

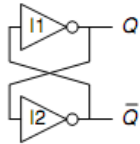
Semillero ADT: Circuitos basados en Flip-Flops 1ª Parte

Objetivos:

Conocer el funcionamiento básico de un Flip-Flop tipo D y las distintas configuraciones posibles.

Hacer la descripción de hardware de un circuito básico de sincronización.

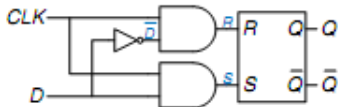
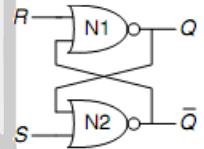
Marco de Referencia:



Una vez conocido el funcionamiento de las compuertas lógicas el siguiente paso es hacer arreglos de compuertas con distintos usos, un arreglo muy conocido es el "Latch". En la primera figura se muestra un arreglo de compuertas NOT que logran un circuito Bi-estable (De dos estados distintos) sin entradas. (Analícelo: si $Q=0$ o si $Q=1$).

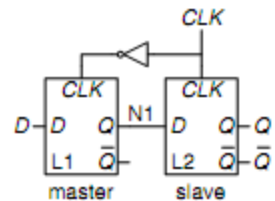
La siguiente figura es un Latch SR el cual tiene un comportamiento Bi-estable pero ahora depende de las entradas. (Analícelo: si $R=0$ y $S=1$, si $R=1$ y $S=0$, si $R=1$ y $S=1$ y especialmente si $R=0$ y $S=0$).

Los Latch no son deseables en el diseño de hardware porque entran en estados permanentes o inesperados como en el caso especial del Latch SR.



Sin embargo se pueden hacer configuraciones estables de estos, como es el caso del latch D. Analice el comportamiento del Latch D con reloj.

Los Latch están hechos de compuertas, los Flip-Flops están hechos de Latches y los Flip-Flops son la base de los circuitos secuenciales (Que funcionan por pulsos de reloj) complejos. En la figura se muestra el esquemático de un flip-flop tipo D. Consta de dos Latch tipo D Maestro y Esclavo. Si se analiza su funcionamiento se puede resumir en una simple explicación. Y debido a esta también se le da el nombre de **registro**.



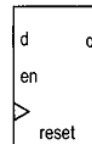
"El flip-flop tipo D muestra a la salida (Q) lo que había en la entrada (D) cuando hubo un flanco de reloj (CLK)"

Al Flip-Flop se le pueden agregar más características como:

-Funcionamiento en flanco descendente o ascendente

-Enable -Reset Síncrono o Asíncrono

Observe la tabla para ver el comportamiento del Enable y el Reset



| reset | clk | en | q* |
|-------|-----|----|----|
| 1 | - | - | 0 |
| 0 | 0 | - | q |
| 0 | 1 | - | q |
| 0 | f | 0 | q |
| 0 | f | 1 | d |

Bibliografía: HARRIS, David M., HARRIS, Sarah. Digital Design and Computer Architecture.

Flip-Flop tipo D en Verilog:

(2 Formas de DESCRIBIR lo mismo)

```
module flipflop(D,Q,CLK);  
input D;  
input CLK;  
output reg Q;
```

```
    always @ (posedge CLK)  
        Q<=D;  
endmodule
```

```
module flipflop(input D, output OUT, input CLK);  
  
    reg Q;  
  
    always @ (posedge CLK)  
        Q<=D;  
  
    assign OUT=Q;  
  
endmodule
```

