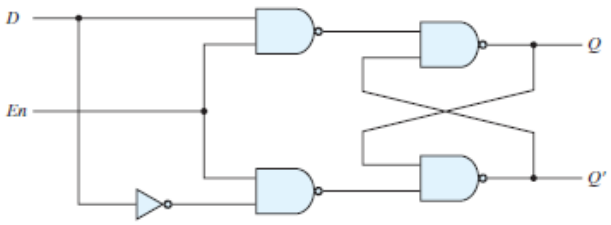
1.- Determina de forma precisa el tipo de dispositivo que se define en este módulo Verilog:

```
module D_latch (Q, D, enable);
output Q;
input D, enable;
reg Q;
always @ (enable or D)
  if (enable) Q <= D;       // Equivale a: if (enable == 1)
endmodule

// Alternativa sintaxis (Verilog 2001, 2005)

module D_latch (output reg Q, input enable, D);
always @ (enable, D)
   if (enable) Q <= D;
endmodule
```

Respuesta: **D Latch (Transparent Latch)**

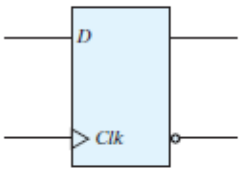


En	*D*	Next state of Q
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

2.- Determina de forma precisa el tipo de dispositivo que se define en este módulo Verilog:

```
module D_FF (Q, D, Clk);
output Q;
input D, Clk;
reg Q;
always @ ( posedge Clk)
Q <= D;
endmodule
```

Respuesta**:** **D flip-flop sin reset**

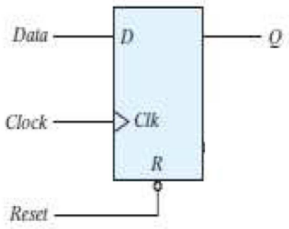


Clk	*D*	*Q*
↑	0	0
↑	1	1

3.- Determina de forma precisa el tipo de dispositivo que se define en este módulo Verilog:

```
module DFF ( output reg Q, input D, Clk, rst);
always @ ( posedge Clk, negedge rst)
    if (!rst)
        Q <= 1'b0;                  // Equivale a: if (rst == 0)
    else
        Q <= D;
endmodule
```

Respuesta: D flip-flop con reset asíncrono



R	Clk	D	Q
0	X	X	0
0	↑	0	0
0	↑	1	1

4.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, enable, q);
input   clk, enable;
output  reg [7:0] q;

always @ (posedge clk)
  begin
    if (enable)
      q <= q + 1;                   // si no  q mantendrá el valor q <= q
  end
endmodule
```

Respuesta:

Contador síncrono activo a flanco positivo y con habilitación (cuenta si enable = 1)

5.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, reset, ld, d, q);
input   clk, reset, ld;
input   [7:0] d;
output  reg [7:0] q;

always @ (posedge clk, negedge reset)
          if (reset == 0)
                    q <= o;
          else if (!ld)
                     q <= d;
           else
                     q <= q + 1;
endmodule
```

Respuesta:

Contador síncrono activo a flanco positivo con reset asíncrono activo a nivel bajo y con carga en paralelo síncrona con ld =0.

6.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, up_down, q);
input   clk, up_down;
output  [7:0] q;
reg     [7:0] q;
integer direction;

always @ (posedge clk)
  begin

    if (up_down)
      direction = 1;
    else
      direction = -1;

    q <= q + direction;

  end
endmodule
```

Respuesta:

Contador ascendente-descendente síncrono activo a flanco positivo.

- Cuenta descendente si up_down =0.
- Cuenta ascendente si up_down =1.

7.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, reset, enable , ld, q);
input   clk, reset, enable, ld;
input   [7:0] d;
output  reg [7:0] q;

always @ (posedge clk)
        if (reset)
                q <= 0;
        else if (!ld)
                q<= d;
        else if (enable)
                q<= q + 1;
endmodule
```

Respuesta:

- Contador ascendente activo a flanco positivo con carga paralelo síncrona y reset síncrono activo a nivel alto.
- Si ld = 0, en el flanco positivo carga en la salida q, el valor de d.
- Si ld =1, cuenta ascendente.

8.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, up_down, enable, q,);
input   clk, up_down, enable;
output  reg [7:0] q;
integer direction;

always @ (posedge clk)
        if (enable ==  0)
                q <= q;
        else if  (up_down)
                q <= q + 1;
        else
                q <= q -1;
endmodule
```

Respuesta:

Contador ascendente-descendente síncrono activo a flanco positivo con habilitación (enable).
Si enable = 1, cuenta descendente si up_down =0; cuenta ascendente si up_down = 1.
Si enable = 0, no cuenta.

9.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, clear, enable, q);
input   clk, clear, enable;
output  [7:0] q;
reg     [7:0] q;

always @ (negedge clk) begin
if (!clear)
        q <= 0;
else if (enable)
        q <= q + 1;
endmodule
```

Respuesta:
Contador ascendente activo a flanco negativo con entrada de habilitación y con reset síncrono.
Si enable = 1, si clear = 1, cuenta descendente; si clear = 0, se resetea (q <=0) con el siguiente flanco negativo.
Si enable = 0, no cuenta y mantiene el valor registrado.

10.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, reset, up_down, ld, q,);
input   clk, up_down, ld;
output  reg [7:0] q;

always @ (negedge clk, negedge reset)
        if (reset == 0)
                q <= 0;
        else if (!ld)
                q <= d;
        else if (up_down)
                q <= q + 1;
        else
                q <= q - 1;
always @ (, negedge reset)
endmodule
```

Respuesta:
Contador ascendente-descendente síncrono activo a flanco negativo, con reset asíncrono activo a nivel bajo y con carga paralelo síncrona.
Si ld = 1, cuenta descendente si up_down =0 y cuenta ascendente si up_down =1.
Si ld = 0, carga en q el valor de d, con el flanco descendente.

11.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, enable, clear, ld, q);
input   clk, enable, clear, ld;
output  reg [7:0] q;

always @ (negedge clk) begin
if (!clear)
q <= 0;
else if (!ld)
q <= d;
else if (enable)
q <= q + 1;
end
endmodule
```

Respuesta:
Contador ascendente síncrono activo a flanco positivo, con reset síncrono y con carga paralelo síncrona y entrada de habilitación sólo para el conteo.

- Si clear = 0 se resetea con el flanco negativo.
- Si clear = 1 y enable = 0 no cuenta, y además, si ld = 0, carga en q el valor de d, con el flanco descendente.
- Si clear = 1 y enable = 1:
    - Si ld = 1, cuenta ascendente
    - Si ld = 0, carga en q el valor de d, con el flanco descendente.

12.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, enable, clear, up_down, q,);
input   clk, enable, clear, up_down;
output  reg [7:0] q;
integer direction;
```

```
always @ (negedge clk)
begin
if (up_down)
        direction = 1;
else
        direction = -1;
if (!clear)
        q <= 0;
else if (enable)
        q <= q + direction;
end
endmodule
```

Respuesta:
Contador ascendente-descendente activo a flanco descendente, con reset síncrono y entrada de habilitación.

Si clear = 0 se resetea con el flanco descendente
Si clear = 1 y enable = 1:

- Si up_down =0 cuenta descendente.
- Si up_down =1 cuenta ascendente .
- Si clear = 1 y enable = 0 no cuenta.

13.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, reset, enable, up_down, ld, q);
input   clk, enable, up_down, ld;
output reg rebose;
output  reg [7:0] q;

always @ (posedge clk, posedge reset)
        if (reset)
                q <= 0;
        else if (!ld)
                q <= d;
        else if (enable== 0)
                q <= q;
        else if (up_down)
                q <= q + 1;
        else
                q <= q - 1;
always @ (enable, up_down, q)          //Bloque combinacional
        if ((enable && up_down && q == 255 ) || (enable && !up_down && q == 0 ))
```

```
                rebose = 1;
        else    rebose = 0;
endmodule
```

Respuesta:
Contador ascendente-descendente síncrono activo a flanco ascendente, con reset asíncrono activo a nivel alto y con carga paralelo síncrona y entrada de habilitación sólo para el conteo.

Si enable = 1:

- Si ld = 1, cuenta descendente si up_down =0, y cuenta ascendente si up_down =1.
- Si ld = 0, carga en q el valor de d con el flanco ascendente.

Si enable = 0 no cuenta, y además, si ld = 0, carga en q (salida) el valor de d, con el flanco ascendente.

