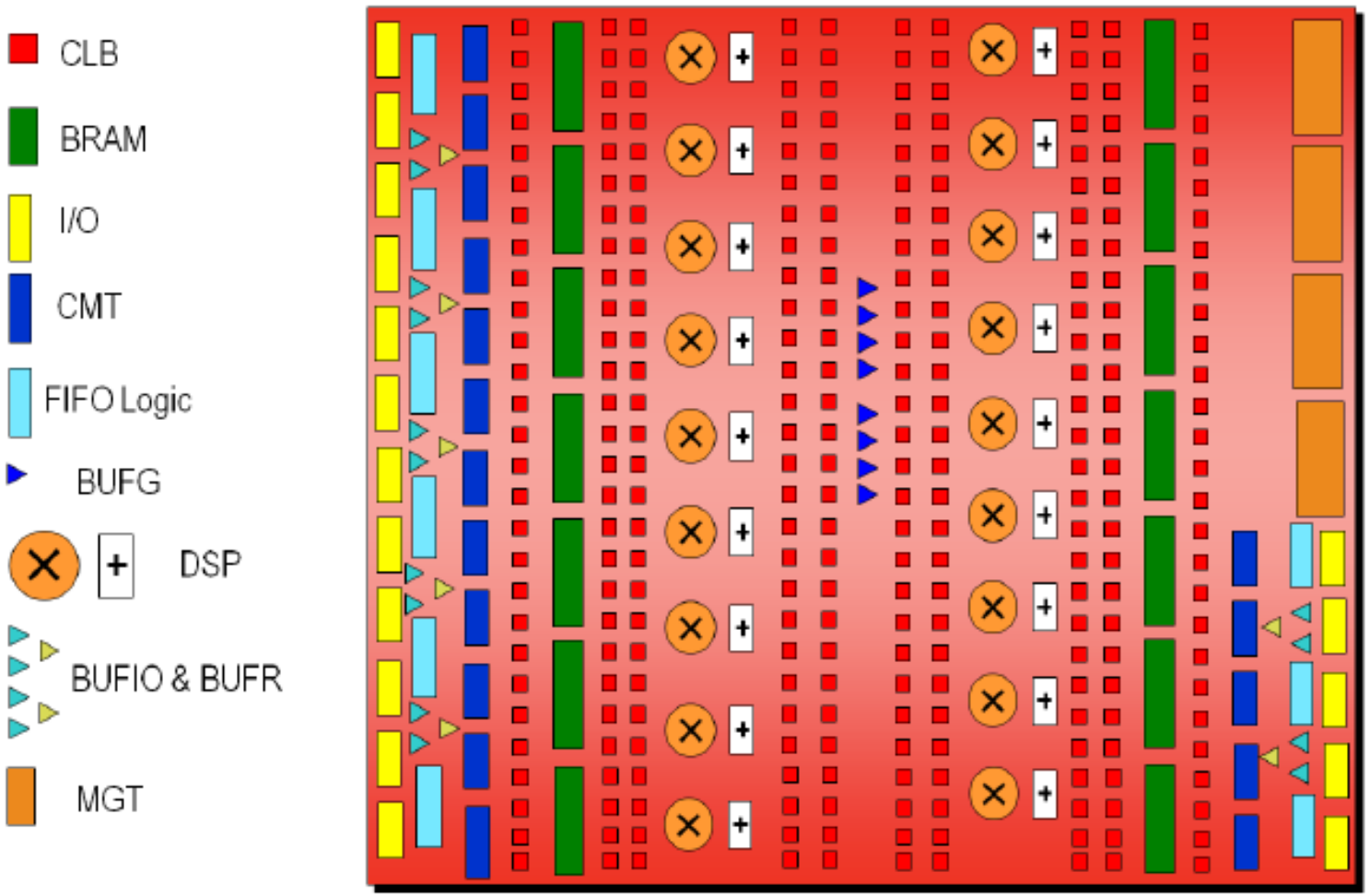
Práctica 3

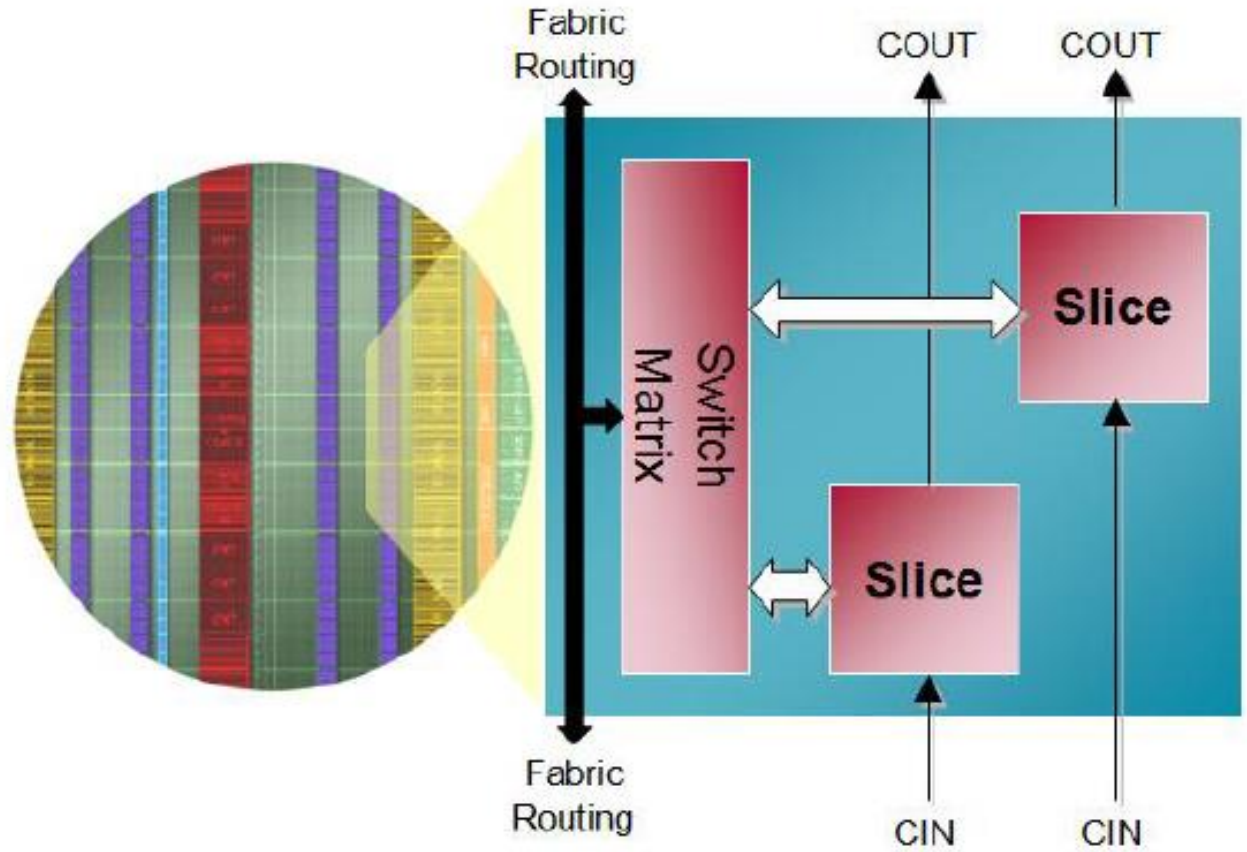
Diapositivas utilizadas en las explicaciones