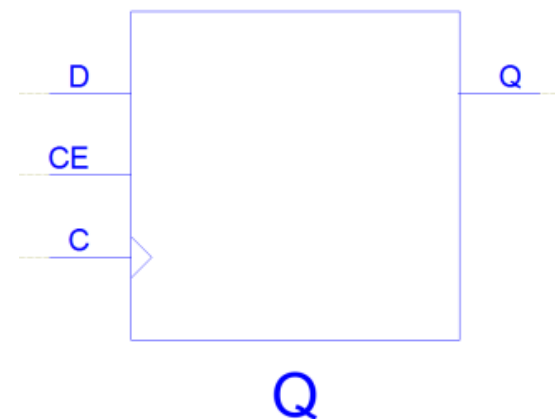
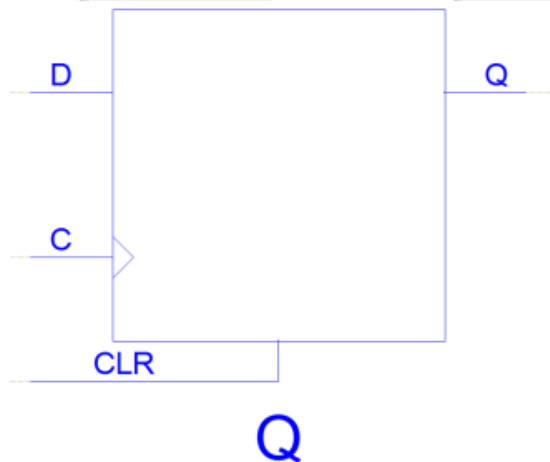
Para el Flip-Flop tipo D en Verilog con flanco descendente se utiliza "negedge"

Flip-Flop D y Reset asíncrono (no depende del reloj)

```
module flip_flop_arst(D,Q,CLK,Reset);  
input D;  
input CLK;  
input Reset;  
output reg Q;  
  
always @ (posedge CLK, posedge Reset)  
    if (Reset)  
        Q<=1'b0;  
    else  
        Q<=D;  
endmodule
```

Flip-Flop D y Enable Síncrono

```
module flip_flop_en(D,Q,CLK,Enable);  
input D;  
input CLK;  
input Enable;  
output reg Q;  
  
always @ (posedge CLK)  
    if (Enable)  
        Q<=D;  
    else  
        Q<=Q;  
endmodule
```



NOTA: También es posible y muy útil hacer múltiples Registros con buses.

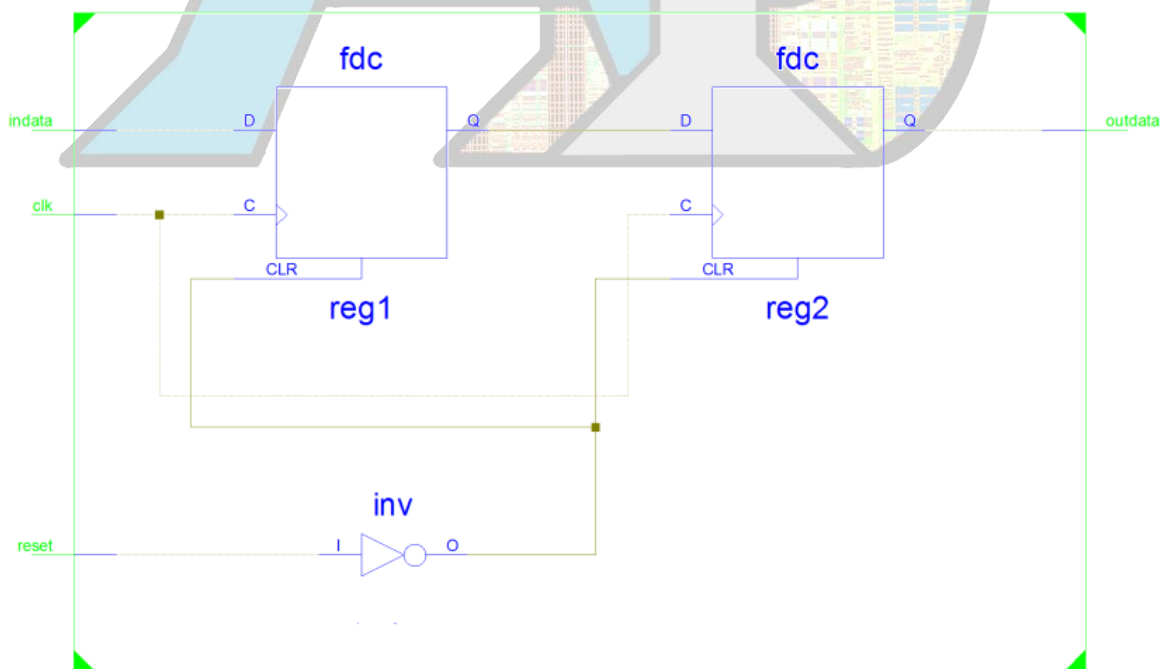
Por ejemplo:

```
module flipflop(D,Q,CLK);  
input [3:0]D;  
input CLK;  
output reg [3:0] Q;  
  
    always @ (posedge CLK)  
        Q<=D;  
  
endmodule
```

