:/Users/Aadhithan/Documents/
/count_tb.v(24)
#    Time: 385 ps  Iteration: 0  Instance: /count_tb
```

RTL:



Design: Mod 12 counter:

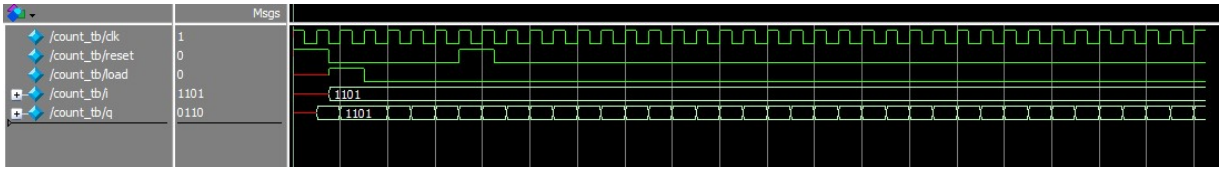
Code:

```
module count4(input clk,reset,load,[3:0]i,output reg [3:0]q);
always@(posedge clk)begin
    if(reset||q==4'b1011) q <= 4'd0;
    else if (load) q <= i;
    else q <= q+1;
end
endmodule
```

Testbench:

```
module count_tb();
reg clk,reset,load;
reg [3:0]i;
wire [3:0]q;
count4 dut(clk,reset,load,i,q);
initial begin
    clk = 1;
    forever #5 clk = ~clk;
end
initial begin
    $monitor("@time : %3d - for load = %b output is %b",$time,load,q);
    reset <= 1;
    #15;
    reset <= 0;
    load <= 1;
    i <= 4'b1101;
    #15;
    load <=0;
    #40;
    reset <= 1;
    #15;
    reset <= 0;
    #300;
    $finish;
end
endmodule
```

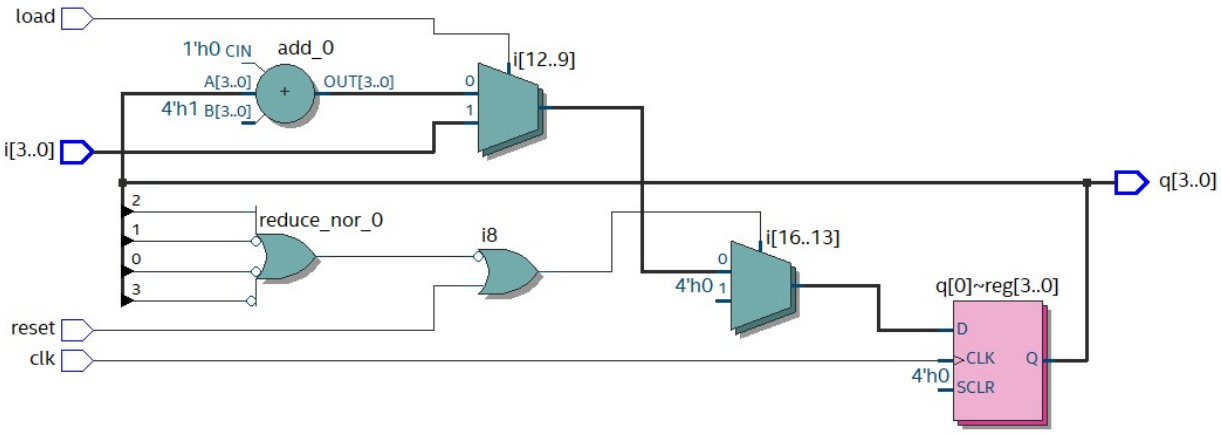
Wave:



Output:

