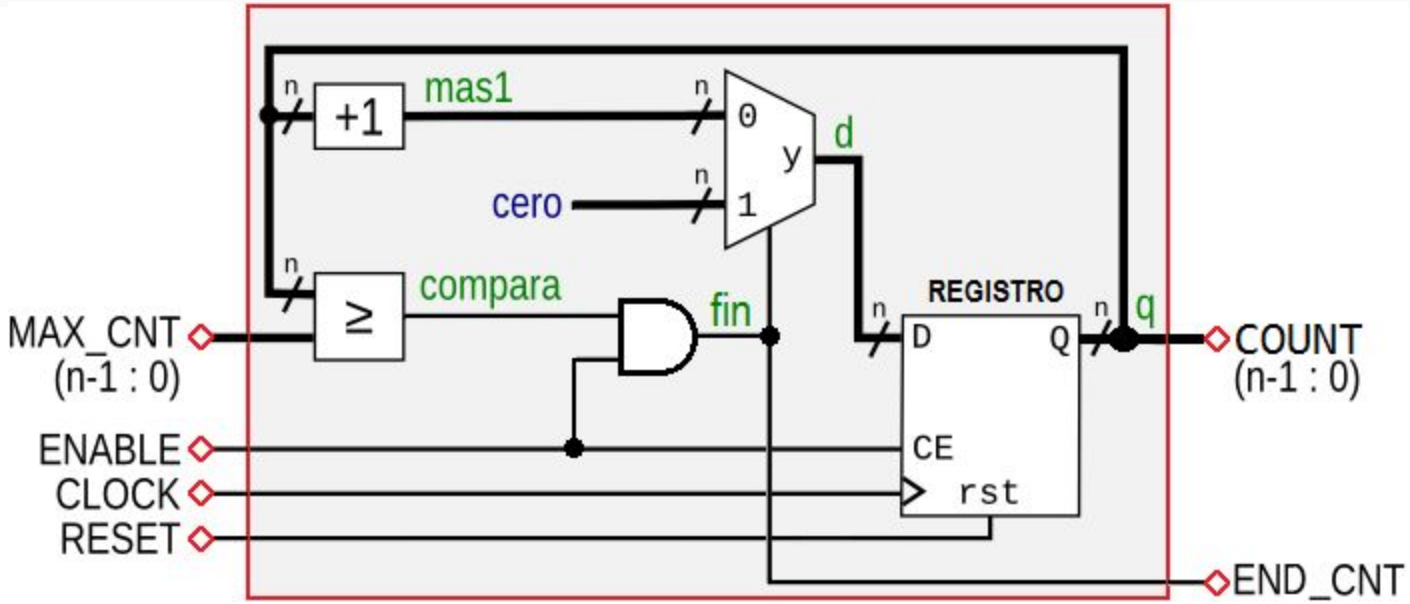
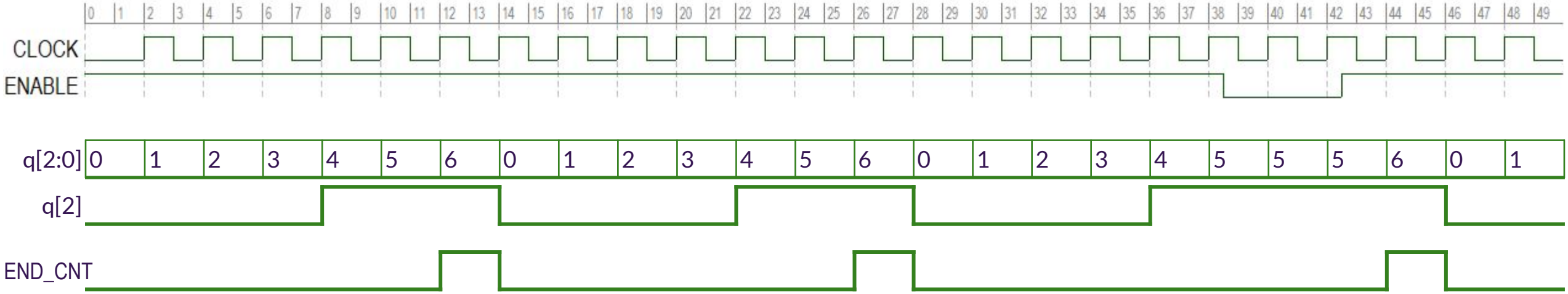
Contador módulo variable