Ejemplos de circuitos basados en Flip-Flops

A este modulo se le llama "Sincronizador Doble" como su nombre lo dice sirve para sincronizar una entrada como un pulsador o switch de manera que se evite que la señal se lea en un estado que no es 0 ni 1 (Ej: Una entrada debe pasar de 0V a 3.3V al mover un switch, ese cambio no es inmediato) Para entenderlo mejor puede investigar sobre los tiempos de Setup (t_s) y Hold (t_h) de un pulso digital y sobre el término "Metaestabilidad".

ANTES DE PASAR A LA SIGUIENTE PÁGINA, INTENTE DESCRIBIR EL CIRCUITO EN VERILOG



Código Verilog

//Esta es una forma de describirlo, hay varias formas de describir el mismo circuito

```
module doublesync(indata, outdata, clk, reset);
input indata,clk,reset;
output outdata;

reg reg1, reg2;

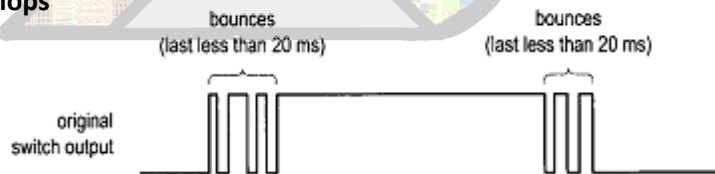
always @(posedge clk, negedge reset)
begin
    if (!reset)      //reset negado (active low: activo bajo)
    begin
        reg1 <= 1'b0;
        reg2 <= 1'b0;
    end else
    begin
        reg1 <= indata;
        reg2 <= reg1;
    end
end

assign outdata = reg2;

endmodule
```

Ejercicio de circuitos en base a Flip-Flops

Este es el problema:



Esta señal proviene de un pulsador (Ej:BTN_EAST, BTN_SOUTH) eso no es conveniente y menos si esa señal va al pin de reloj de un registro.

-A continuación se muestra un circuito propuesto denominado “atrapa-flancos con reset automático” Primero analice y entienda el funcionamiento del mismo. Si no entiende, PREGUNTE.

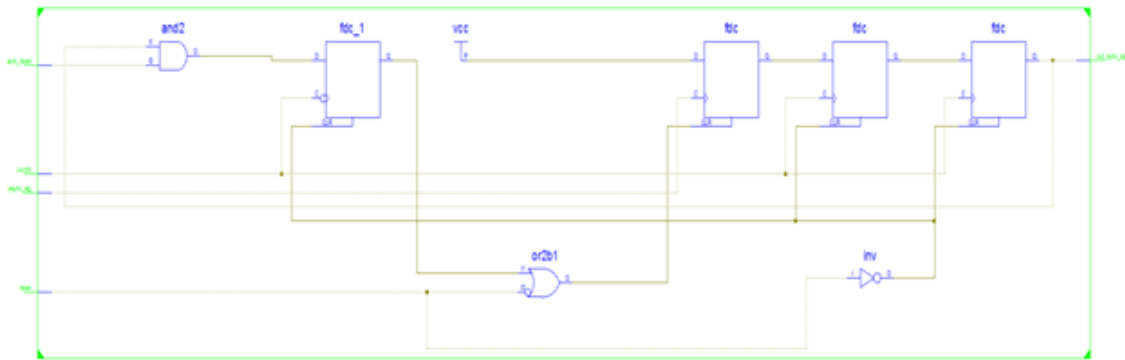
-Describalo en Verilog y simúlelo. “Para simular en Xilinx ISE 12.3 diríjase al video tutorial”

