

Semillero ADT: Circuitos basados en Flip-Flops 2ª Parte

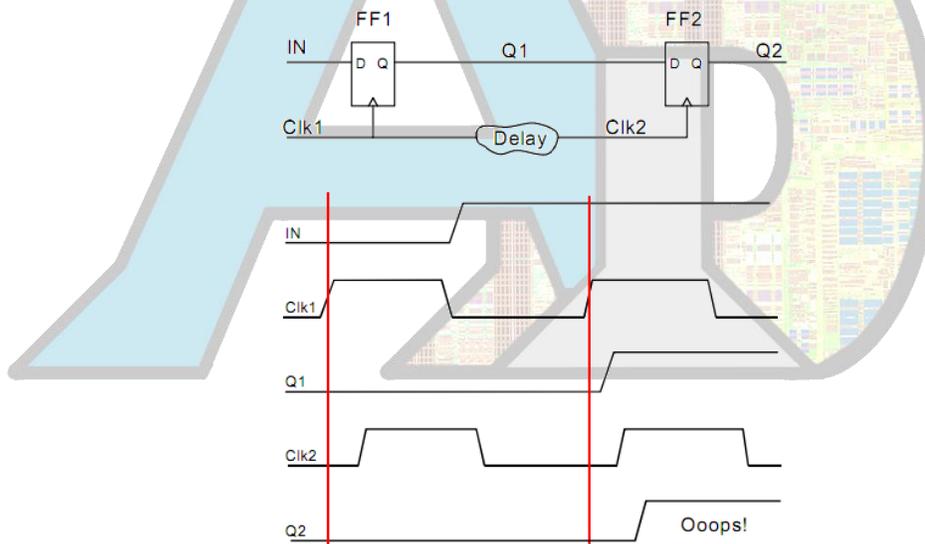
Objetivos:

Aplicar los conocimientos sobre circuitos basados en Flip-Flops básicos para usarlo en la sincronización de señales en distintos dominios de reloj.

Marco de Referencia:

La sincronización de señales con distinto dominio de reloj es un área de trabajo en el diseño de circuitos VLSI (Very Large Scale Integration) que necesita de un buen entendimiento y trabajo para su buen funcionamiento. Tome por ejemplo su PC, la tarjeta de video, los buses seriales, las memorias RAM, los dispositivos de almacenamiento; todos tienen relojes distintos y sin embargo todos deben converger a un dominio de frecuencia único en el procesador. En esta guía se hará una introducción a los problemas que se encuentran y los distintos circuitos que se utilizan para lograr esto.

El primero problema a mencionar es el "Clock Skew", el cual es el fenómeno que se da cuando una señal de reloj tarda un tiempo de skew (t_{skew}) en llegar a diferentes dispositivos del circuito, por ejemplo:



Fuente: Lulea University of Technology. Computation Structures Lecture 4

<http://www.sm.luth.se/csee/courses/smd/>

Lamentablemente, para resolver este problema las únicas soluciones eficientes son el buen diseño de enrutamiento y el uso de relojes globales. Los relojes globales tienen aproximadamente la misma distancia desde donde se genera hasta todos los dispositivos que lo usen. (La serie FPGA Spartan 3 tiene 8 posibles relojes Globales, "posibles", porque también se pueden conectar señales externas a los buffers globales). Para usar un buffer global con un reloj, use la siguiente instrucción:

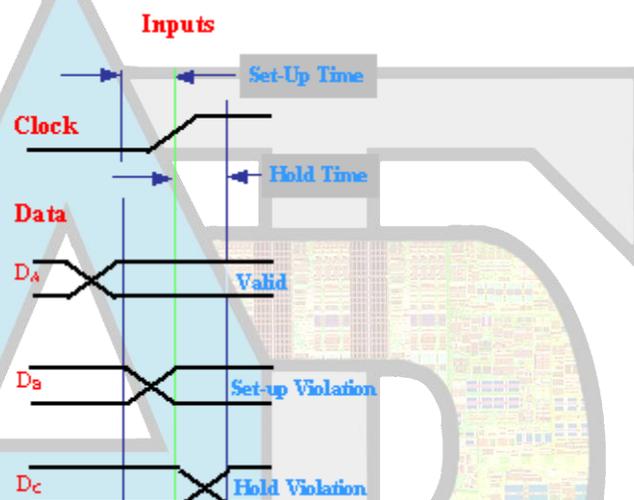
```
BUFG CLK_BUFFER_NAME(.I(CLK_NAME),.O(GLOBAL_CLK_NAME));
```

Para más información puede consultar:

www.cse.psu.edu/~eachempa/CSE478/slides/21synch_des_tech_8new.ppt

Otra solución posible es negar el reloj entre cada registro, considere si su diseño se puede valer de este tipo de solución.