```
VSIM 40> run -all
# @time :   0 - for load = x output is xxxx
# @time :  10 - for load = x output is 0000
# @time :  15 - for load = 1 output is 0000
# @time :  20 - for load = 1 output is 1101
# @time :  30 - for load = 0 output is 1101
# @time :  40 - for load = 0 output is 1110
# @time :  50 - for load = 0 output is 1111
# @time :  60 - for load = 0 output is 0000
# @time :  70 - for load = 0 output is 0001
# @time :  80 - for load = 0 output is 0000
# @time :  90 - for load = 0 output is 0001
# @time : 100 - for load = 0 output is 0010
# @time : 110 - for load = 0 output is 0011
# @time : 120 - for load = 0 output is 0100
# @time : 130 - for load = 0 output is 0101
# @time : 140 - for load = 0 output is 0110
# @time : 150 - for load = 0 output is 0111
# @time : 160 - for load = 0 output is 1000
# @time : 170 - for load = 0 output is 1001
# @time : 180 - for load = 0 output is 1010
# @time : 190 - for load = 0 output is 1011
# @time : 200 - for load = 0 output is 0000
# @time : 210 - for load = 0 output is 0001
# @time : 200 - for load = 0 output is 0000
# @time : 210 - for load = 0 output is 0001
# @time : 220 - for load = 0 output is 0010
# @time : 230 - for load = 0 output is 0011
# @time : 240 - for load = 0 output is 0100
# @time : 250 - for load = 0 output is 0101
# @time : 260 - for load = 0 output is 0110
# @time : 270 - for load = 0 output is 0111
# @time : 280 - for load = 0 output is 1000
# @time : 290 - for load = 0 output is 1001
# @time : 300 - for load = 0 output is 1010
# @time : 310 - for load = 0 output is 1011
# @time : 320 - for load = 0 output is 0000
# @time : 330 - for load = 0 output is 0001
# @time : 340 - for load = 0 output is 0010
# @time : 350 - for load = 0 output is 0011
# @time : 360 - for load = 0 output is 0100
# @time : 370 - for load = 0 output is 0101
# @time : 380 - for load = 0 output is 0110
# ** Note: $finish    : C:/Users/Aadhithan/Documents/count_tb.v(24)
#    Time: 385 ps  Iteration: 0  Instance: /count_tb
# 1
```

RTL:



Design : up/down counter:

Code:

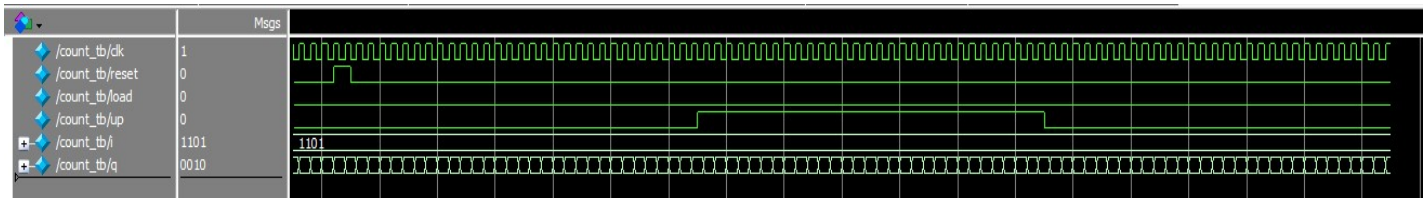
```
module count4(input clk,reset,load,up,[3:0]i,output reg [3:0]q);
always@(posedge clk)begin
    if(reset) q <= 4'd0;
    else if (load) q <= i;
    else if (up) q <= q+1;
    else if (!up) q<=q-1;
end
endmodule
```

Testbench:

```
module count_tb();
reg clk,reset,load,up;
reg [3:0]i;
wire [3:0]q;
count4 dut(clk,reset,load,up,i,q);
initial begin
    clk = 1;
    forever #5 clk = ~clk;
end
initial begin
    $monitor("@time : %3d -
 for load = %b for operation up =%b  output is %b",$time,load,up,q);
    reset <= 1;
    #15;
    reset <= 0;
    up <=1;
    load <= 1;
    i <= 4'b1101;
    #15;
    load <=0;
    #40;
    up <=0;
    #40;
    reset <= 1;
    #15;
    reset <= 0;
    #300;
    up<=1;
    #300;
```