Presenta una salida de rebose que se pone a 1 cuando el contador está habilitado y está en ascendente y presenta su valor máximo (255) o cuando el contador está habilitado y está en descendente y presenta su valor mínimo (0)

14.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module counters (clk, reset, en, q);
input   clk, reset, en;
output  reg [7:0] q;

always @ (posedgeclk)
        if (reset == 0)
                q <= 0;
        else if (en == 0)
                q <= q;
        else if (q == 199)
                q <= 0;
        else
                q <= q + 1;
endmodule
```

Respuesta:
Contador ascendente activo a flanco ascendente módulo 200 (cuenta de 0 a199 inclusive), con reset síncrono activo a nivel bajo y entrada de habilitación (en) activa a nivel alto.

15.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module cont (CLK, RES, Q);
```

```
input CLK;
input RES;
output [3:0] Q;
reg [3:0] Q, nQ;

always @(posedge CLK or negedge RES)
        if(RES==0) Q<=0;
        else Q<=nQ;
always @(Q)
        nQ=Q+1;
endmodule
```

Respuesta:
Contador ascendente con reset asíncrono activo a nivel bajo.
El contador está diseñado como máquina de estados síncrona, definiendo el estado siguiente en un bloque combinacional a partir del estado actual.

16.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module cont (CLK, RES, UD, Q);
input CLK;
input RES;
input UD;
output [3:0] Q;
reg [3:0] Q, nQ;

always @(posedge CLK or negedge RES)
        if(RES==0)
                Q<=0;
        else
                Q<=nQ;

always @(Q or UD)
        if(UD==1)
                nQ=Q+1;
        else
                nQ=Q-1;

endmodule
```

Respuesta:
Contador ascendente-descendente con reset asíncrono activo a nivel bajo.

El contador está diseñado como máquina de estados síncrona, definiendo el estado siguiente en un bloque combinacional a partir del estado actual.

17.- Determina de forma precisa el tipo de contador que se define en este módulo Verilog:

```
module cont_mN (input clk, input clr, output reg [N-1:0] Q );
  parameter N=3;
  parameter M=9;

always @(posedge clk or posedge clr)
        if(clr)
                Q<=0;
        else if(Q==M)
                Q<=0;
        else
                Q<=Q+1;

endmodule
```

Respuesta:
Contador ascendente módulo M (módulo 9, en este ejemplo) activo a flanco ascendente con reset asíncrono activo a nivel alto.
El módulo (número de estados del contador) está parametrizado y basta con cambiar los parámetros para realizar otro contador de distinto módulo.

18.- En el módulo clkdiv, si clk presenta una frecuencia de 50 MHz,

a) Determinar la frecuencia de salida de las salidas clka y club.
b) En estas señales de salida ¿Cuál será la relación entre tiempo en ON (nivel alto) y su periodo? (o sea, ¿cuál es su duty cycle o ciclo de trabajo?)

```
module clkdiv( input mclk,  input clr, output clka, output clkb);

reg [15:0] q;
always @(posedge mclk or posedge clr)
  if(clr)
        q <= 0;
  else
        q <= q + 1;


assign clka  = q[0];
assign clkb = q[15];
```

endmodule

Respuesta:

a) clka corresponde a la salida menos significativa del contador.
La salida q[0] del registro que se asigna a la salida clka del contador divide la frecuencia del reloj del contador por 2. Luego la frecuencia de clka será de 50MHz/2=25MHz
La salida q[1] del registro divide la frecuencia del reloj del contador por 2^2=4.
La salida q[2] del registro divide la frecuencia del reloj del contador por 2^3=8.
….
La salida q[15] del registro que se asigna a la salida clkb del contador divide la frecuencia del reloj del contador por 2^16=65536, luego la frecuencia de clkb será 762,94 Hz.

b) El contador no tiene un tope de conteo establecido por lo que contará en anillo, es decir que al llegar al valor máximo que puede registrar el registro q del contador, en este caso (2^16)-1=65535=16'b1111111111111111, el contador pasará a cero. Es estas condiciones todos los bits del contador, incluídos q[0] y q[15] estarán la mitas del periodo de su señal a 0 y la otra mitad a 1. Por tanto, el DUTY CYCLE o ciclo de trabajo es del 50 % ó lo que es igual de 0,5.