Ahora, el problema principal a profundizar es la Metaestabilidad y para entenderla primero se deben tener claros los conceptos de tiempo de set-up(th) y de hold(th), para esto observe la siguiente imagen:

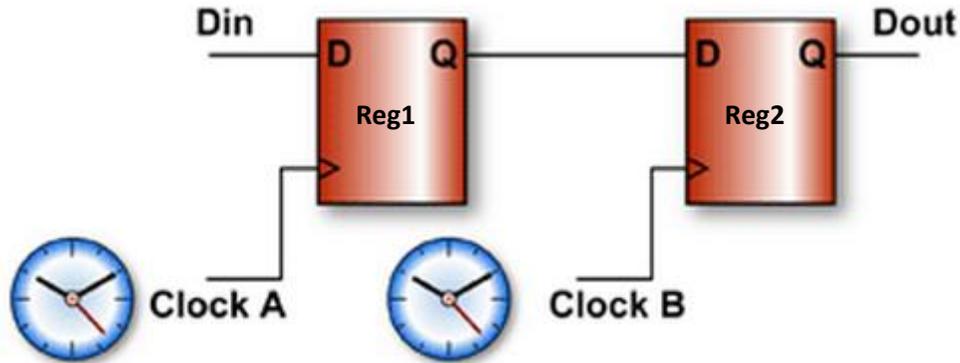


Fuente: http://www.interfacebus.com/Design_MetaStable.html

Siempre que exista una violación de los tiempos de set-up o hold, habrá metaestabilidad, es decir, el circuito actuará de una manera inesperada, ya sea: mostrando el valor contrario al deseado, en algunos casos mostrará el valor deseado, en otros casos puede mostrar un valor fuera de las especificaciones de funcionamiento (por ejemplo oscilaciones) o tendrá un retardo.

Cuando una señal es asíncrona y queremos sincronizarla, simplemente usamos un circuito como el sincronizador doble analizado previamente o inclusive con más etapas (Multi-Stage Synchronizer).

Pero si tenemos un bus de datos que trabajan a una frecuencia de un reloj **A** y necesitamos usarlo en un circuito que trabaja con un reloj **B** que sea o no de otra frecuencia, necesitamos circuitos más elaborados para su sincronización entre estos dos dominios de reloj.



Fuente: http://www.edn.com/article/510269-Designing_for_EOL_with_an_ASIC.php

Para esto existen diferentes casos, analícelos cuidadosamente:

I Caso: $\text{Clk_B} = \text{Clk_A}$ (son el mismo)

Este caso es seguro, siempre y cuando se use la restricción PERIOD en el UCF. Esta restricción la encuentra en el Datasheet de la tarjeta Spartan 3ª junto a otras restricciones:

```
NET "CLK_50MHZ" LOC = "E12" | IOSTANDARD = LVCMOS33;
NET "CLK_50MHZ" PERIOD = 20.0ns HIGH 40%;
```

Si quisiera agregarle la restricción PERIOD al reloj de 133.33MHz ¿Qué debería agregarle al UCF?

II Caso: $\text{Clk_B} = \sim\text{Clk_A}$ (es el inverso)

Este caso es seguro, siempre y cuando se use la restricción PERIOD en el UCF.

III Caso: $\text{Clk_B} = \text{División de Clk_A}$

Este caso NUNCA es seguro, tenga en cuenta que hay un retardo en la generación del nuevo reloj Clk_X a partir de Clk_Y que lo deja en las condiciones ideales para violar los tiempos de setup y hold.

IV Caso: $\text{Clk_B} = \text{División de } \sim\text{Clk_A}$

Solo es seguro si los relojes son significativamente más lentos que la capacidad del dispositivo.

Spartan 3A → 250MHz

Pero, generalmente no es seguro.

V Caso: $\text{Clk_B} = \text{Clk_A}$ (son el mismo, pero están conectados al dispositivo en pines distintos)

Nunca es seguro ya que el mapeo de la lógica en el dispositivo nunca tiene en cuenta los caminos externos a este.

NOTA: No es el caso de los relojes del Starter Kit Spartan 3A.

