



Diseño de Circuitos Síncronos III

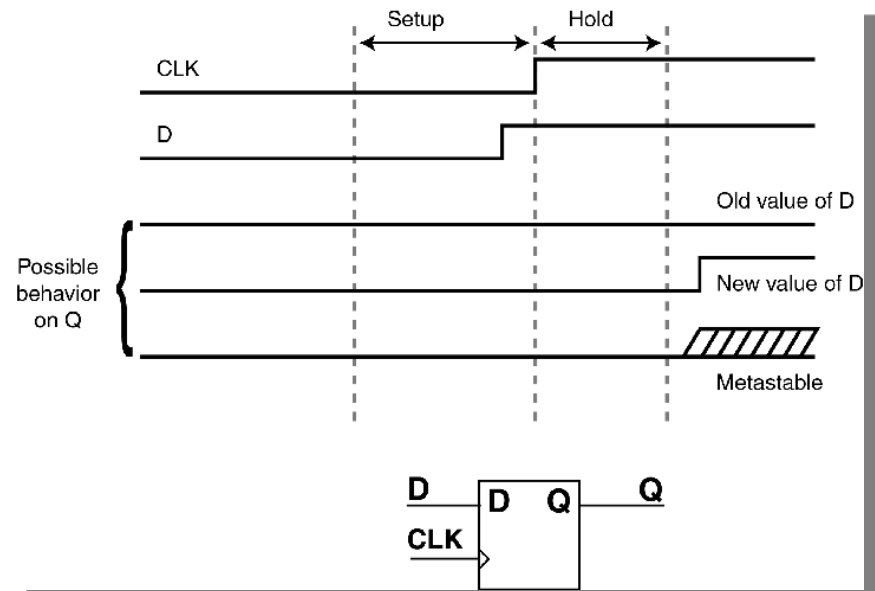
Jhon Jairo Padilla Aguilar

Metaestabilidad

- Cuando se muestrea un dato que cambia utilizando una señal de reloj, el orden de los eventos determina el resultado.
- Cuando dos eventos ocurren muy cerca en el tiempo, violando los tiempos de establecimiento y retención, ocurre una falla de sincronización.

Metaestabilidad

- La señal de reloj ocurre en un momento en que no se cumplen los tiempos de establecimiento y retención
- la salida tiene 3 posibles valores:
 - El antiguo valor de D
 - El nuevo valor de D
 - Un valor metaestable (que sólo permanece un tiempo muy corto) y no se sabrá el resultado final con certeza