Artix-7 FPGA Architecture Overview



Configurable Logic Block (CLB) in 7-Series FPGAs

- **Primary resource for design**
  - Combinatorial functions
  - Flip-flops
- **CLB contains two slices**
- **Connected to switch matrix for routing to other FPGA resources**
  - Carry chain runs vertically in a column from one slice to the one above



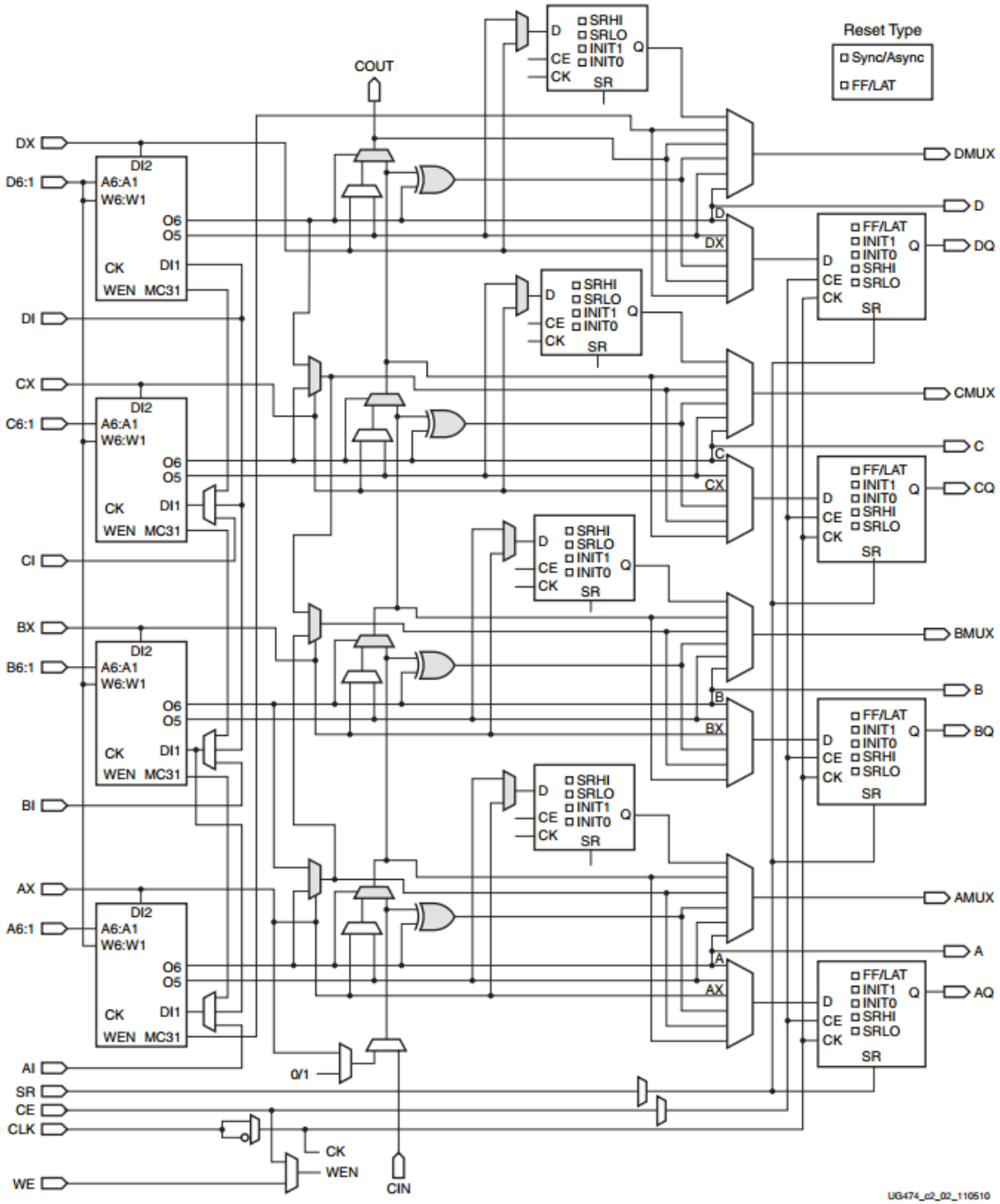
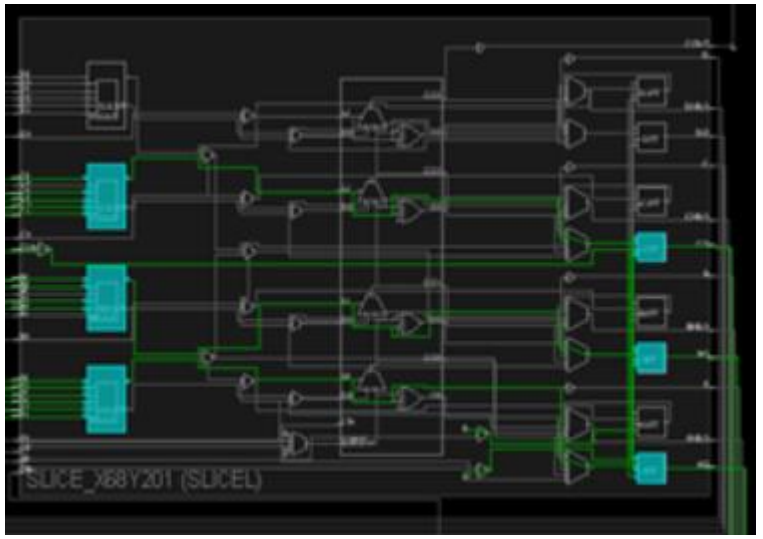
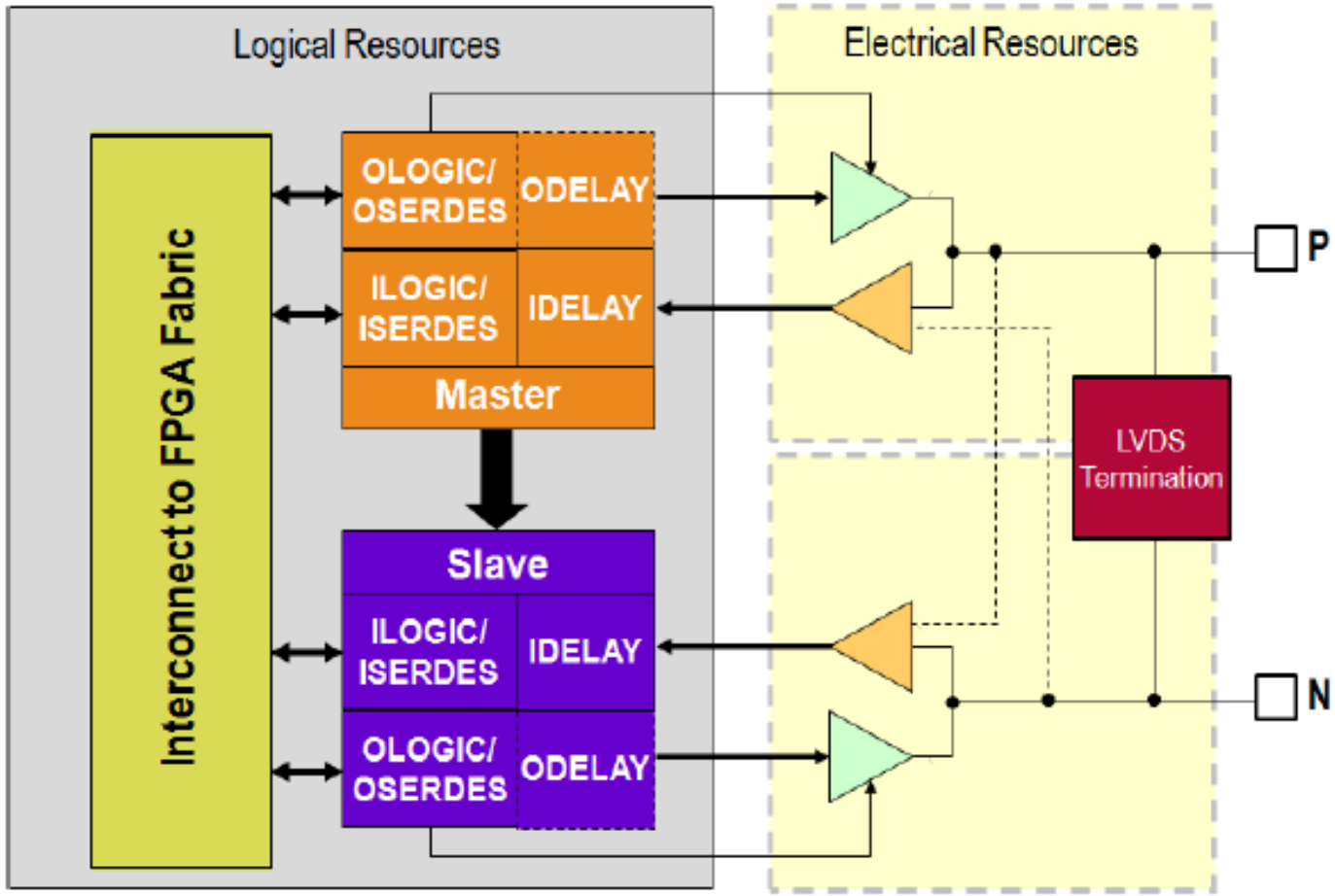