```
    up<=0;
    #300;
    $finish;
end
endmodule
```

Wave:



Output:

```
VSIM 7> run -all
# @time :   0 - for load = x for operation up =x  output is xxxx
# @time :  10 - for load = x for operation up =x  output is 0000
# @time :  15 - for load = 1 for operation up =1  output is 0000
# @time :  20 - for load = 1 for operation up =1  output is 1101
# @time :  30 - for load = 0 for operation up =1  output is 1101
# @time :  40 - for load = 0 for operation up =1  output is 1110
# @time :  50 - for load = 0 for operation up =1  output is 1111
# @time :  60 - for load = 0 for operation up =1  output is 0000
# @time :  70 - for load = 0 for operation up =0  output is 0001
# @time :  80 - for load = 0 for operation up =0  output is 0000
# @time :  90 - for load = 0 for operation up =0  output is 1111
# @time : 100 - for load = 0 for operation up =0  output is 1110
# @time : 110 - for load = 0 for operation up =0  output is 1101
# @time : 120 - for load = 0 for operation up =0  output is 0000
# @time : 130 - for load = 0 for operation up =0  output is 1111
# @time : 140 - for load = 0 for operation up =0  output is 1110
# @time : 150 - for load = 0 for operation up =0  output is 1101
# @time : 160 - for load = 0 for operation up =0  output is 1100
# @time : 170 - for load = 0 for operation up =0  output is 1011
# @time : 180 - for load = 0 for operation up =0  output is 1010
# @time : 190 - for load = 0 for operation up =0  output is 1001
# @time : 200 - for load = 0 for operation up =0  output is 1000
# @time : 210 - for load = 0 for operation up =0  output is 0111
# @time : 220 - for load = 0 for operation up =0  output is 0110
# @time : 230 - for load = 0 for operation up =0  output is 0101
# @time : 240 - for load = 0 for operation up =0  output is 0100
# @time : 250 - for load = 0 for operation up =0  output is 0011
# @time : 260 - for load = 0 for operation up =0  output is 0010
# @time : 270 - for load = 0 for operation up =0  output is 0001
# @time : 280 - for load = 0 for operation up =0  output is 0000
# @time : 290 - for load = 0 for operation up =0  output is 1111
# @time : 300 - for load = 0 for operation up =0  output is 1110
# @time : 310 - for load = 0 for operation up =0  output is 1101
# @time : 320 - for load = 0 for operation up =0  output is 1100
# @time : 330 - for load = 0 for operation up =0  output is 1011
# @time : 340 - for load = 0 for operation up =0  output is 1010
# @time : 350 - for load = 0 for operation up =0  output is 1001
# @time : 360 - for load = 0 for operation up =0  output is 1000
# @time : 370 - for load = 0 for operation up =0  output is 0111
# @time : 380 - for load = 0 for operation up =0  output is 0110
# @time : 390 - for load = 0 for operation up =0  output is 0101
# @time : 400 - for load = 0 for operation up =0  output is 0100
# @time : 410 - for load = 0 for operation up =0  output is 0011
# @time : 420 - for load = 0 for operation up =0  output is 0010
# @time : 425 - for load = 0 for operation up =1  output is 0010
# @time : 430 - for load = 0 for operation up =1  output is 0011
# @time : 440 - for load = 0 for operation up =1  output is 0100
# @time : 450 - for load = 0 for operation up =1  output is 0101
# @time : 460 - for load = 0 for operation up =1  output is 0110
# @time : 470 - for load = 0 for operation up =1  output is 0111
# @time : 480 - for load = 0 for operation up =1  output is 1000
# @time : 490 - for load = 0 for operation up =1  output is 1001
# @time : 500 - for load = 0 for operation up =1  output is 1010
# @time : 510 - for load = 0 for operation up =1  output is 1011
# @time : 520 - for load = 0 for operation up =1  output is 1100
# @time : 530 - for load = 0 for operation up =1  output is 1101
# @time : 540 - for load = 0 for operation up =1  output is 1110
# @time : 550 - for load = 0 for operation up =1  output is 1111
# @time : 560 - for load = 0 for operation up =1  output is 0000
# @time : 570 - for load = 0 for operation up =1  output is 0001
# @time : 580 - for load = 0 for operation up =1  output is 0010
# @time : 590 - for load = 0 for operation up =1  output is 0011
# @time : 600 - for load = 0 for operation up =1  output is 0100
# @time : 610 - for load = 0 for operation up =1  output is 0101
# @time : 620 - for load = 0 for operation up =1  output is 0110
# @time : 630 - for load = 0 for operation up =1  output is 0111
# @time : 640 - for load = 0 for operation up =1  output is 1000
# @time : 650 - for load = 0 for operation up =1  output is 1001
# @time : 660 - for load = 0 for operation up =1  output is 1010
# @time : 670 - for load = 0 for operation up =1  output is 1011
# @time : 680 - for load = 0 for operation up =1  output is 1100
# @time : 690 - for load = 0 for operation up =1  output is 1101
# @time : 700 - for load = 0 for operation up =1  output is 1110
# @time : 710 - for load = 0 for operation up =1  output is 1111
# @time : 720 - for load = 0 for operation up =1  output is 0000
```