Metaestabilidad

Curva de Energía de la Metaestabilidad

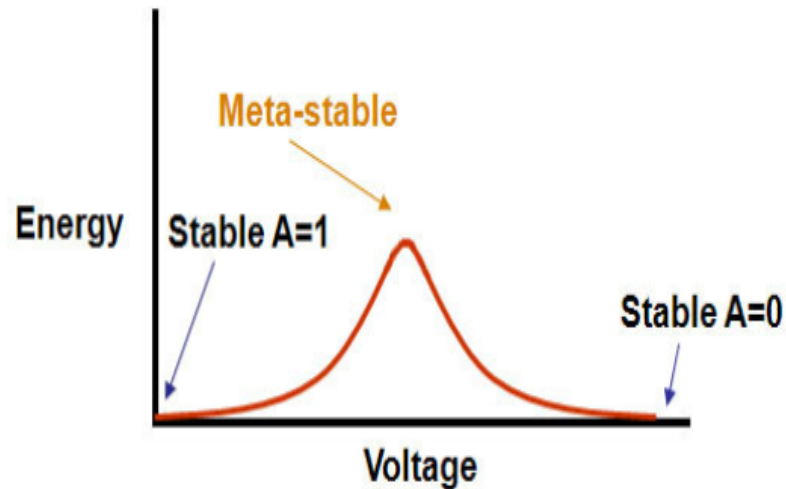
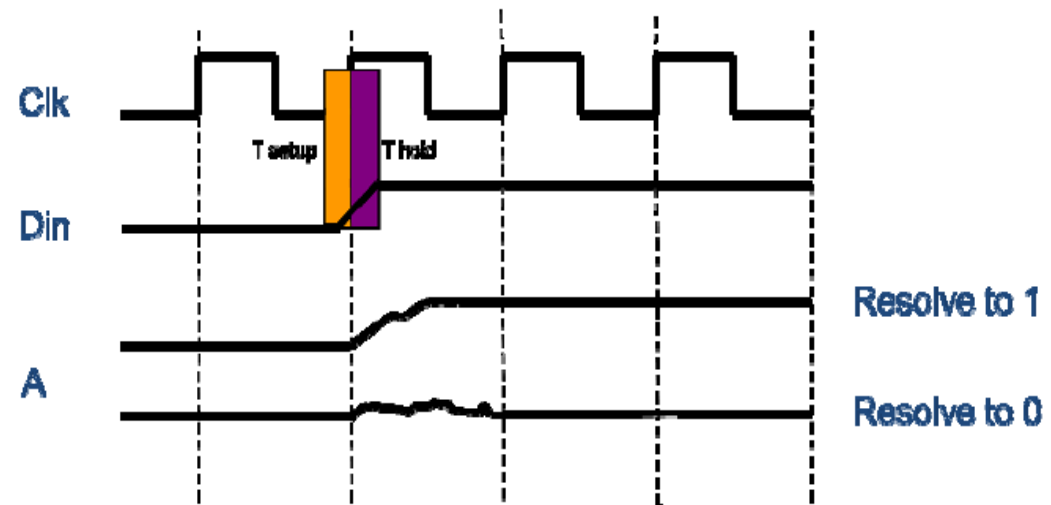
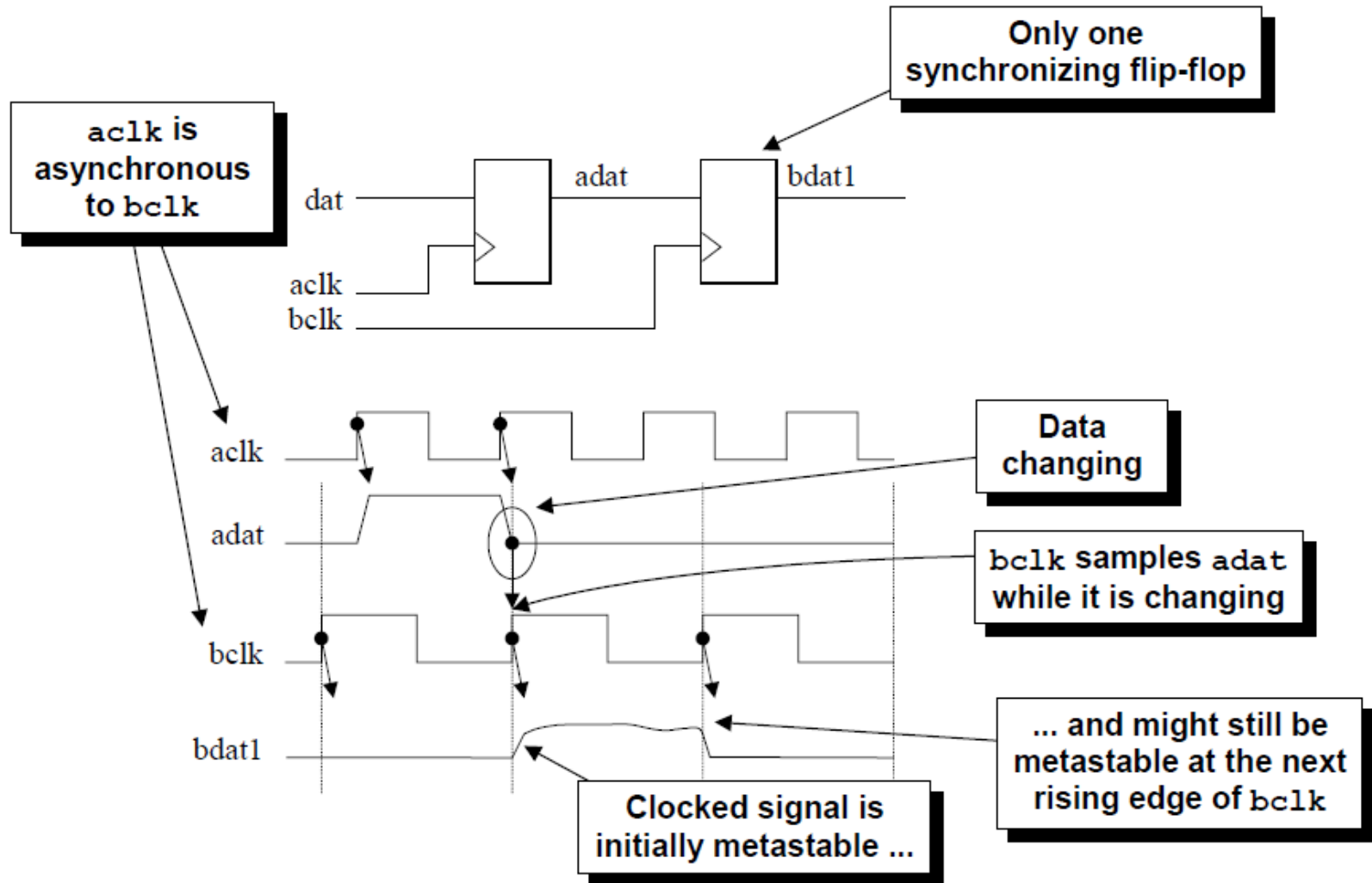