Figure 2-3: **Diagram of SLICEM**

SLICE

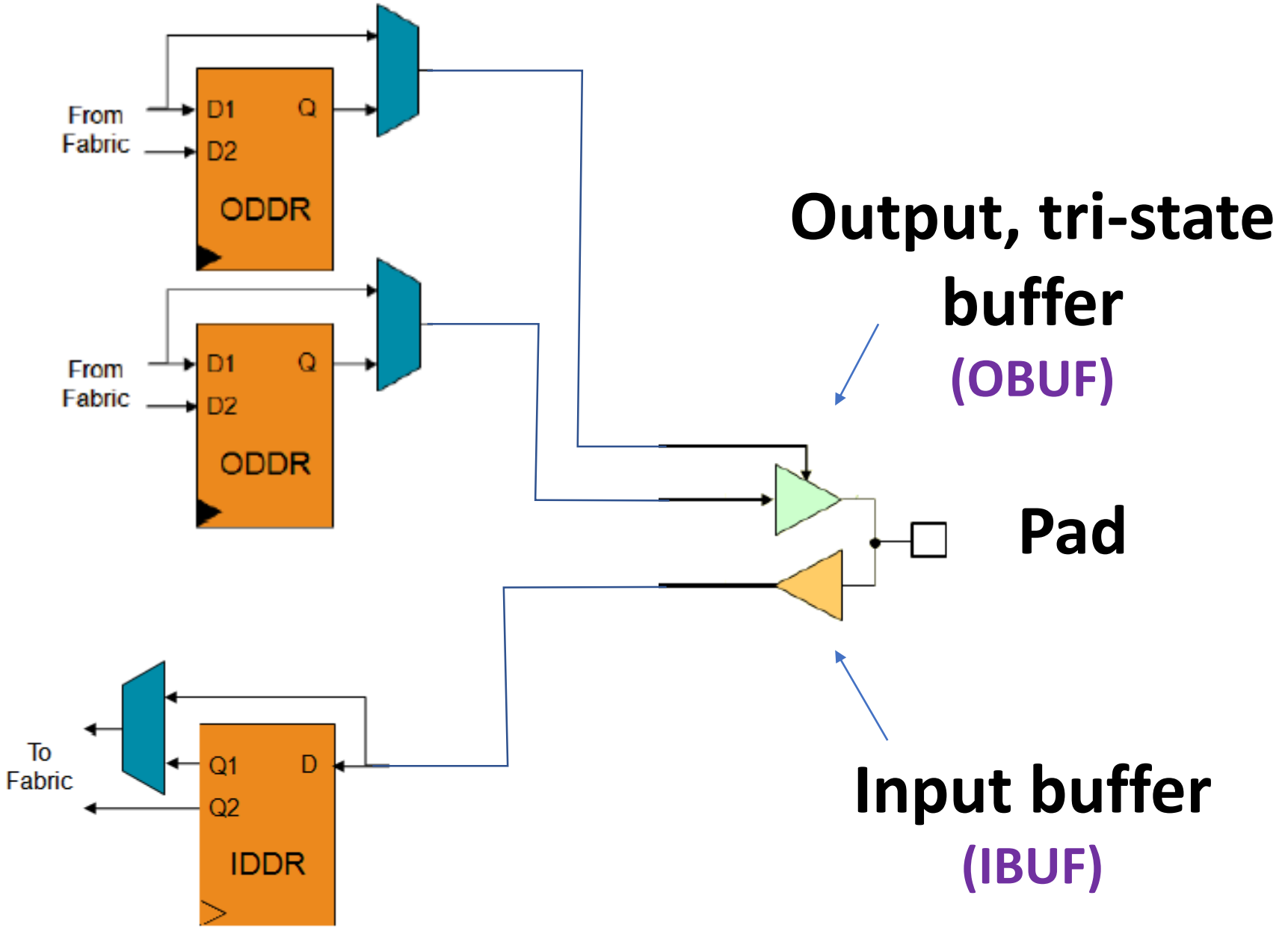


I/O Block Diagram



Entradas y salidas básicas

OLOGIC



ILOGIC

Output, tri-state buffer

(OBUF)

Pad

Input buffer

(IBUF)

STA: Static Timing Analysis – Qué es

Analizar de forma estática (sin una simulación en la que avance el tiempo) si el circuito cumple los requisitos de temporización.

- Entradas del proyecto: *constraints*
  - Periodo de reloj.
  - Tiempos de propagación de datos fuera de la FPGA.
  - Etc.
- Entradas de la tecnología:
  - Retardos en los diferentes elementos.
  - Requisitos de *setup, hold,* etc.
  - Estructura
  - Jitter previsible para el reloj
  - Etc.
- Salida:
  - Pasa / no pasa = ¿Estamos seguros de que cualquier pieza funcionará?

