APARTADO 1

```
module cont(
  input  ck,up,reset,
  output  [2:0] z,
  output cy);

          reg [2:0] q;

          assign cy = &q;
          assign z = q;

          always @(posedge ck)
            if (reset==1 )
              q <= 0;
            else if (up==1)
              q <= q + 1;

endmodule
```

APARTADO 2

```
module dec3a8(
  input  [2:0] a,
  output reg [7:0] w);

  always @ (a)
    begin
      w = 0;
      w[a] = 1;
    end

endmodule
```

APARTADO 3

```
module sistema(
   input ck, up, reset,
   output [7:0] z,
   output cy);

   wire [3:0] net;

   cont instancia_cont ( .ck(ck), .up(up), .reset(reset), .z(net), .cy(cy));
   dec3a8 instancia_dec( .a(net), .w(z));

endmodule
```

APARTADO 4

```
module sistema_tb;

 reg tb_ck;    // entradas
 reg tb_up;
 reg tb_reset;

 wire [7:0] tb_z;    // salidas
 wire tb_cy;

 sistema instancia_sistema (.ck(tb_ck),  .up(tb_up),  .reset(tb_reset), .z(tb_z), .cy(tb_cy));


 always
  begin
   #10;
   tb_clk = ~tb_clk;
  end

 initial begin
   tb_ck = 0;
   tb_up = 0;
   tb_reset = 0;


   @(negedge tb_ck)
   tb_reset = 1;
   @(negedge tb_ck)
   tb_reset = 0;

   repeat(2)
     @(negedge tb_ck);       // espera 2 bajadas de reloj
   tb_up = 1;         // Pongo a contar al contador

   repeat(10)
     @(negedge tb_ck);       // espera 10 bajadas de reloj
   tb_reset = 1;         // activo reset sincrono prioritario sobre up

   $finish;

 end

endmodule
```