<http://www.youtube.com/watch?v=hmHKP0fi9V8>

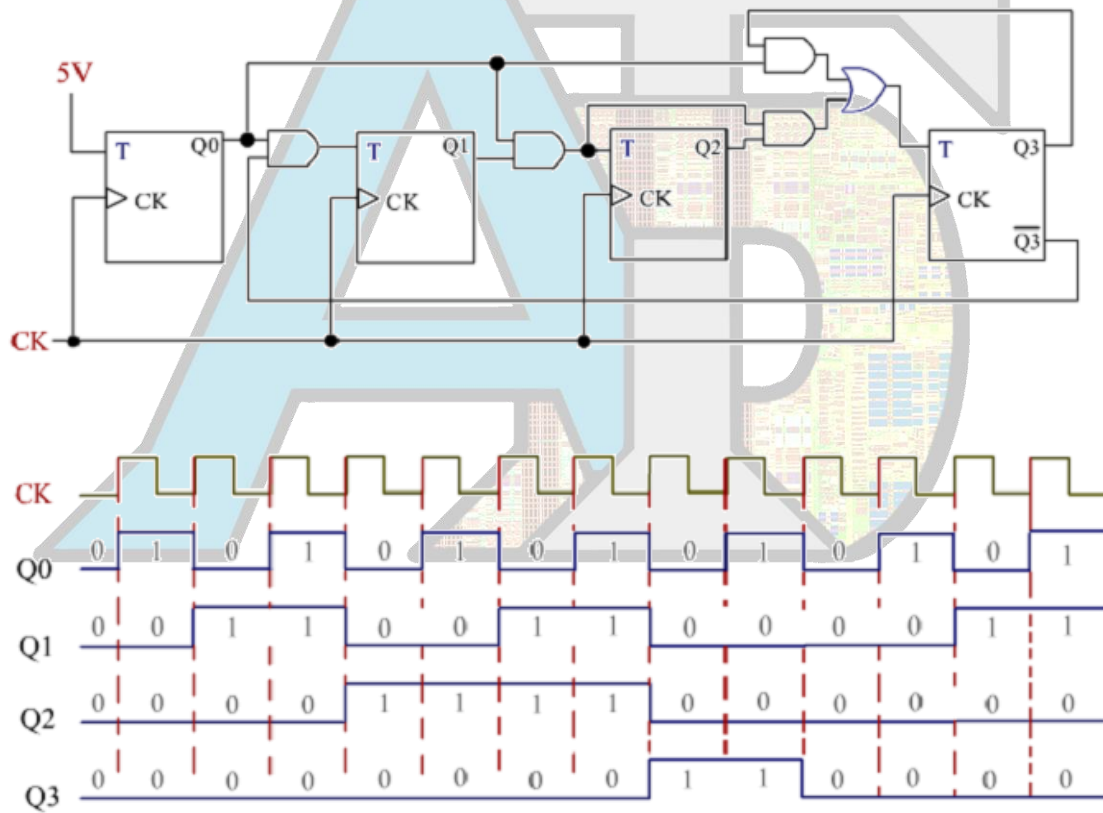
-¿Qué frecuencia de reloj debo usar para cumplir con las especificaciones de la gráfica anterior?

-Proponga otro circuito.*opcional

Atrapa-flancos con reset automático



Contador de décadas síncrono



NOTA: Observe que los flip-flops son tipo T no tipo D, los flip-flop tipo T muestran en Q el Q anterior negado($\sim Q$), si T es 1, de lo contrario conserva el valor de Q.