STA: Static Timing Analysis - Necesidad

Si nos funciona en el laboratorio, ¿funcionará en la aplicación?

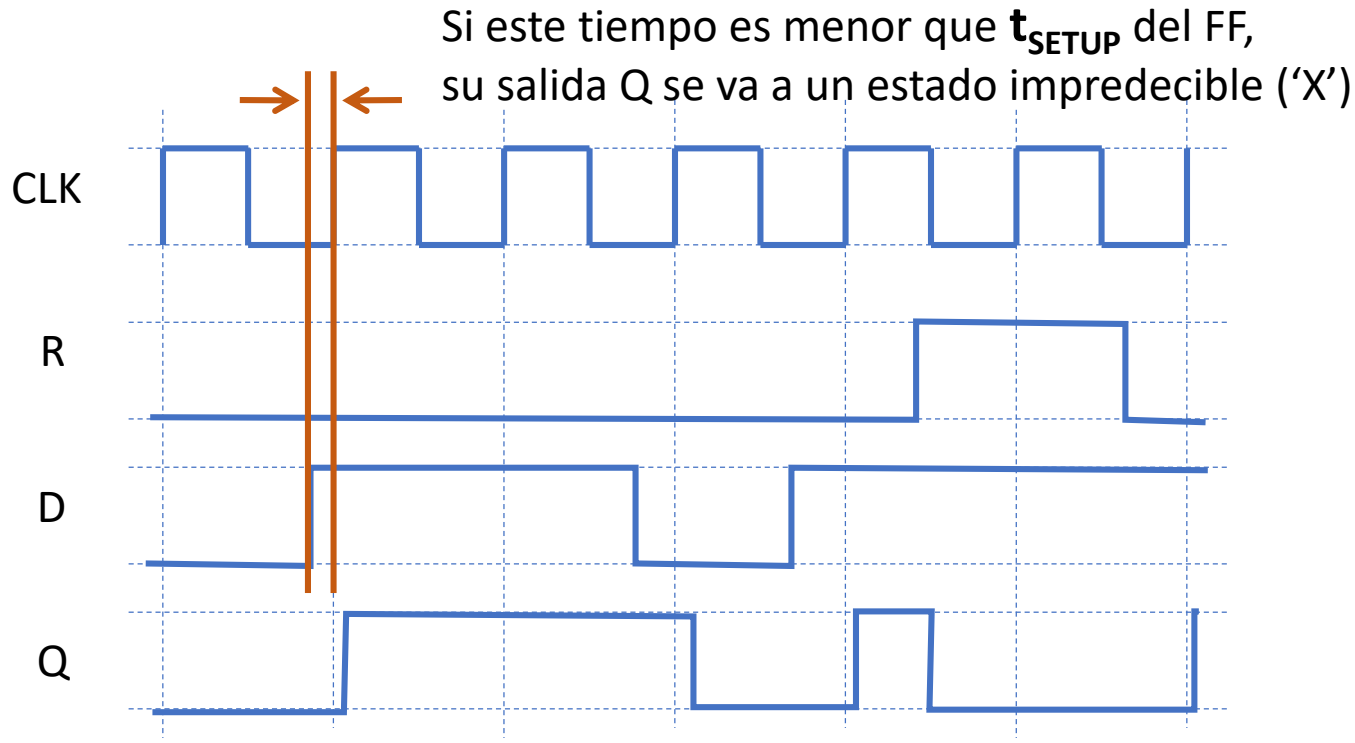
⇒ **No tiene por qué**

- **Pueden variar nuestras características de operación**
  - **P**roceso -> Por fabricación, salen piezas más rápidas que otras
  - **V**oltaje -> Velocidad a 0.98V de alimentación ≠ velocidad a 1.01V
  - **T**emperatura -> La velocidad también depende de ella

  => Caracterización **PVT** realizada por el fabricante.
- **Pueden variar las circunstancias externas**
  - P.ej. la velocidad de los chips que nos rodean.

Timing: *Setup* y *hold*

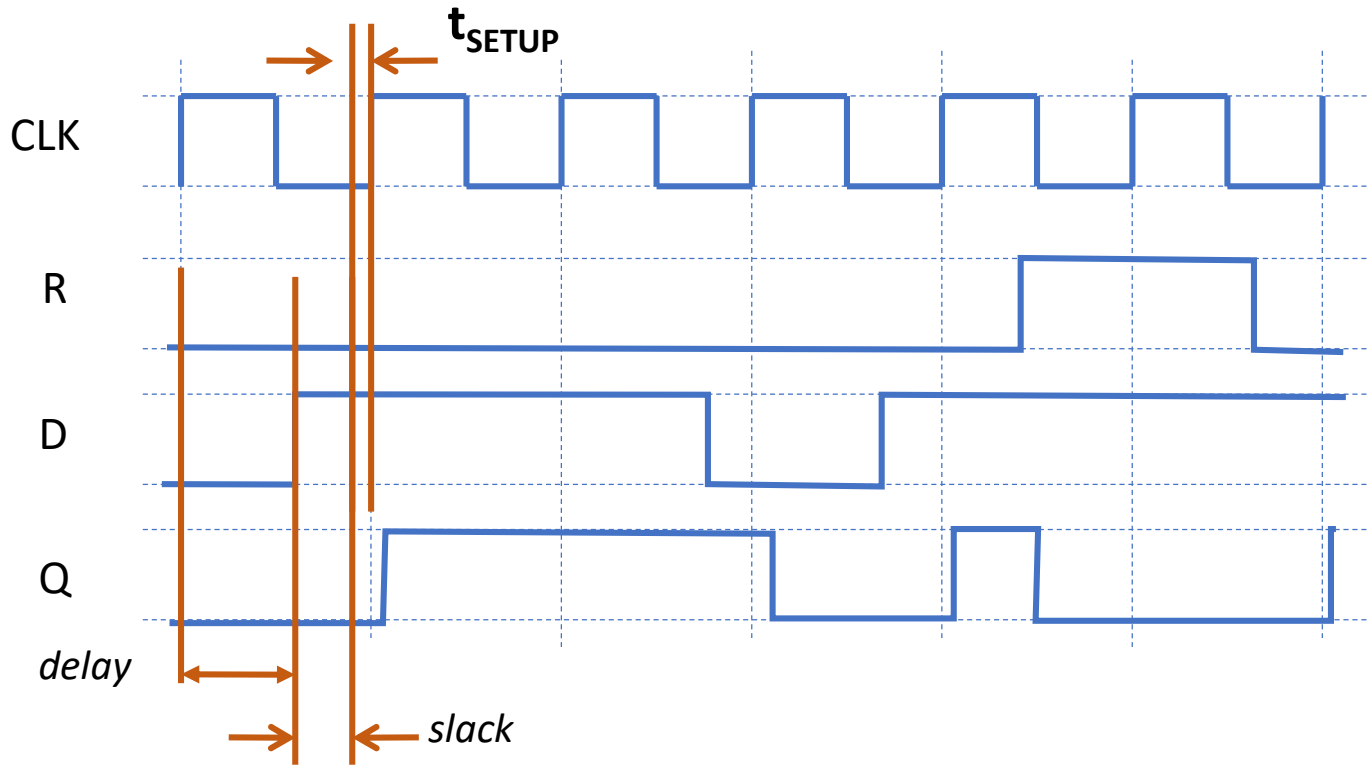
- ***Setup***: Los FF no son “perfectos”, necesitan que el dato de entrada esté estable un mínimo de tiempo antes:



- ***Hold***: Mismo concepto, pero el requerimiento es de estabilidad de *D* tras el flanco de reloj. ← También hay que cumplirlo, pero lograrlo es menos problemático, en estas prácticas no le prestaremos atención.

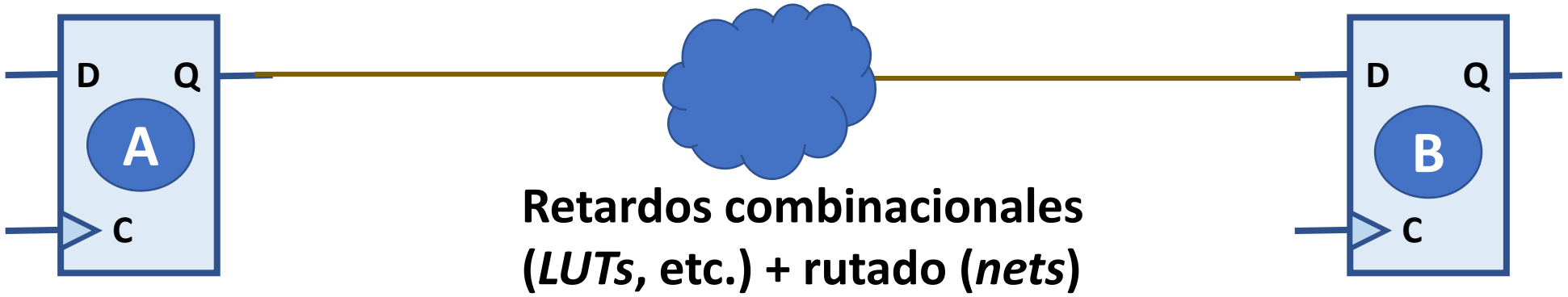
Slack y frecuencia máxima de operación

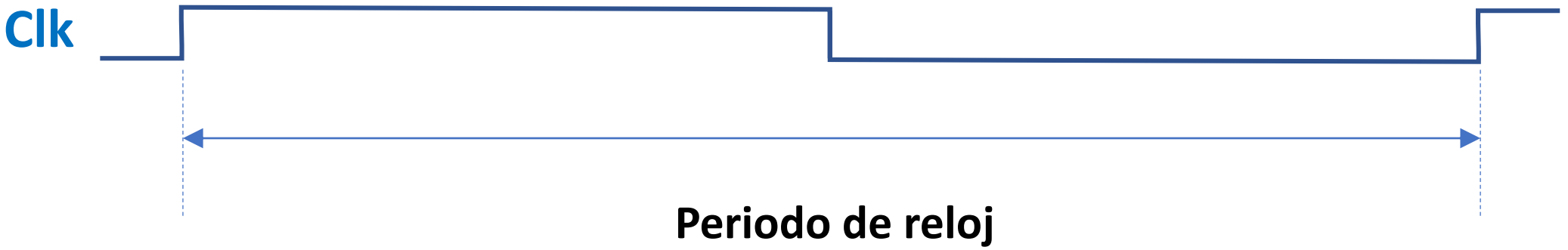
- El **slack** es el tiempo que nos sobra para poder cumplir con los requisitos (si es negativo, mal)



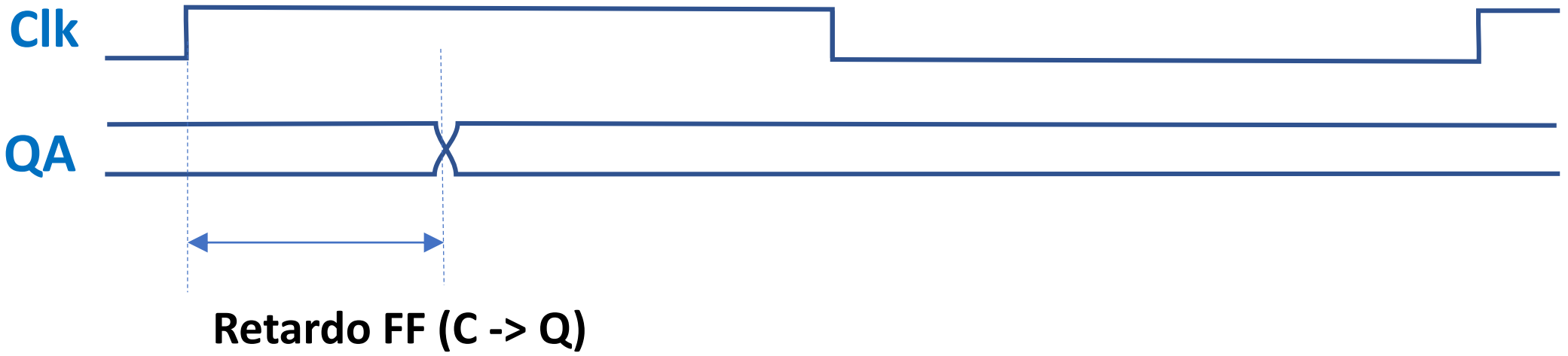
Si encogemos más y más el periodo de reloj, el retardo (*delay*) va a ser el mismo y el tiempo de *setup* también, pero el *slack* disminuye. Cuando es 0 hemos alcanzado la frecuencia máxima para ese "*path*".

