Q2

-Input evnt high when conditions met(done)

-Internal counter that counts number of cycles which evnt is high (done)

-When count exceeds threshold(6 bit thresh input), asserts alarm output

and stops counter(done)

-rst(done)

-alarm is a combinational circuit determined form count value and threshold

Module alarmsound(input evnt, clk, rst,sel, [5:0]thresh,

Output ~~reg~~ alarm);

Reg [5:0] count;

always@(posedge clk)

begin

~~if(evnt)~~

if(rst)

count<=6’b000000;

else

if((evnt)&&(!alarm))

count <= count +1’b1;

end

assign alarm = (count >thresh);

endmodule

~~count<= count +1’b1;~~

~~if (count>thresh)~~

~~count<=thresh+1’b1;~~

~~alarm ==1’b1;~~

~~else~~

~~alarm ==1’b0;~~

~~end~~

~~endmodule~~

Q3a)

Module bcount(input clk, rst,

Output reg[4:0] q );

always@(posedge clk)

begin

if(rst)

q<= 5’b00000;

else begin

~~q<= q+1’b1;~~

if(q== 5’b10101)

q<=5’b00000;

else

q <=q+1’b1;

end

end

endmodule

b)

Module bcount(input clk, rst, [4:0]countmax~~[4:0]~~,

Output reg[4:0] q );

always@(posedge clk)

begin

if(rst)

q<= 5’b00000;

else begin



~~q<= q+1’b1;~~

if(q== countmax~~+1~~)

q<=5’b00000;

else

q<=q+1’b1;

end

end

endmodule

c)

Module bcount(input clk, rst, [5:0]start\_val,

Output reg[5:0] q );

always@(posedge clk)

begin

if(rst)

q<= start\_val;

else

q<= q-1’b1;

end



endmodule