Diagrama de tiempos de la metaestabilidad



Ejemplo de metaestabilidad: Relojes asíncronos

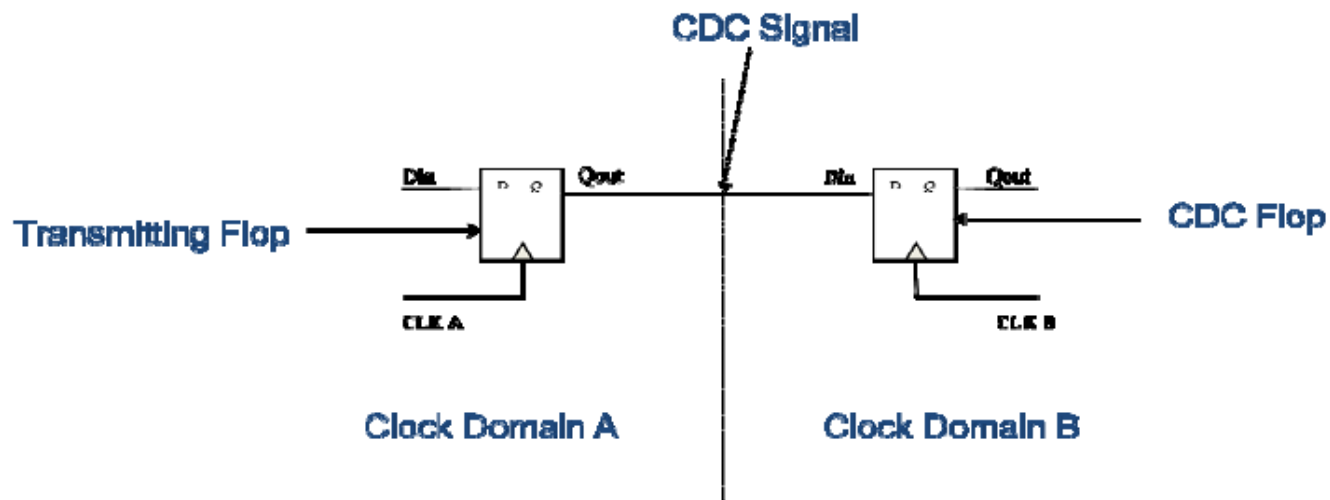


Limitaciones de las herramientas de verificación funcionales

- La verificación funcional de los circuitos está basada en simulaciones y en análisis teórico de los diagramas de tiempo
- Características ideales de estas herramientas:
 - Retardo de las señales igual a cero
 - Retardo cero en la respuesta de las compuertas
 - Retardo unitario para los flip-flops
 - Comportamiento del reloj ideal

Clock Domains Crossing (CDC)

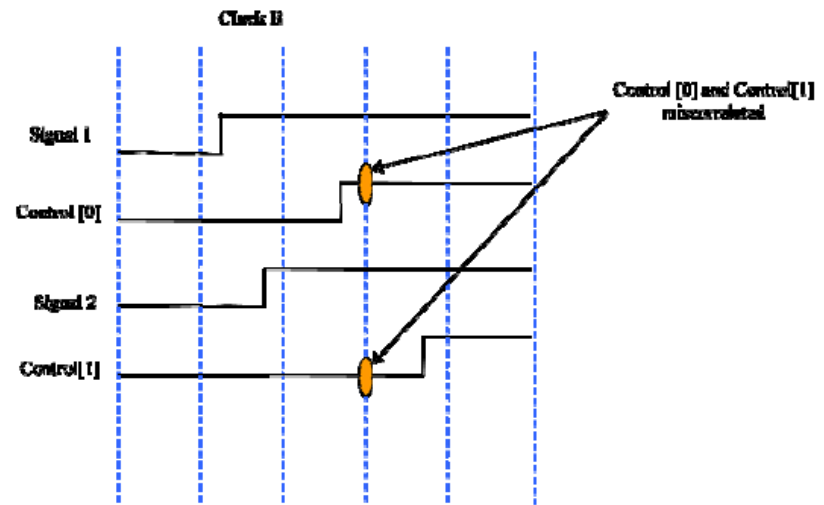
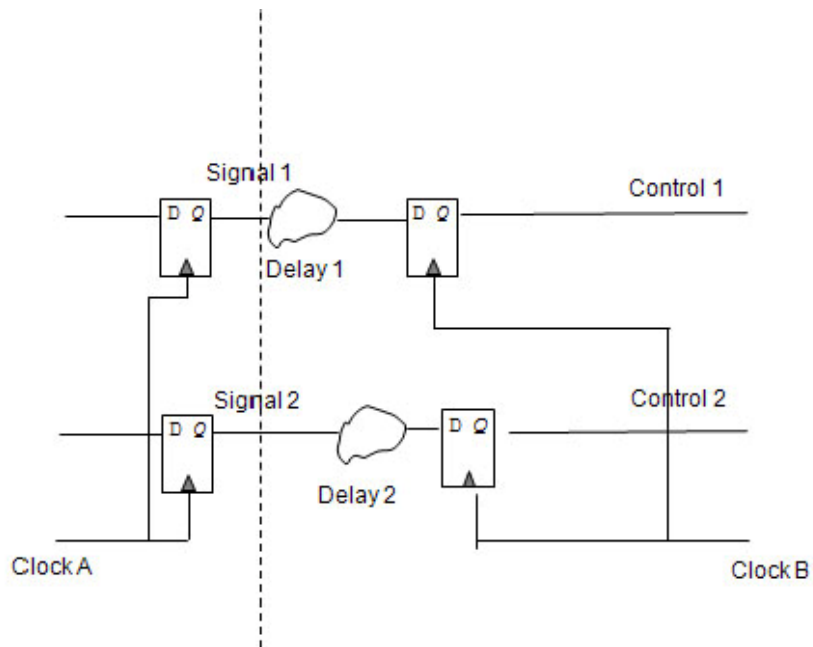
- Dominio de reloj:
 - Conjunto de flip-flops que son disparados por una misma señal de reloj
- Clock Domain Crossing:
 - Camino de flip-flops en que el Flip-Flop transmisor es disparado por un reloj que es asíncrono con respecto al reloj del flip-flop receptor.