```
# @time : 725 - for load = 0 for operation up =0  output is 0000
# @time : 730 - for load = 0 for operation up =0  output is 1111
# @time : 740 - for load = 0 for operation up =0  output is 1110
# @time : 750 - for load = 0 for operation up =0  output is 1101
# @time : 760 - for load = 0 for operation up =0  output is 1100
# @time : 770 - for load = 0 for operation up =0  output is 1011
# @time : 780 - for load = 0 for operation up =0  output is 1010
# @time : 790 - for load = 0 for operation up =0  output is 1001
# @time : 800 - for load = 0 for operation up =0  output is 1000
# @time : 810 - for load = 0 for operation up =0  output is 0111
# @time : 820 - for load = 0 for operation up =0  output is 0110
# @time : 830 - for load = 0 for operation up =0  output is 0101
# @time : 840 - for load = 0 for operation up =0  output is 0100
# @time : 850 - for load = 0 for operation up =0  output is 0011
# @time : 860 - for load = 0 for operation up =0  output is 0010
# @time : 870 - for load = 0 for operation up =0  output is 0001
# @time : 880 - for load = 0 for operation up =0  output is 0000
# @time : 890 - for load = 0 for operation up =0  output is 1111
# @time : 900 - for load = 0 for operation up =0  output is 1110
# @time : 910 - for load = 0 for operation up =0  output is 1101
# @time : 920 - for load = 0 for operation up =0  output is 1100
# @time : 930 - for load = 0 for operation up =0  output is 1011
# @time : 940 - for load = 0 for operation up =0  output is 1010
# @time : 950 - for load = 0 for operation up =0  output is 1001
# @time : 960 - for load = 0 for operation up =0  output is 1000
# @time : 970 - for load = 0 for operation up =0  output is 0111
# @time : 980 - for load = 0 for operation up =0  output is 0110
# @time : 990 - for load = 0 for operation up =0  output is 0101
# @time : 1000 - for load = 0 for operation up =0  output is 0100
# @time : 1010 - for load = 0 for operation up =0  output is 0011
# @time : 1020 - for load = 0 for operation up =0  output is 0010
# ** Note: $finish    : C:/Users/Aadhithan/Documents/Verilog_labs/
#    Time: 1025 ps  Iteration: 0  Instance: /count_tb
```

RTL: