



# Retardo de enrutamiento y Fan-out

Jhon Jairo Padilla Aguilar

# Introducción

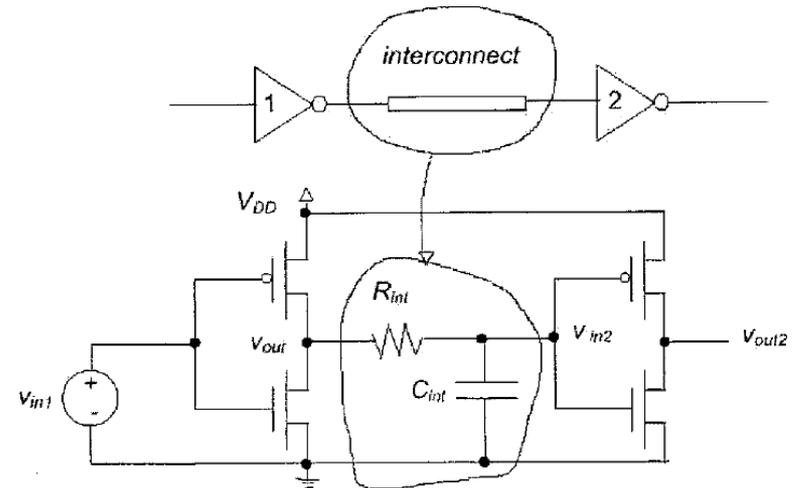
- El retardo de una señal dentro de un chip depende de:
  - La longitud de la ruta que sigue
  - El fan-out

# Efecto de la longitud de la ruta

- A mayor longitud de la ruta:
  - Mayor distancia a recorrer y por tanto mayor retardo
  - Crece la resistencia eléctrica
  - Aumentan las capacitancias parásitas

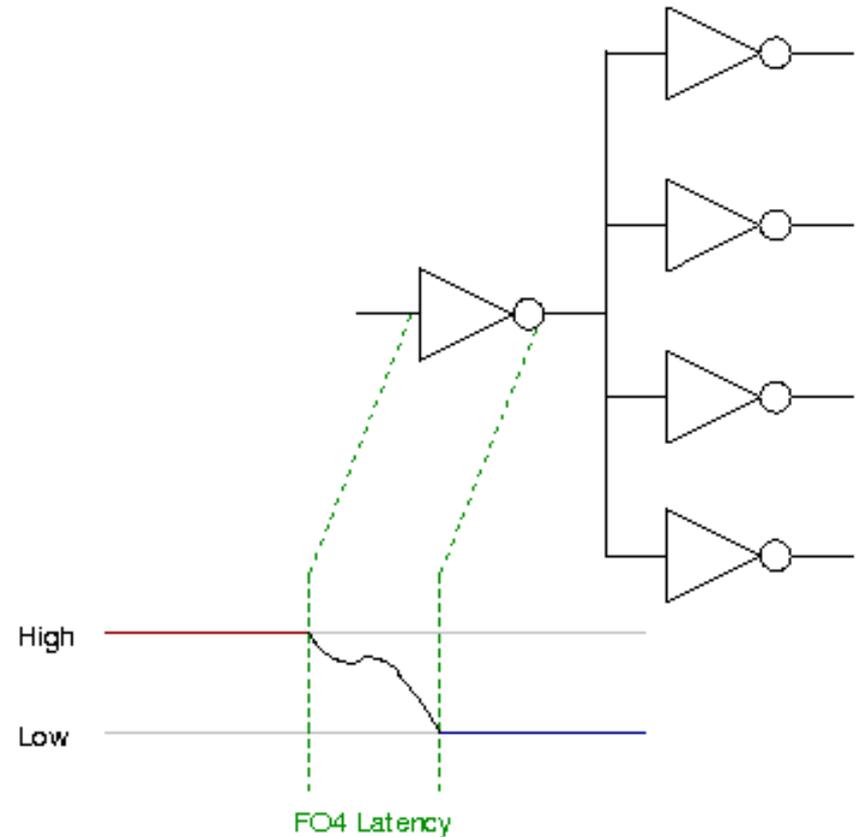
# Efecto de la longitud de la ruta

- A mayor longitud, aumenta la capacitancia y por tanto la respuesta de la línea se vuelve más lenta (filtro pasa-bajo). Esto incrementa el retardo de las señales en responder a los cambios.



# Efecto del Fan-out

- **Fan-out:** Número de Cargas de destino que se pueden conectar a una salida lógica sin alterar demasiado el tiempo de respuesta
- A mayor número de compuertas, las capacitancias de las entradas se suman en paralelo
- La salida que las alimenta verá una capacitancia más grande, lo que aumenta el retardo en la respuesta de las entradas que alimenta.



# Objetivos de diseño

- En consecuencia, deben tenerse como objetivos de diseño:
  - Reducir la longitud de la ruta (lo hace la herramienta Place & route del software de síntesis)
  - Reducir el Fan-Out (se hace mediante el diseño mismo y optimizaciones realizadas por el sintetizador).

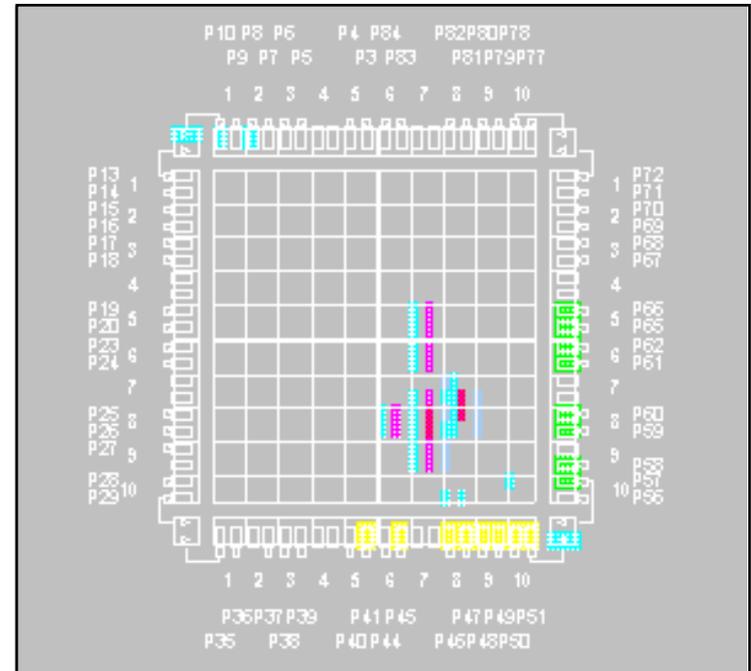
# Restricciones de tiempo y su proyecto

Qué efecto tienen las restricciones de tiempo en su proyecto?

- Las herramientas de implementación no intentan encontrar las rutas que obtienen mayor velocidad
  - En lugar de esto, las herramientas de implementación tratan de alcanzar sus expectativas de rendimiento
- Las expectativas de rendimiento se especifican mediante las restricciones de tiempo
  - Las restricciones de tiempo mejoran el rendimiento del diseño ubicando la lógica junta y agrupada, por lo que pueden usarse recursos de enrutamiento más cortos.
  - El editor de Restricciones se refiere al Editor de restricciones de Xilinx.