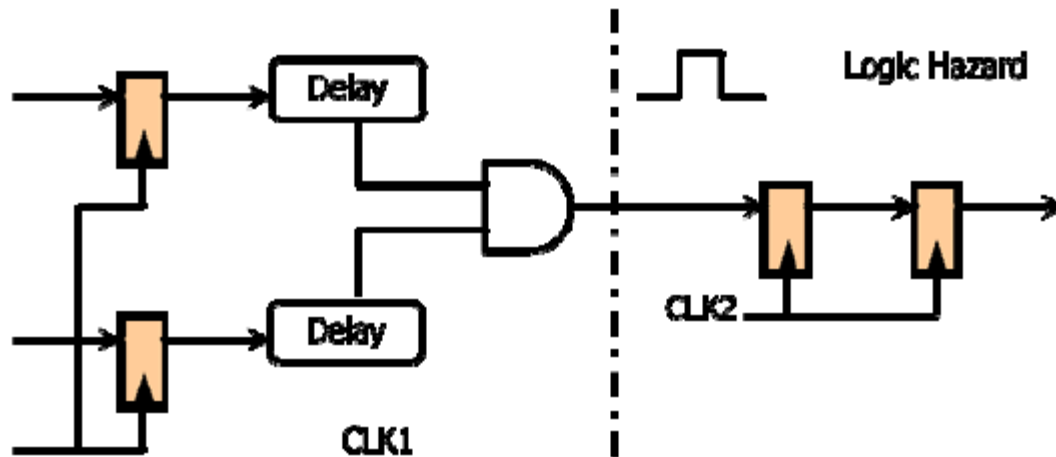
Errores debidos a CDC

- La metaestabilidad introduce los siguientes tipos de fallos en el diseño:
 - Pérdida de correlación
 - Captura de errores
 - Pérdida de señal
 - Propagación de la metaestabilidad
 - Problemas de Skew

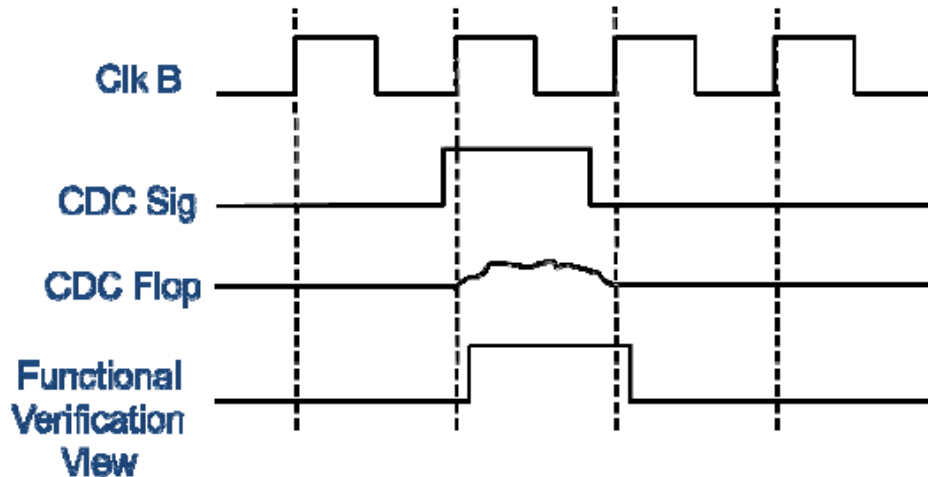
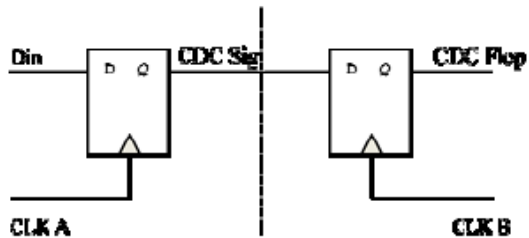
Pérdida de correlación



Captura de Errores (glitches)



Pérdida de señales

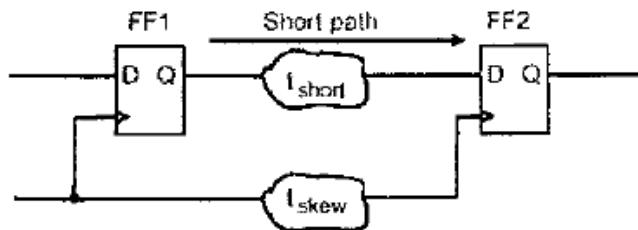
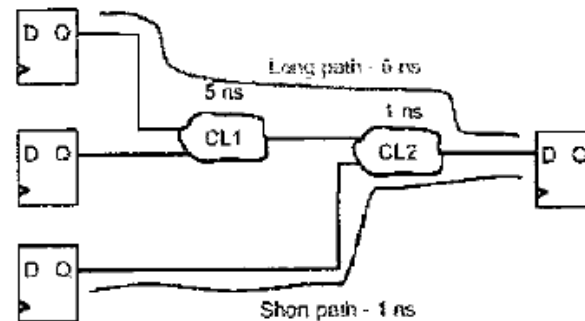


- Señales más cortas que un período de reloj podrían perderse, aunque en el análisis teórico sean capturadas.

El problema del Skew (diferencia en retardos)

Long path and short path

The long path or critical path of a design determines the maximum achievable clock frequency. But why do we have to worry about the short path?