Camino entre FFs

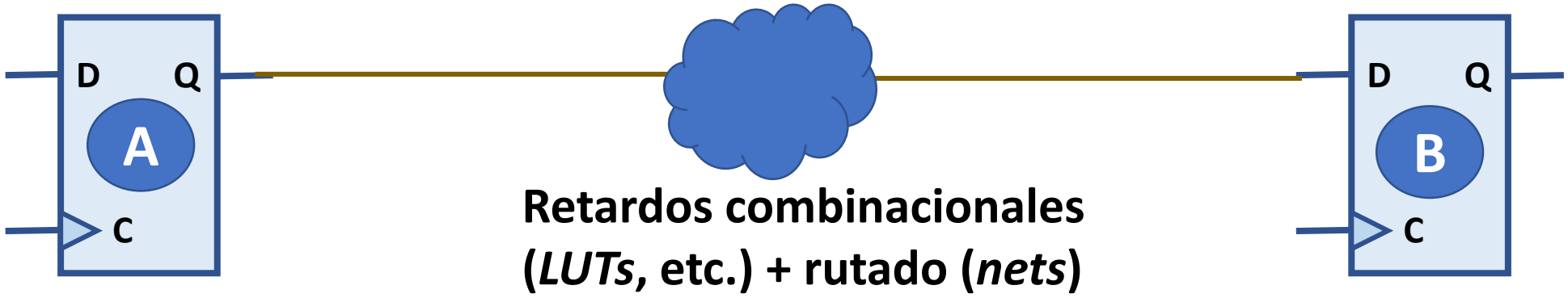


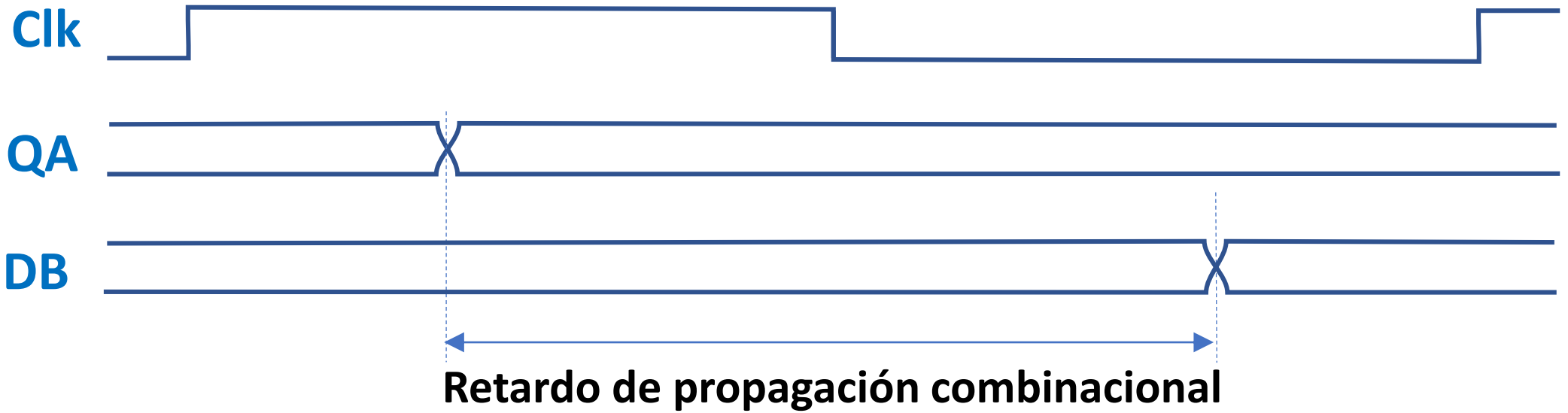


Camino entre FFs

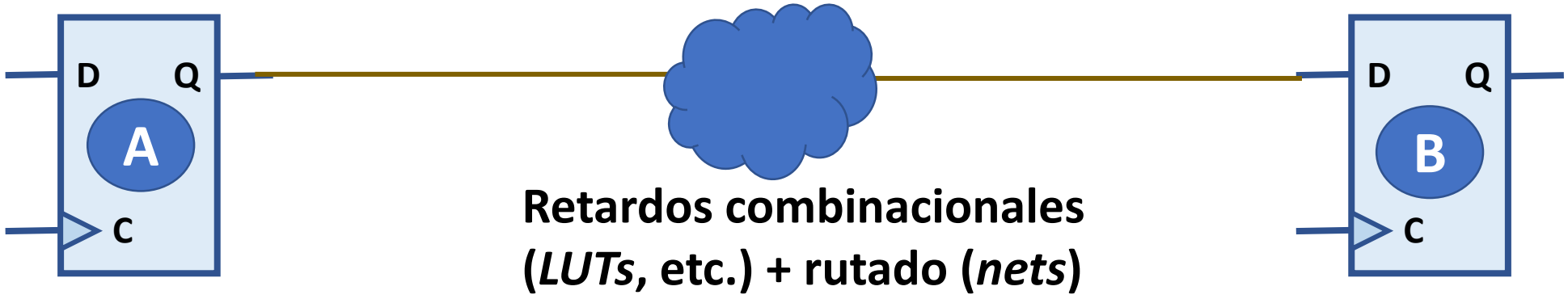


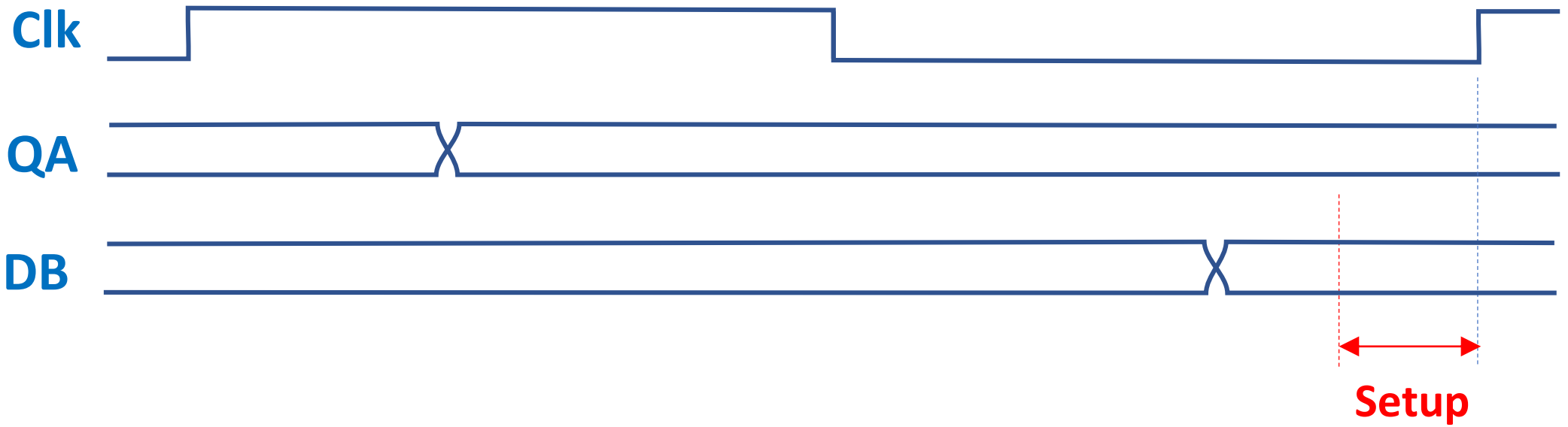
Camino entre FFs



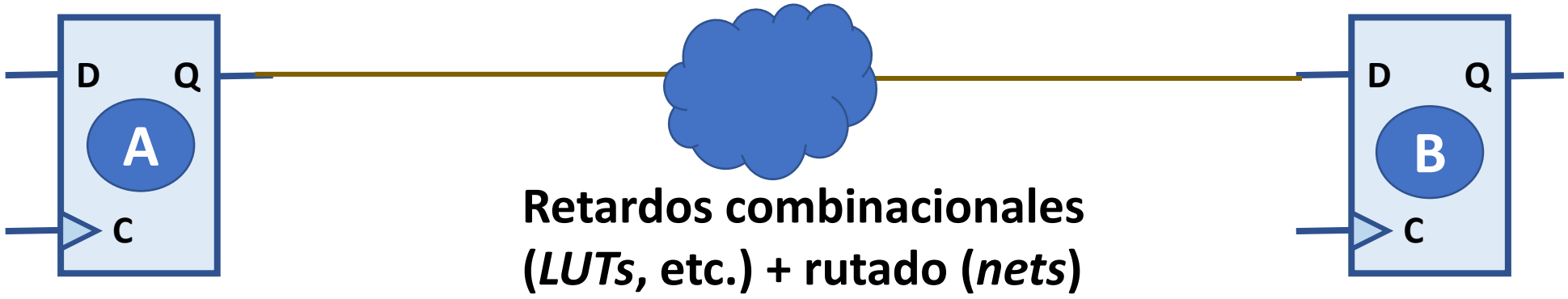


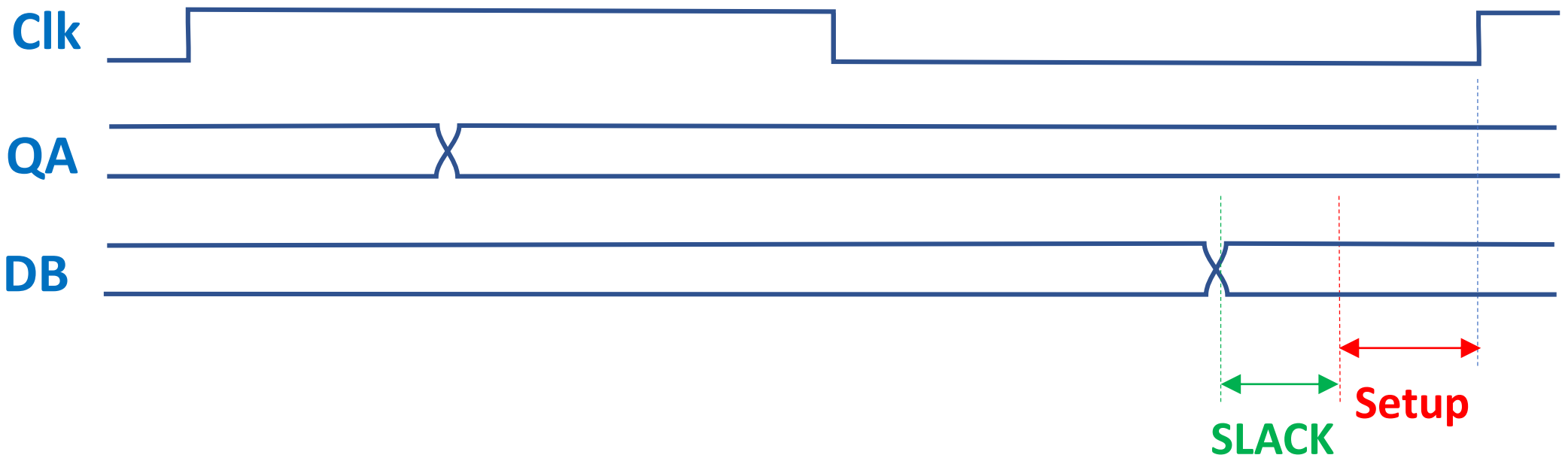
Camino entre FFs



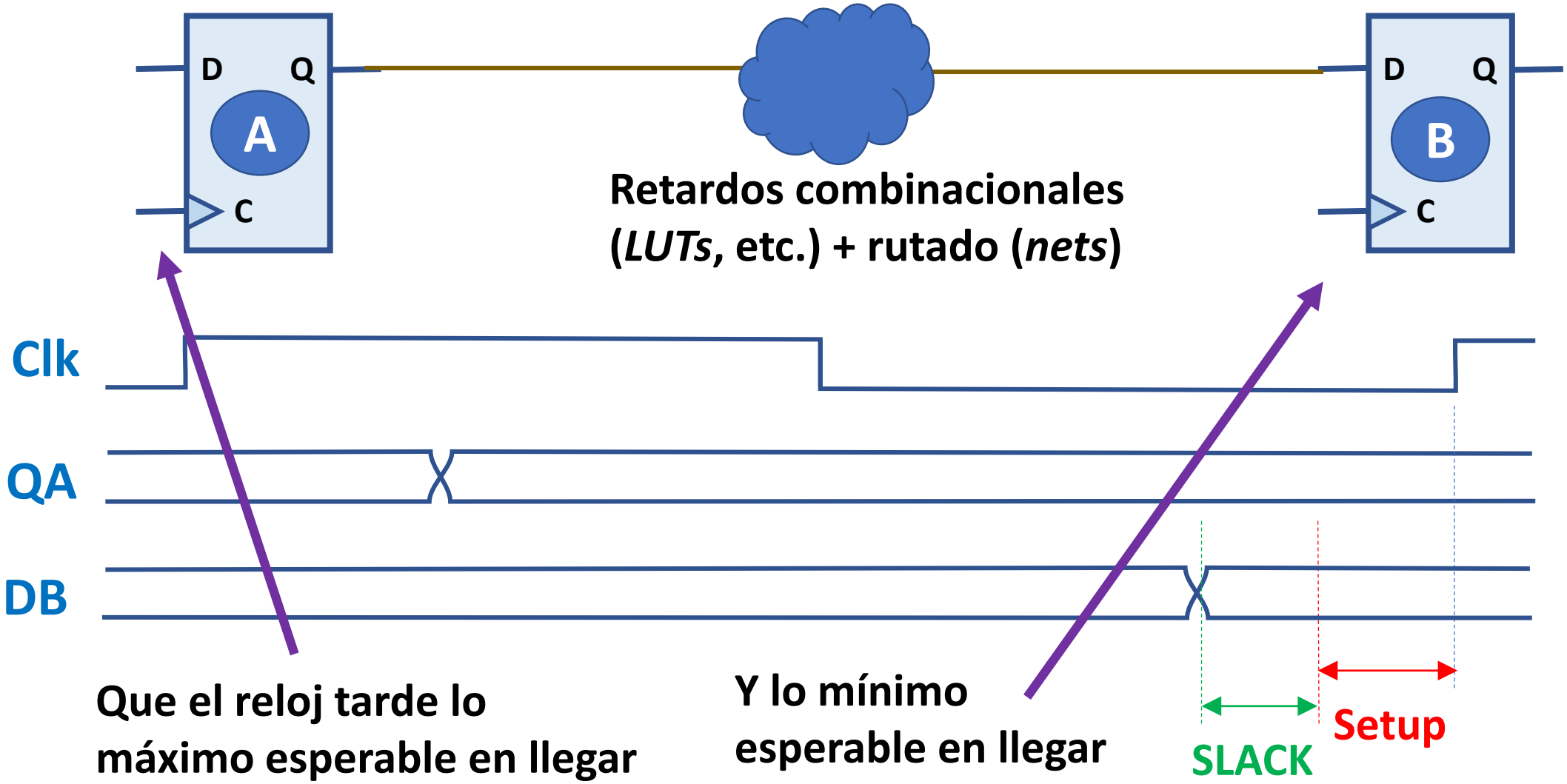


Camino entre FFs

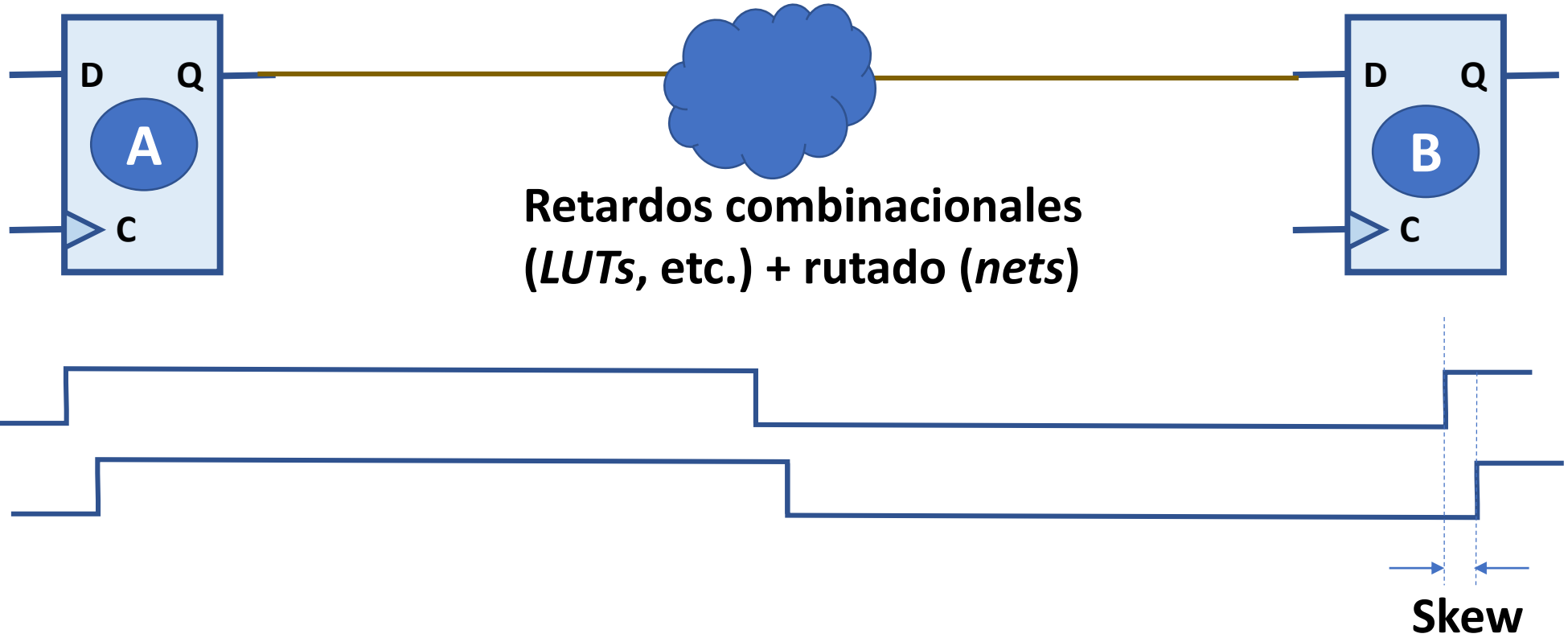




Camino entre FFs: análisis máximo-mínimo



Clock skew



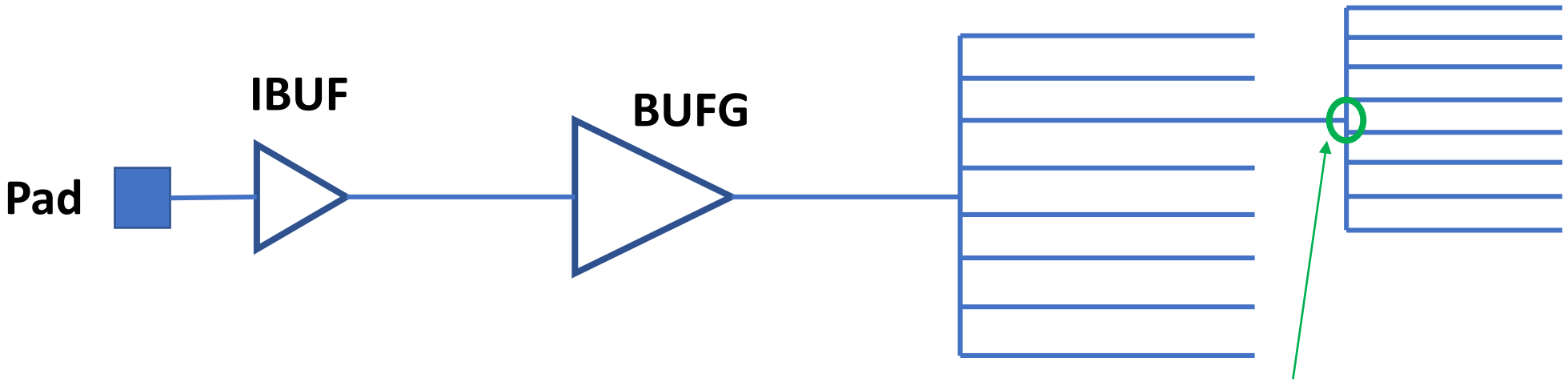
Por los distintos caminos que recorre el reloj, el