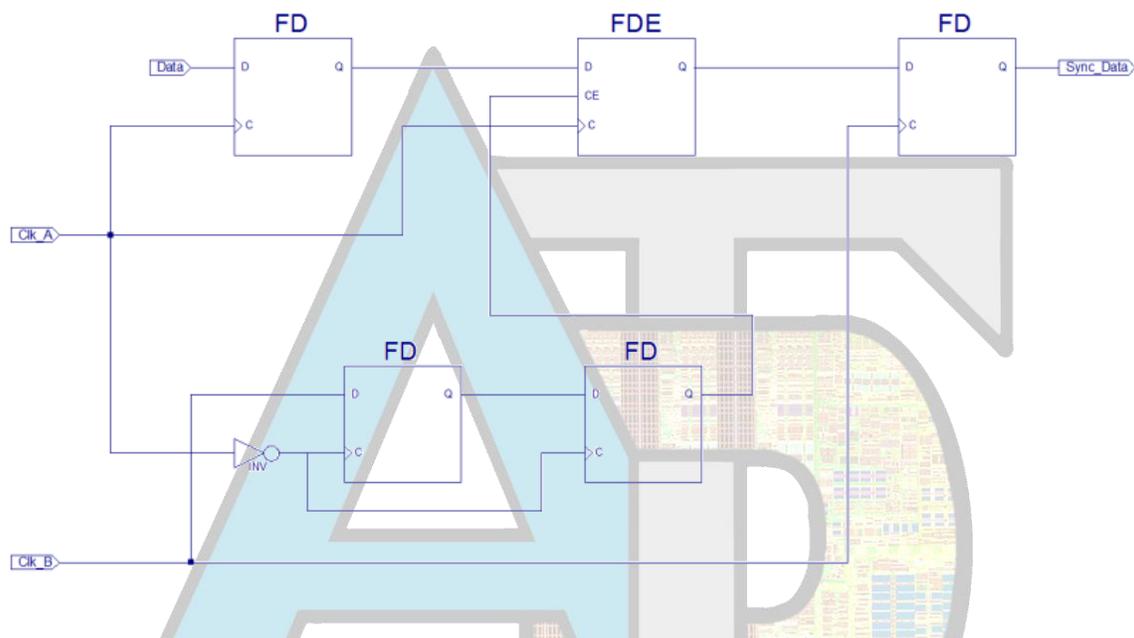
VI Caso: $F_{\text{Clk}_B} \lll F_{\text{Clk}_A}$ (No están relacionados entre sí o no es la división del otro)

Nunca es seguro ya que no se puede asegurar que no haya violaciones de los tiempos de setup y hold.

¿Cómo hacer para que en el Registro 2 (Reg2) siempre haya datos validos?

Sí $F_{\text{Clk}_B} \lll F_{\text{Clk}_A}$

(Clk_A es mucho MÁS RÁPIDO que Clk_B, no importa si Clk_B es división de Clk_A)



NOTA: Data y Sync_Data pueden ser buses (Ej: [127:0])

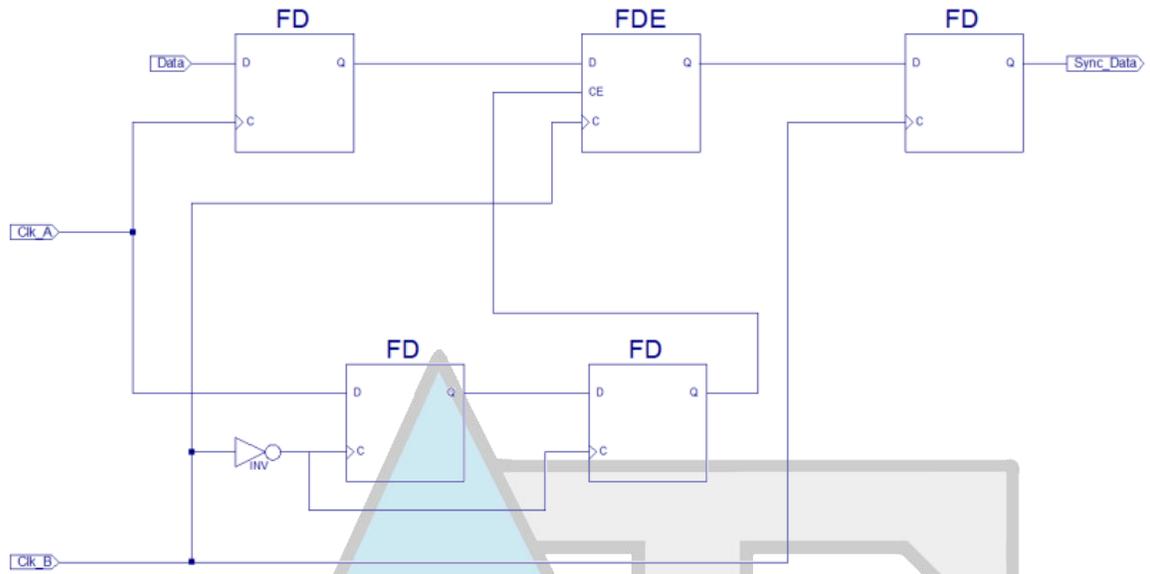
Analice el funcionamiento del circuito
¿Qué deficiencia tiene?

Describa este circuito en Verilog.

Tenga en cuenta que el registro FDE tiene un Enable, es decir, en el caso que exista un flanco ascendente para ese registro, la salida solo cambiará de acuerdo a la entrada si ese valor está en 1.

Sí $F_{Clk_B} \gg F_{Clk_A}$

(Clk_A es mucho MÁS LENTO que Clk_B, no importa si Clk_A es división de Clk_B)



NOTA: Data y Sync_Data pueden ser buses (Ej: [127:0])

Analice el funcionamiento del circuito

¿Qué deficiencia tiene?

Describa este circuito en Verilog.