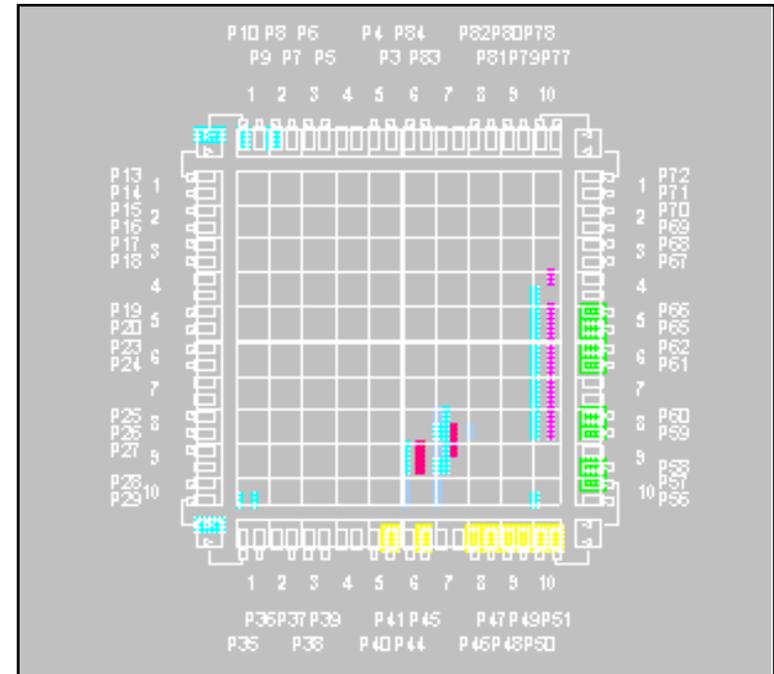
# Sin restricciones de tiempo

- Este diseño no tiene restricciones de tiempo o asignación de pines
- Note la estructura lógica de la ubicación de rutas y los pines.



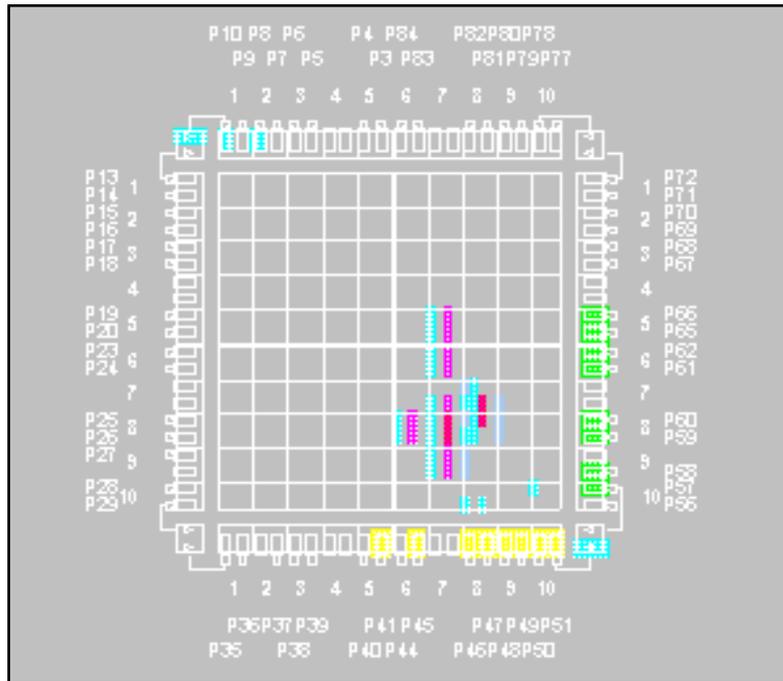
# Con Restricciones de tiempo

- Este es el mismo diseño con restricciones de tiempo
- Note que la lógica se ubica cerca a los pines de entrada/Salida
  - En este diseño no se usaron FFs en los pines I/O
  - Mover la lógica cerca a los pines mejora la temporización interna y externa del chip

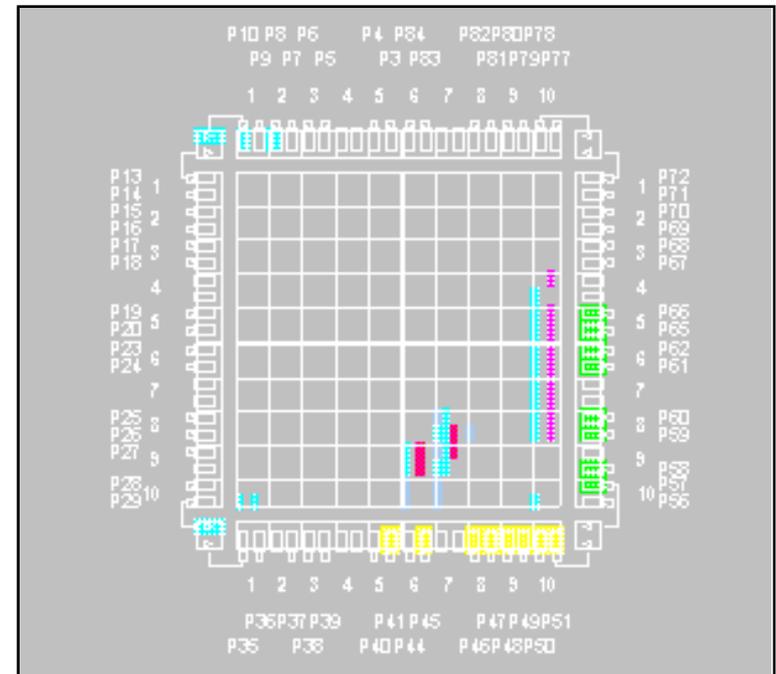


# Cambio de las rutas al usar restricciones globales

- Sin restricciones de tiempo globales

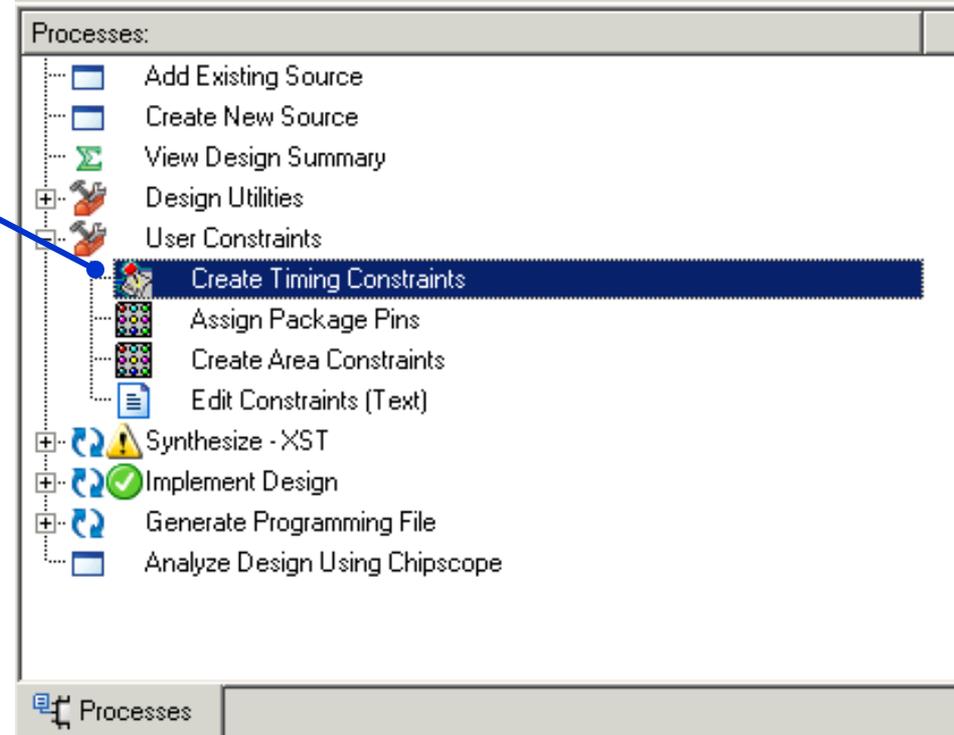


- Con restricciones de tiempo globales



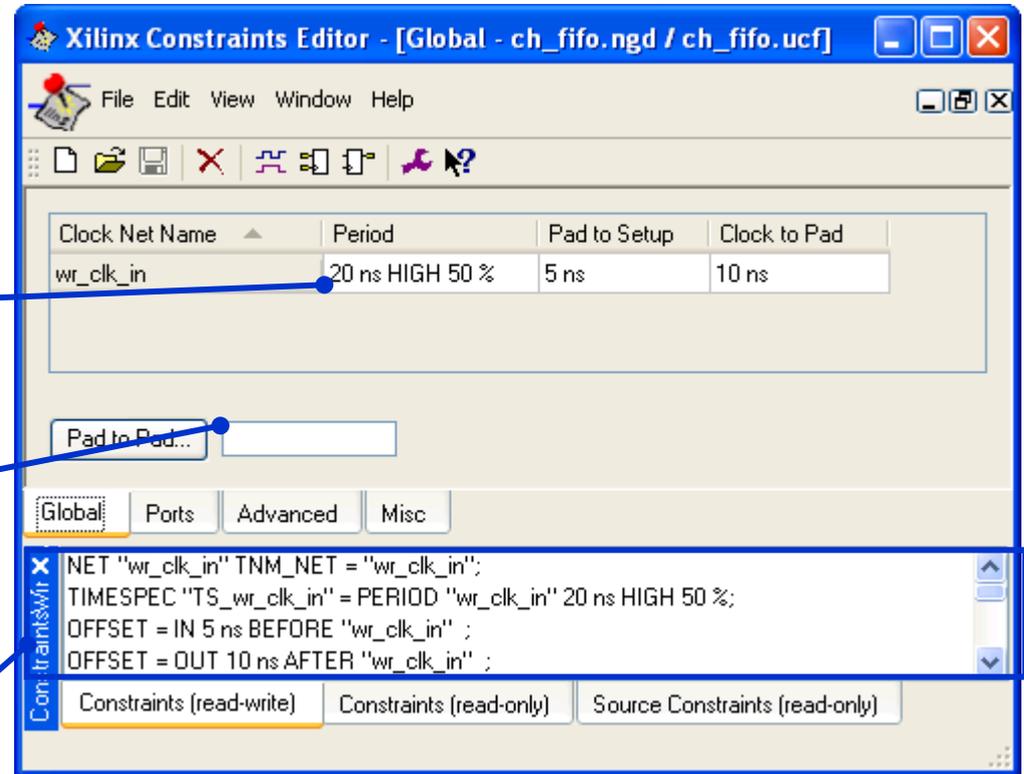
# Editor de Restricciones en Xilinx

- Expanda **User Constraints** en la pestaña de procesos en la ventana Source
- Haga Doble-click en **Create Timing Constraints**



# Ingresar las restricciones: PERIOD y Pad-to-Pad

- Las restricciones PERIOD y Pad-to-Pad pueden ingresarse en Global tab
- Haga Doble-click aquí para hacer una restricción de período
- Restricción Global Pad-to-Pad
- Las restricciones pueden ser borradas seleccionando la restricción en la ventana de texto y presionando **<Delete>**



# PERIOD Constraint Options

- **TIMESPEC Name**
- Valor de restricción específico
  - Active clock edge
  - Duty cycle
- **Relative to other PERIOD TIMESPEC**
  - Útil para diseños con múltiples señales de reloj
  - Puede definir relaciones de frecuencia y fase
- **Input Jitter**

**Clock Period**

Initial active edge used for OFFSET value is set to HIGH

PERIOD

INPUT\_JITTER

TIMESPEC Name: TS\_wr\_clk\_in

Clock Net Name: wr\_clk\_in

Clock Signal Definition

SpecifyTime

Time: 20 Units: ns

Start HIGH  Start LOW

Time HIGH: 50 Units: %

Relative to other PERIOD TIMESPEC

Reference TIMESPEC:

Multiply by  Divide by

Factor: 1

PHASE:

Plus  Minus

Value: 0 Units: ns

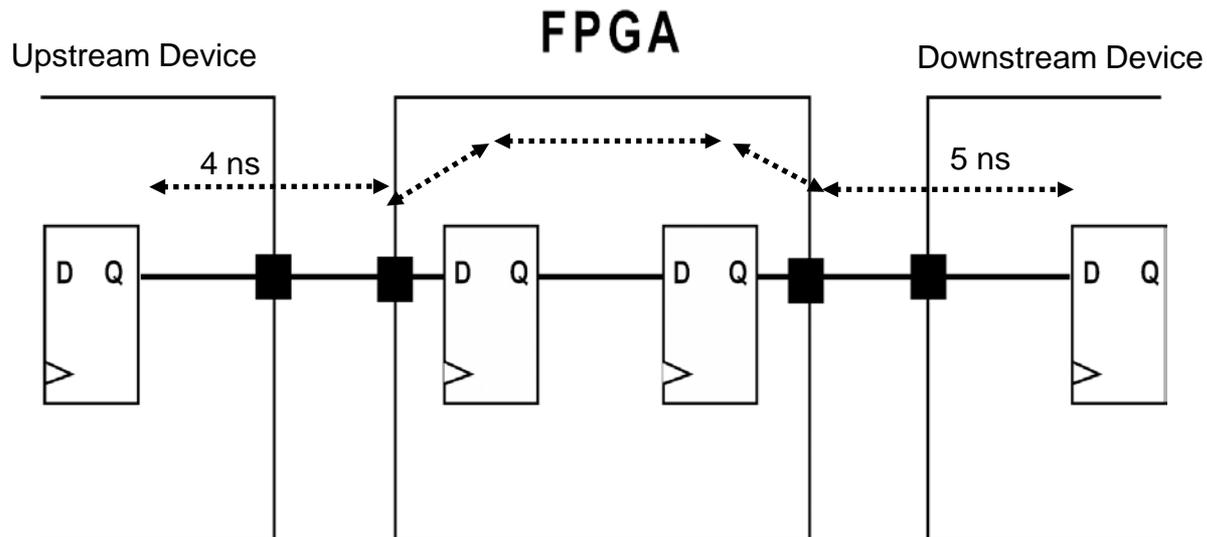
Input Jitter

Time: 0 Units: ps

Comment:

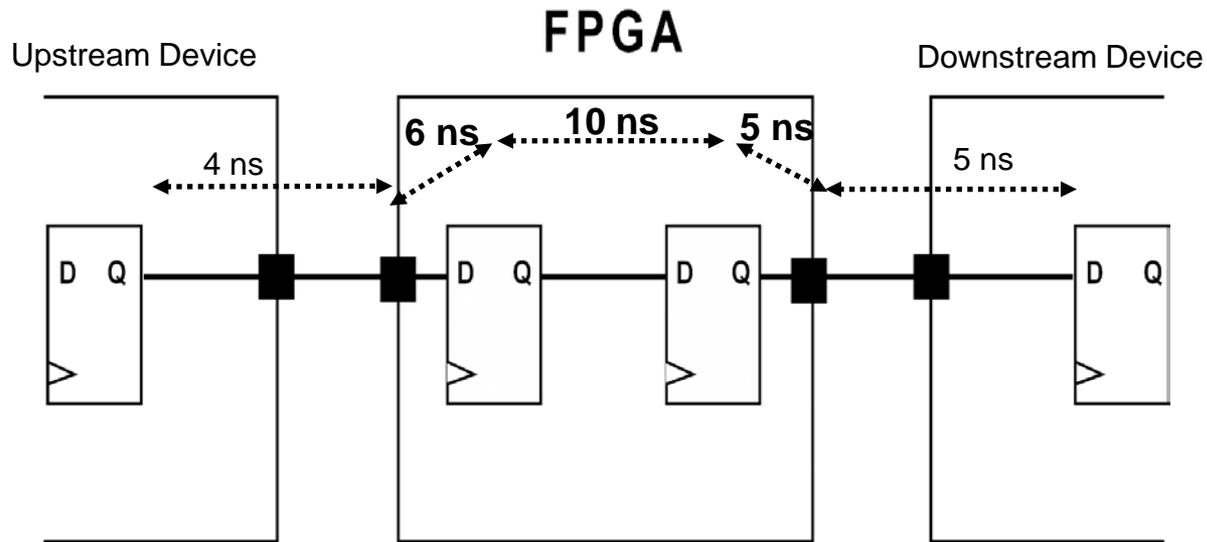
# Ejemplo

- Dado el sistema de abajo, qué valores debería colocar en el editor de restricciones para que el sistema opere a 100 MHz?
  - Asuma que no hay clock skew entre dispositivos



# Respuesta

- Dado el sistema de abajo, qué valores debería colocar en el editor de restricciones para que el sistema opere a 100 MHz?