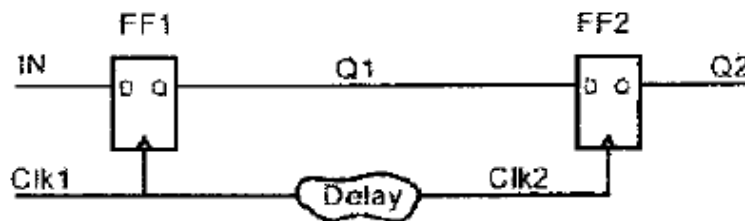
For proper operation the following must be satisfied:

$$t_{pFF1} + t_{short} > t_{skew} + t_{hFF2}$$

or else we will get hold time violations for FF2. As you can see the relationship does not depend on the clock period

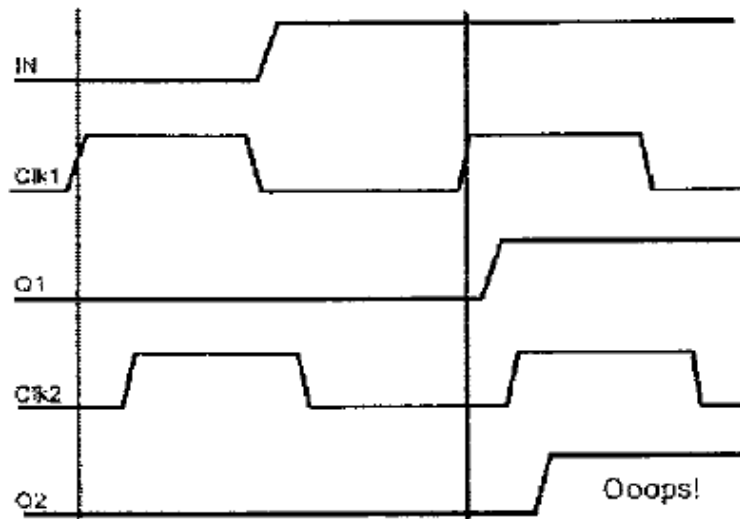
Diferentes retardos en reloj generan errores

Clock skew and race hazards



For proper function the following must be satisfied

$$t_{pFF(\min)} + t_{pLogic(\min)} - t_{hold} - t_{skew(\max)} > 0$$



The short path is important!

Consecuencias del skew

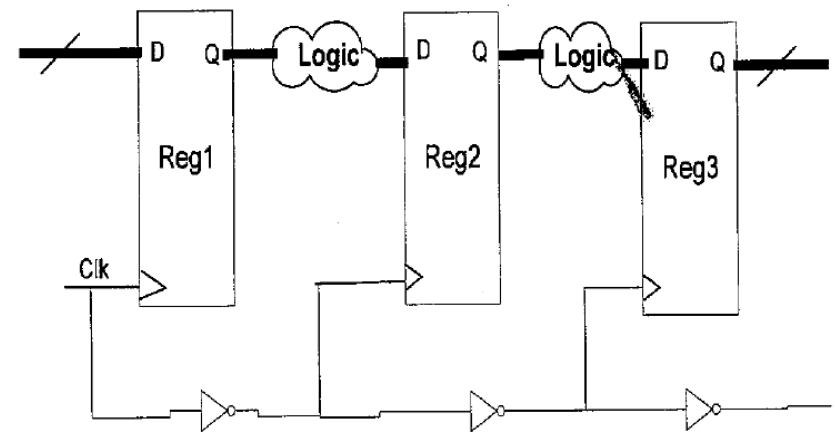
- El mayor problema aparece cuando se muestrean buses de flip-flops
- Si algunos pero no todos los flip-flops tienen problema de skew, se corrompen los datos
- Los problemas de skew no tienen una relación directa con la velocidad de reloj (pueden ocurrir con igual probabilidad con relojes lentos o rápidos)
- Los problemas de Skew dependen de la ruta del reloj. Pero la ruta cambia en cada compilación!!! (se vuelve como problemas de Voodoo!!!)

Reducción de problemas de Skew

- Utilizando señales de reloj globales que pueden manejarse por el usuario
- Se puede reducir un poco el problema de skew dando unas restricciones de tiempo precisas al software compilador:
 - período de reloj deseado
 - tiempos de llegada de las entradas con respecto al reloj
 - máximo retardo de las salidas respecto al reloj
- Habilitar la detección de skew en el software (variables del entorno `XILINX_DOSKEWCHECK=1`; `XILIINX_DORACECHECK=1`), pero igual se pasan por alto algunos problemas de skew.
- Este problema pasa especialmente para nuevos dispositivos en los que aún no se tiene una buena información de los retardos por enrutamiento

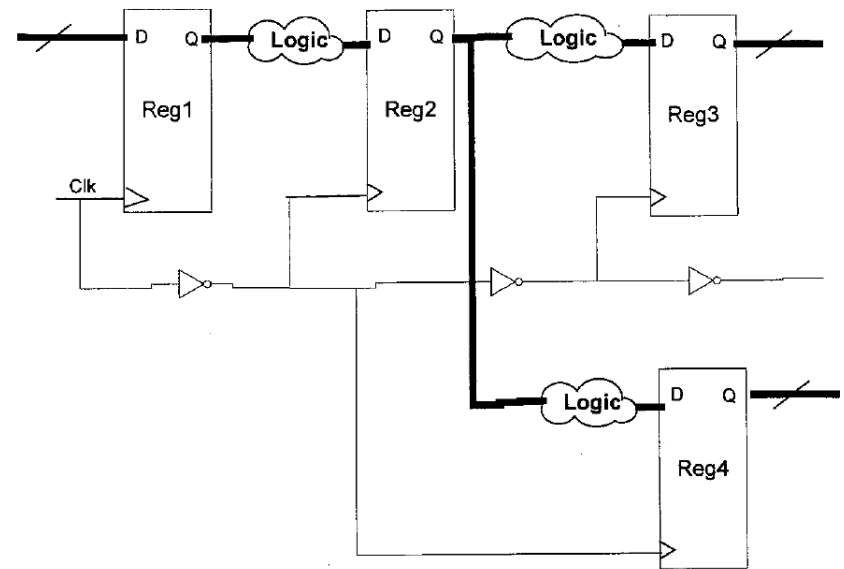
Reducción de problemas de Skew en el diseño

