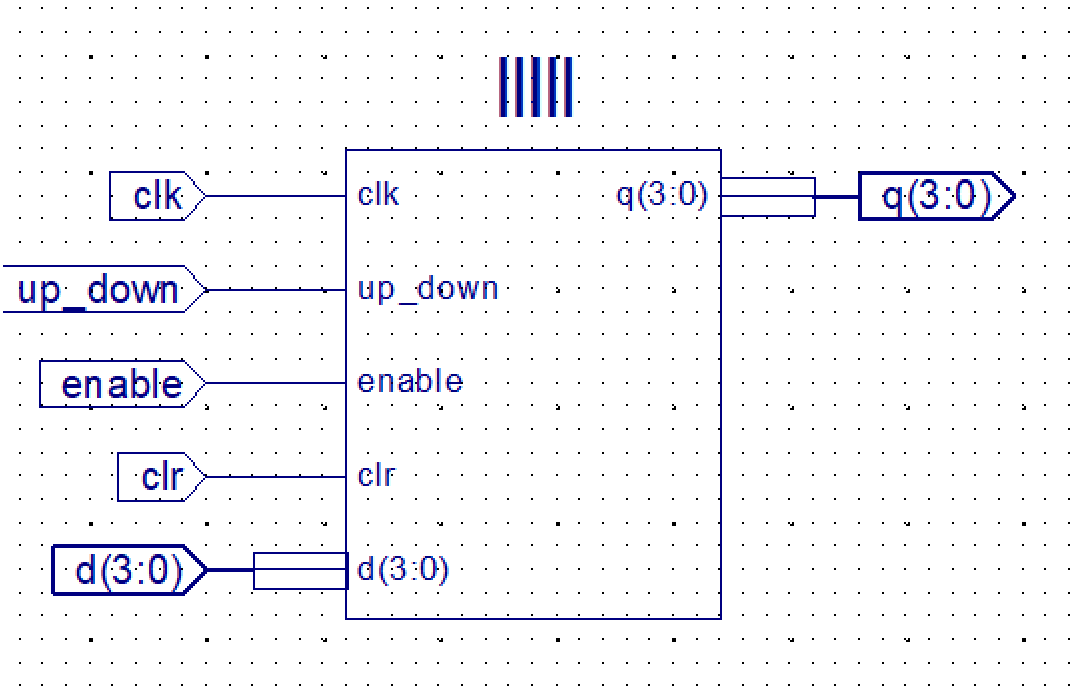
Printout 1/2

```
`timescale 1ns / 1ps
module lllll(clk , up_down , enable , clr , q ,d    );
             input clk , up_down , enable , clr;
            output[3:0]q;
            input[3:0]d;
            reg[3:0]tem = 4'b0000;
            always@(clk,clr)begin
            if(clr)
                        tem = 4'b000;
            else
                        if(!enable)
                                    tem = d;
                        else
                                    if(up_down)
                                                tem = tem + 1'b1;
                                    else
                                                tem = tem - 1'b1;
            end
            assign q = tem;
endmodule
```

```
NET "Clk"  LOC = P6 ;
NET "up_down" LOC = P7 ;
NET "enable" LOC = P8 ;
NET "clr" LOC = P9 ;


NET "d<3>"  LOC = P1;
NET "d<2>"  LOC = P2;
NET "d<1>"  LOC = P3;
NET "d<0>"  LOC = P4;

NET "q<3>"  LOC = P35;
NET "q<2>"  LOC = P36;
NET "q<1>"  LOC = P37;
NET "q<0>"  LOC = P38;
```



printout 2/2

```
`timescale 1ns / 1ps
module sdsd(input clk,output q_2hz,q_083hz);
    reg rq_2hz = 1'b0;
    reg rq_083hz = 1'b0;
    reg[23:0]Counter_2hz = 0;
    reg[23:0]Counter_083hz= 0;
    parameter fin = 20_000_000;
    parameter fout_2hz = 2;
    parameter Coust_2hz = fin/(2*fout_2hz);
    parameter fout_083hz = 083;
    parameter Coust_083hz = fin/(2*fout_083hz);
    always@(posedge clk)
    begin
        Counter_2hz <= Counter_2hz+1'b1;
        Counter_083hz <= Counter_083hz+1'b1;
        if(Counter_2hz == Coust_2hz)
            begin
            Counter_2hz <= 0;
            rq_2hz <= ~rq_2hz;
            end
        if(Counter_083hz == Coust_083hz)
            begin
            Counter_083hz <= 0;
            rq_083hz <= ~rq_083hz;
            end
    end
endmodule
```

```
NET "clk" LOC = P1;
NET "q_2hz" LOC = P11;
NET "q_083hz" LOC = P12;
```