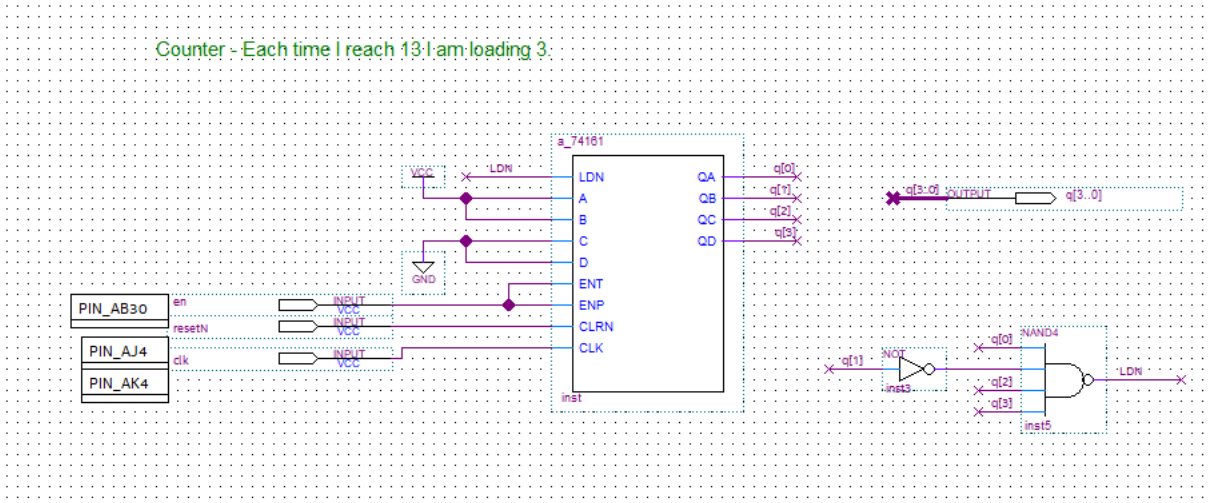
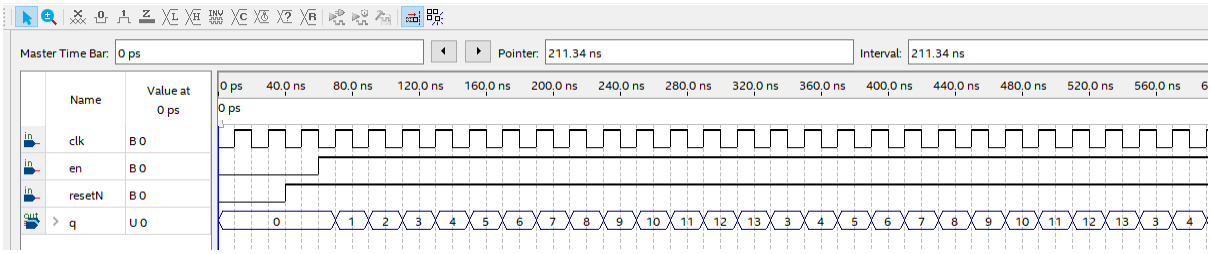
Counter BDF:



Simulation Waveform:



Counter Code:

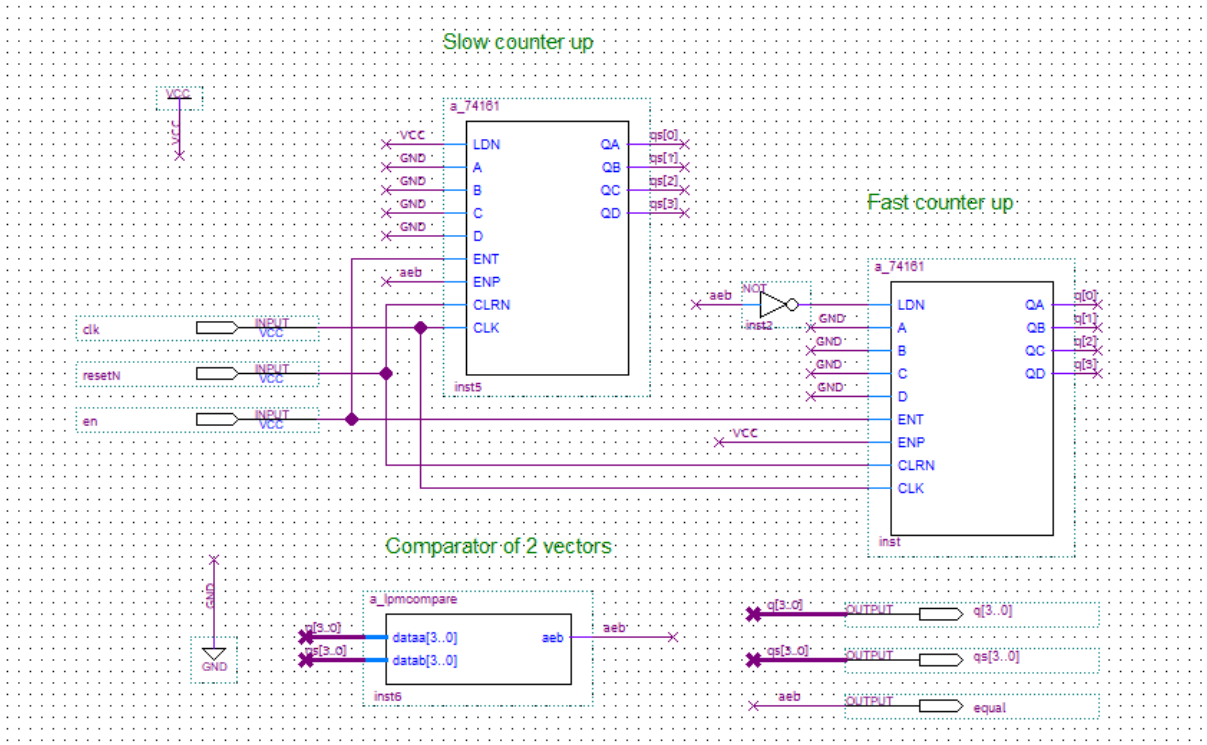
```
// A default counter

module a_74161(
        input logic LDN, //Load
        input logic A,
        input logic B,
        input logic C,
        input logic D,


        input logic ENT,
        input logic ENP,
        input logic CLRN,
        input logic CLK,
        output logic QA,
        output logic QB,
        output logic QC,
        output logic QD

);


logic [3:0] count;
```

```
always_ff @(posedge CLK, negedge CLRN)
begin
        if (!CLRN) begin
                count <= 4'b0;
        end
        else if (ENP  && ENT  ) begin
          if ( !LDN  ) begin
                count <= {D,C,B,A};
          end
        else
                count <= count + 1'b1;

        end
end

assign QA = count[0];
assign QB = count[1];
assign QC = count[2];
assign QD = count[3];

endmodule
```

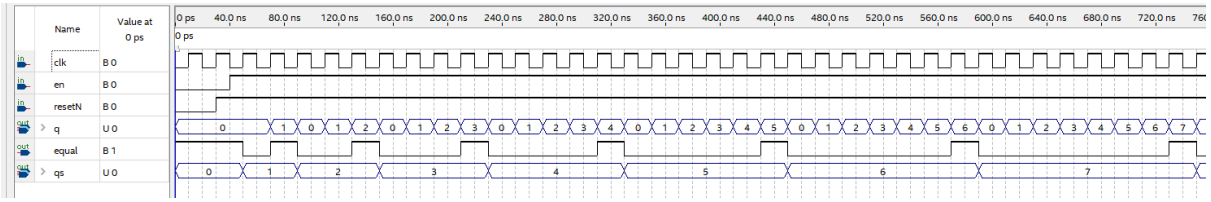
Cascaded Counter:

We have two counters: a fast counter and a slow counter, along with a comparator. Each time the outputs of the slow and fast counters are equal, the slow counter increases and the fast counter loading zero.

BDF:



Simulation Waveform:



Comparator Code:

```
// Comparator - compare the output between Counter#1 and Coounter#2

module a_lpmcompare (
 input logic [3:0] dataa,
 input logic [3:0] datab,
 output logic aeb
);

assign aeb = ( dataa == datab ) ? 1'b1 : 1'b0;

endmodule
```