- Inversión de la señal de reloj entre cada par de flip-flops:
 - El problema es que la frecuencia efectiva de reloj se duplica (el período de reloj se divide en dos)



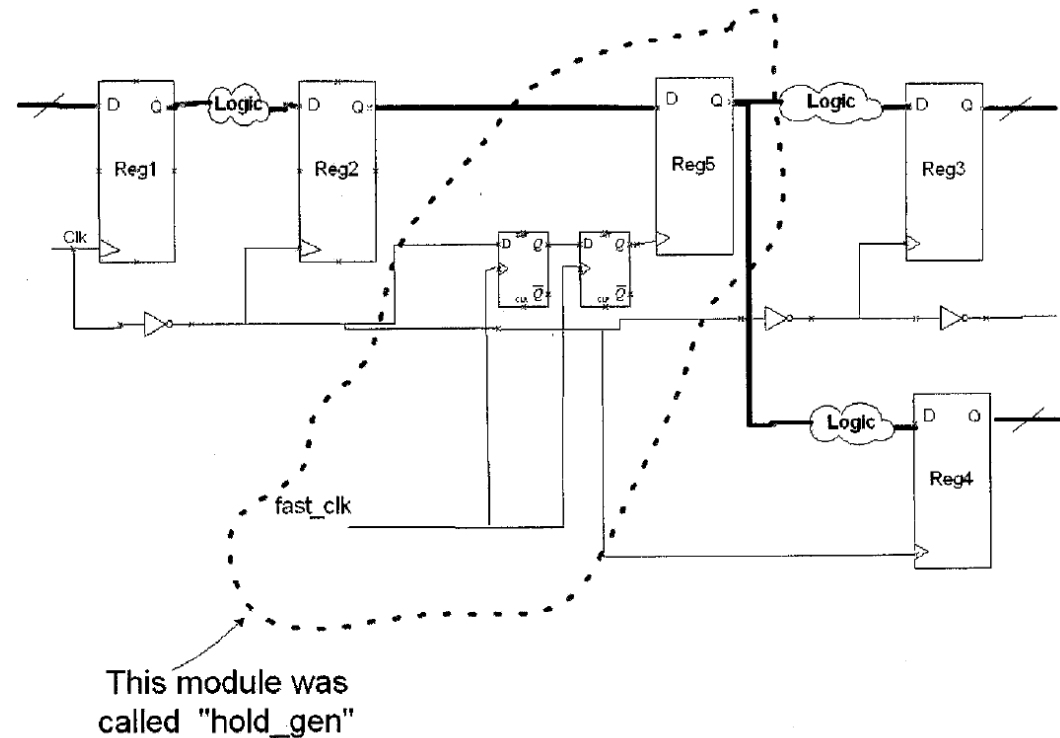
Situación típica con Skew

- En esta situación se violan los tiempos de establecimiento entre Reg2 y Reg4
- Supongamos que la topología no puede cambiarse



Posible solución

- Supone una señal de reloj lenta
- Se retarda la señal de reloj de reg2 por dos períodos de reloj de la señal fast_clk (tiene que ser mucho más rápida que el clk)

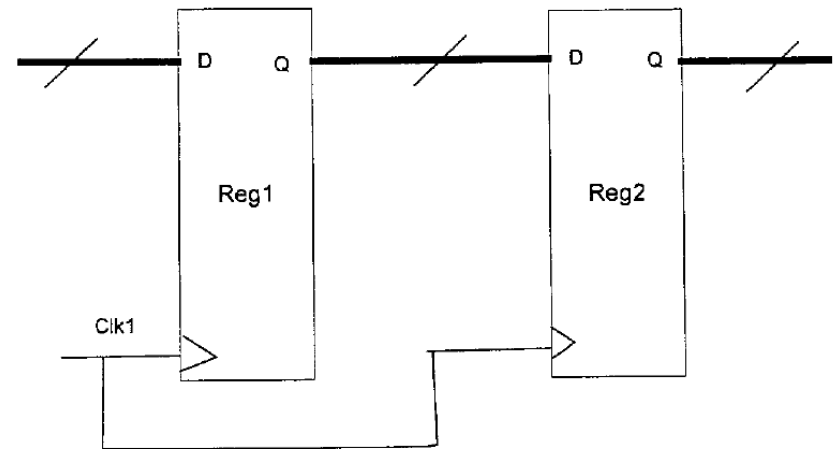


Algunas soluciones a los problemas de CDC

- Uso de sincronizadores para evitar metaestabilidad (**ya lo vimos** y funciona para señales de 1 bit)
- Transferencia de datos entre dominios de reloj utilizando protocolos de transferencia (handshake), tales como el protocolo Start-finish (**ya lo vimos**)
- Pero el problema es cuando se desea transferir buses de datos!!! (los FF cambian en diferentes momentos)