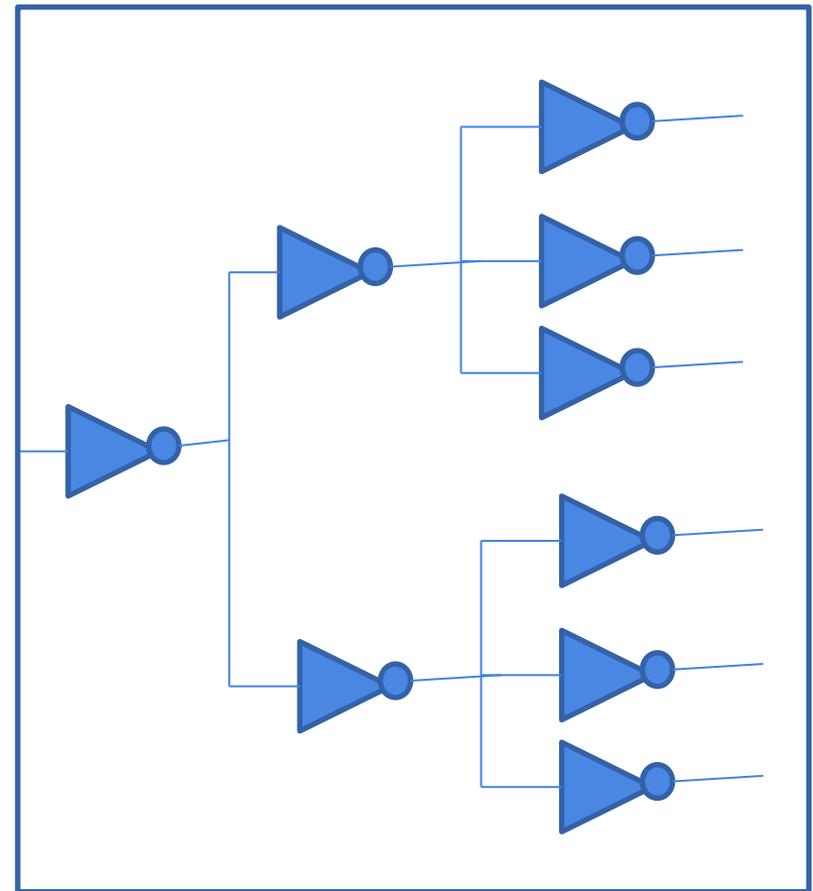
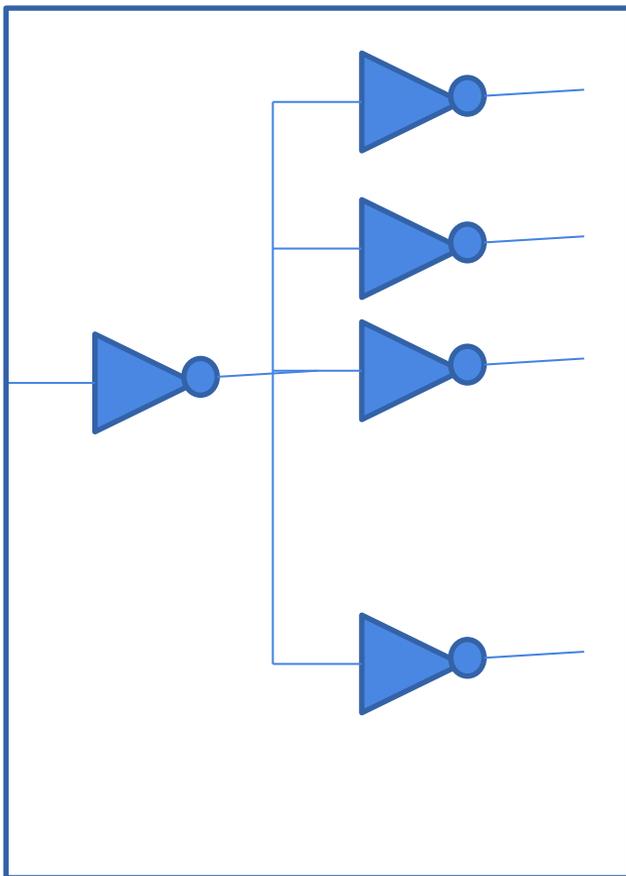
- Respuesta: PERIOD = 10 ns , OFFSET IN = 6 ns, and OFFSET OUT = 5 ns