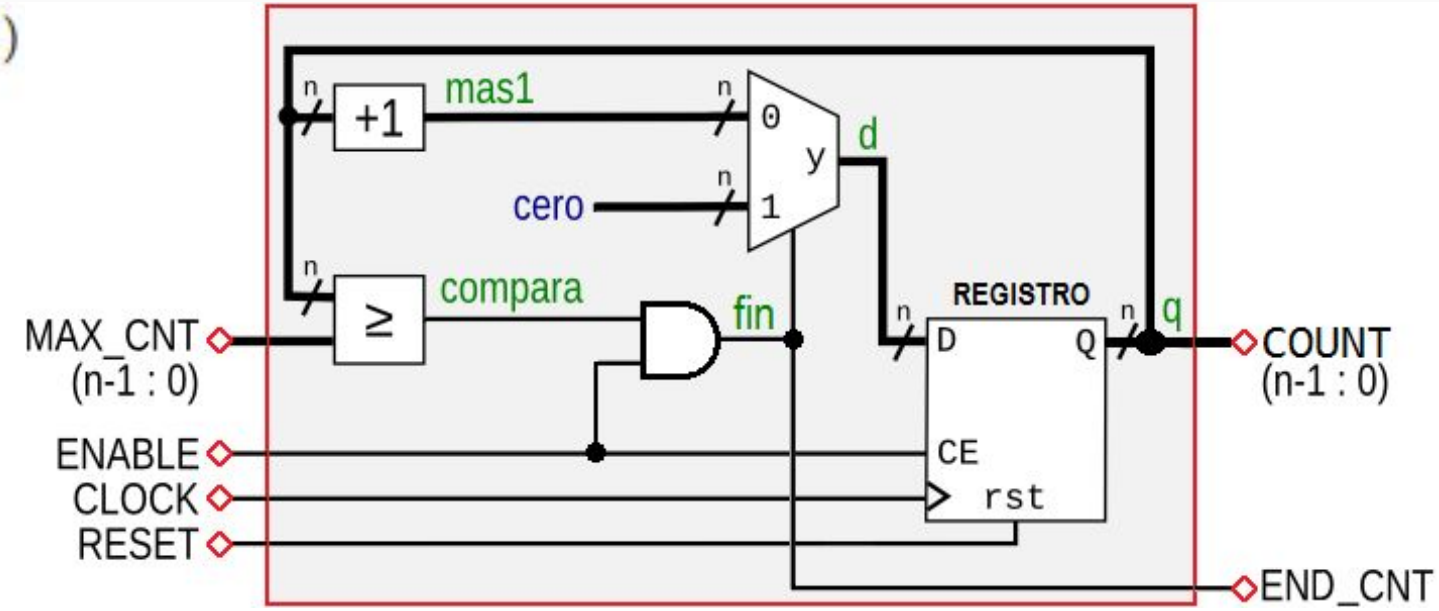
EJEMPLO:

n = 3, RESET = 0, MAX_CNT = 6

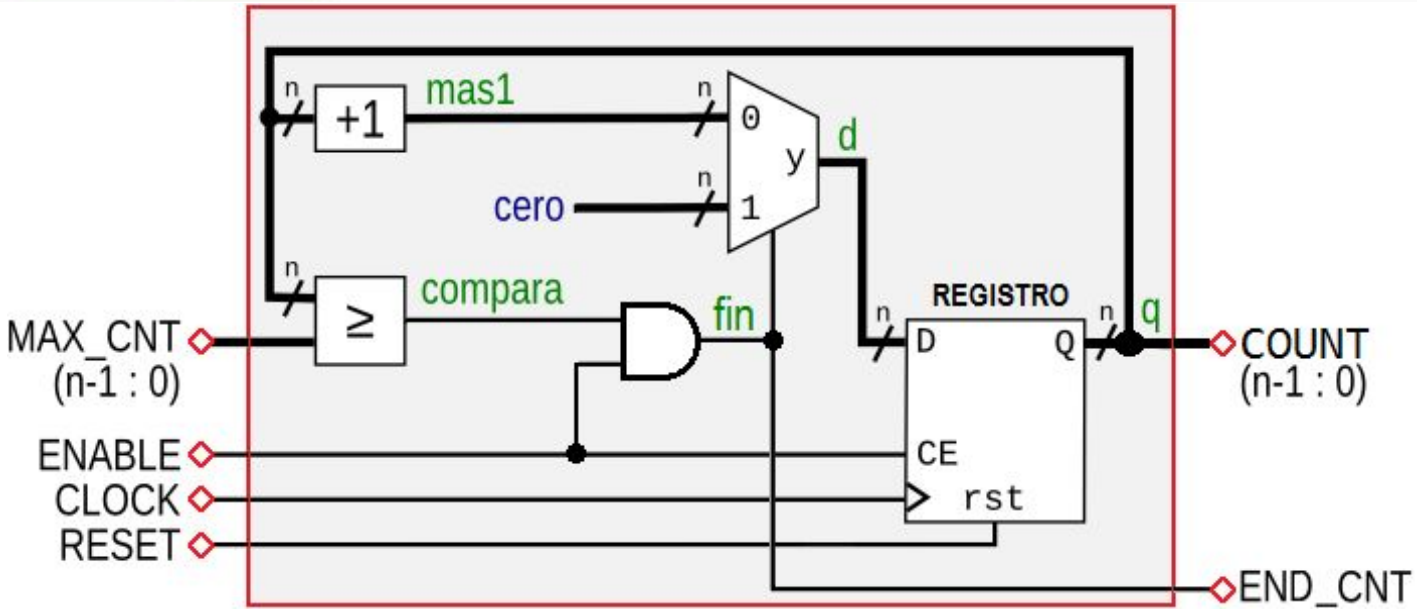


Descripción en verilog

```
1  module contador #(parameter n=4)
2      input CLOCK,
3      input RESET,
4      input ENABLE,
5      input  [n-1:0] MAX_CNT,
6      output END_CNT,
7      output [n-1:0] COUNT
8  );
9
10 wire [n-1:0] d, mas1;
11 wire compara, fin;
12 reg  [n-1:0] q = {n{1'b0}};
13
```



Descripción en verilog




```
14  // sumador
15  assign mas1 = q+1;
16
17  //comparador
18  assign compara = (q>=MAX_CNT);
19  assign fin = compara & ENABLE;
20
21  //mux 2-1
22  assign d = fin? {n{1'b0}} : mas1;
23
24  //registro
25  always @(posedge CLOCK)
26      if (RESET)
27          q <= {n{1'b0}};
28      else if (ENABLE)
29          q <= d;
30
31  // salidas
32  assign COUNT = q;
33  assign END_CNT = fin;
34  endmodule
```

Testbench

```
1  `timescale 1ns/10ps
2  module contador_tb();
3  parameter bits = 3;
4  parameter dt = 50;
5
6      reg clock;
7      reg reset;
8      reg enable;
9      wire fin_cuenta;
10     wire [bits-1:0] cuenta;
11
12 contador #(bits) uut(
13     .CLOCK(clock),
14     .RESET(reset),
15     .ENABLE(enable),
16     .MAX_CNT(3'd6),
17     .END_CNT(fin_cuenta),
18     .COUNT(cuenta)
19 );
20
```

```
21 always begin
22     clock = 1'b0;   #(1*dt);
23     clock = 1'b1;   #(1*dt);
24 end
25
26 initial begin
27     reset = 1'b0;
28     enable = 1'b1;  #(dt* 38.5);
29     enable = 1'b0;  #(dt* 4);
30     enable = 1'b1;  #(dt* 7.5);
31
32     $finish;
33 end
34
35 initial begin
36     $dumpfile("ARCHIVO.vcd");
37     $dumpvars;
38 end
39 endmodule
```

Implementación

```
1  module contador_labsland(
2      input  CLOCK_50,
3      input  [1:0] SW,
4      output [9:5] LEDR
5  );
6
7  wire [29:0] cuenta;
8  assign LEDR[9:5] = cuenta[29:25];
9
10 contador #(30) uut(
11     .CLOCK(CLOCK_50),
12     .RESET(SW[1]),
13     .ENABLE(SW[0]),
14     .MAX_CNT(30'd500_000_000),
15     .COUNT(cuenta)
16 );
17
18 endmodule
```