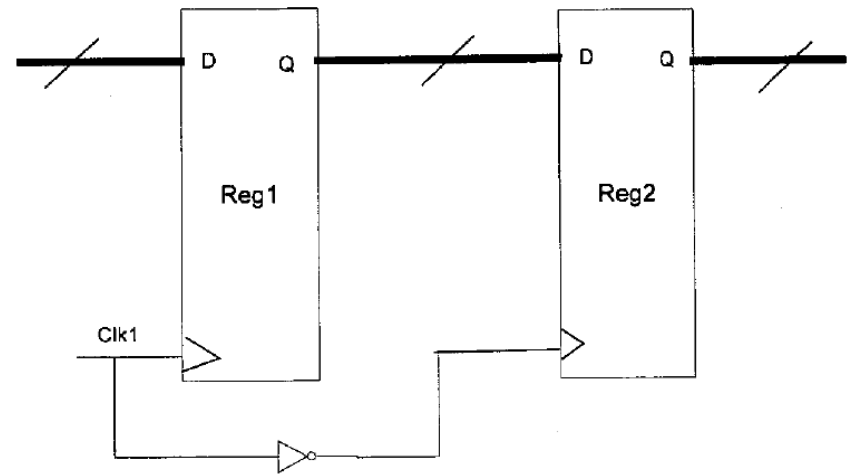
Situación: $\text{clk1}=\text{clk2}$

- Segura mientras se hayan especificado las restricciones de tiempo en los archivos ncf o ucf



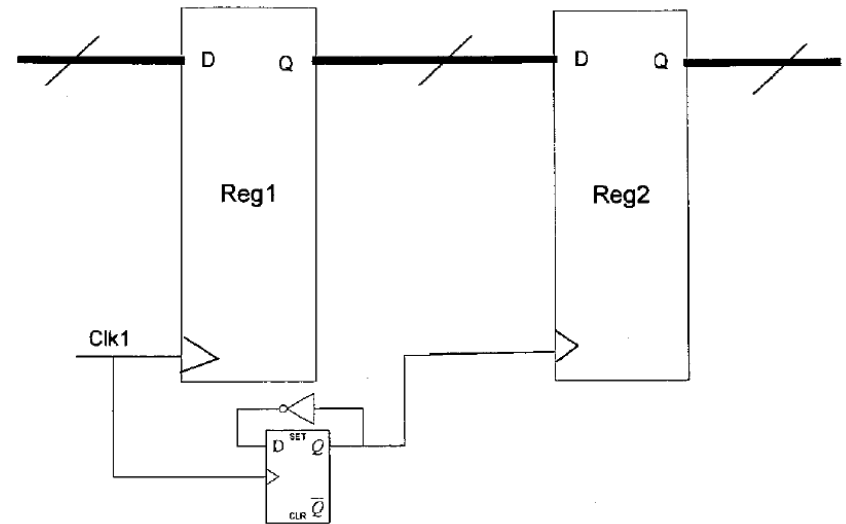
Situación: $\text{clk2} = \neg \text{clk1}$

- Seguro mientras se especifiquen las restricciones de clk1 en el archivo ncf o ucf
- Reg2 captura la señal medio período después



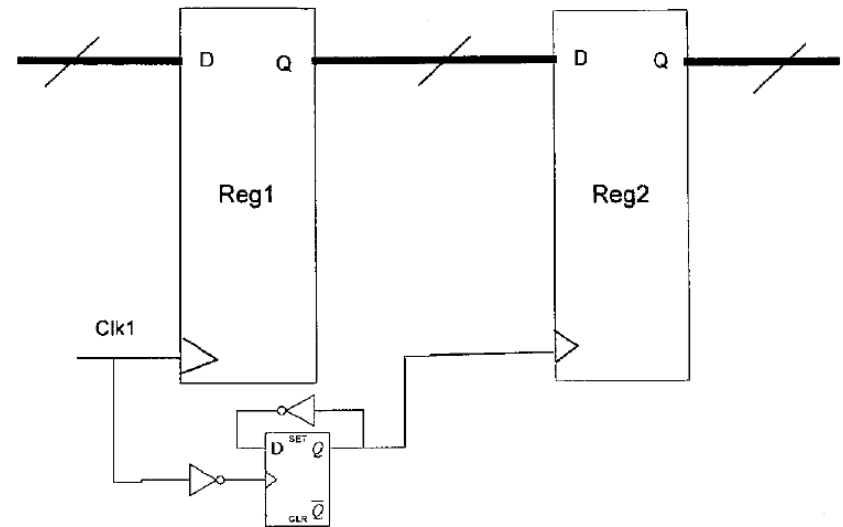
Clk2=division del clk1

- Nunca es seguro!!!
- El problema es debido al cambio del clk2 un tiempo pequeño después de dos ciclos de clk1
- Se violan los tiempos de Establecimiento y retención
- Esto se aplica a cualquier división del reloj