## Guía sobre Fan-out

- Durante la síntesis se mantiene una guía razonable de Fan-out.
- Sin embargo, el usuario puede especificar unos lineamientos de fan-out máximos
- En Xilinx el Fan-out por defecto es 100, mientras que el mínimo posible es 8
- Fan-outs grandes pueden causar grandes retardos y problemas de enrutamiento.
- Durante la síntesis, el objetivo es mantener un fanout menor que el máximo permitido en las restricciones (pero no necesariamente se cumple)

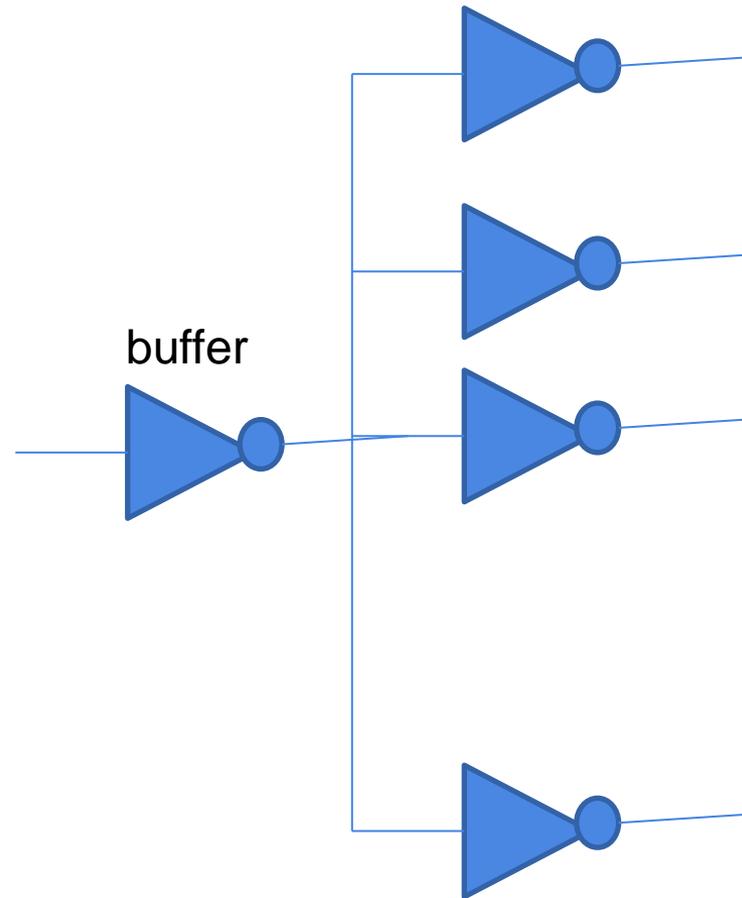
# Soluciones para el Fan-out: Replicación y división de salidas

El problema es cuando hay demasiados destinos y habría que hacer muchas subdivisiones, lo que aumenta el número de bits de registro y el número de LUTs. Otra opción es usar Pipelining



## Solución 2: Buffering

- Aumenta retardos y consume más recursos. No se usa a menos que se acerque bastante al límite de fan-out
- Otra opción es usar Buffers Globales en el diseño. Esto reduce el retardo para redes de fanout grande y libera recursos de enrutamiento para otras señales



# Recomendaciones

- Fan-outs entre 50 y 100 producen resultados aceptables.
- Usar Fan-outs menores que 50, especialmente algo así como 25 o 10, causa replicación de lógica excesiva, lo que hace que se infle el diseño y se ponga lento.
- Considere hacer buffers globales para señales de alto fan-out, aún cuando ellas sean lentas. Esto con el objetivo de reducir recursos de enrutamiento

## Aumento de velocidad: Evite anidar estructuras IF e IF/ELSE

- Los IF o IF/ELSE anidados forman codificadores de prioridad
- Por el contrario, las estructuras CASE no tienen prioridad
- Utilice Buffers globales para señales que van hacia muchas entradas CLK ó CE. Esto genera un skew de no más de 1ns
- Siempre que se pueda utilice señales cuyos cambios de nivel sean lentos, esto reduce las interferencias que se producen en las líneas internas del chip