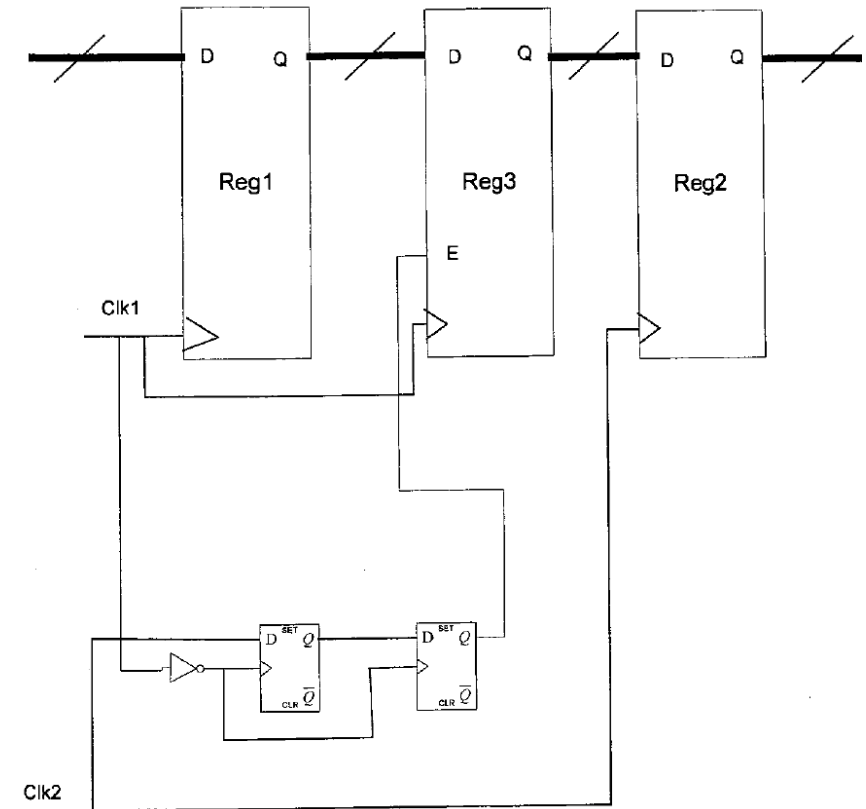
Clk2= division de !clk1

- Generalmente no es seguro!!!
- Aunque bajo ciertas circunstancias podría funcionar:
 - Si clk1 y clk2 son muy lentos con respecto a la capacidad del chip



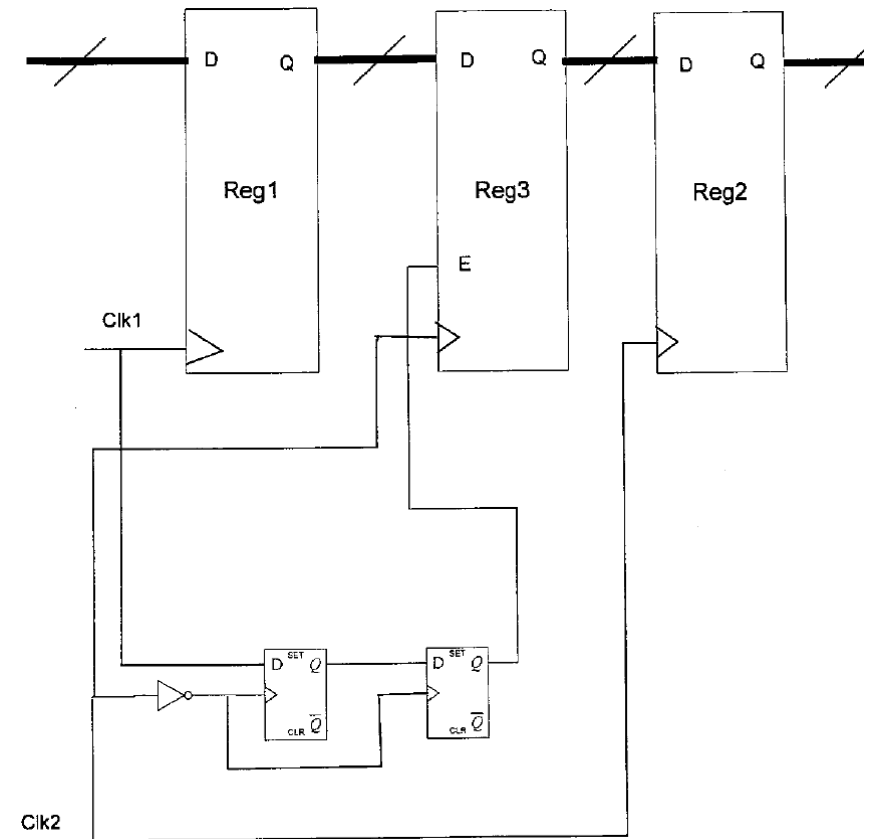
Clk2 mucho más lento que clk1

- Podría usarse una restricción FROM-TO entre Reg2 y Reg3 con un valor menor que la mitad de clk2
- Si clk2 es muy lento, podría no ser necesario en la práctica especificar las restricciones



Clk2 es significativamente más rápido que clk1

- Debe usarse una restricción FROM-TO desde Reg1 a Reg3, con un valor menor de 1.5 veces clk2 (preferiblemente menos de un período de clk2, para evitar problemas de skew)



Conclusiones

- Para decidir si una interconexión CDC es segura, pregúntese lo siguiente:
 - Puede el compilador calcular automáticamente el skew entre los relojes?
 - Qué restricciones de tiempo tales como PERIOD y FROM-TO, son necesarias para asegurar la confiabilidad
 - Cuáles son los escenarios extremos que podrían ocurrir bajo el circuito?
- Si existe el riesgo de ocurrir un error, este ocurrirá
- Usted debe planear bien las interfaces CDC o le ocurrirán problemas de VoDoo!!!