flanco no llegará exactamente a la vez a ambos FFs.

Para el análisis de cumplimiento en *setup*:

- **Llega después al destino → a nuestro favor**
- **Llega antes al destino → en nuestra contra**

Clock pessimism

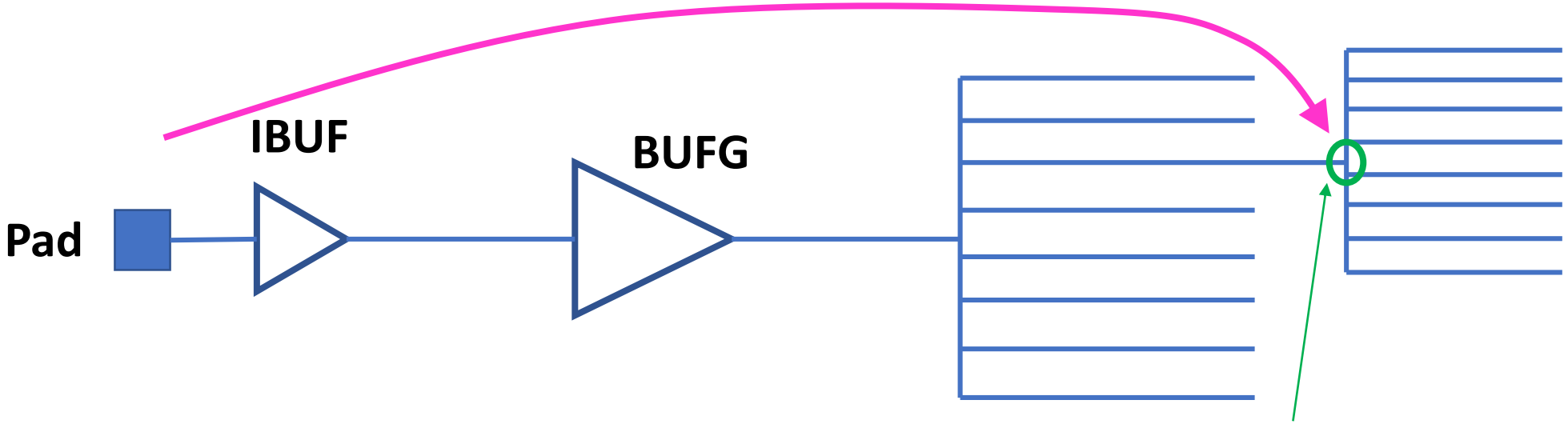
Red de distribución de reloj a los FFs (*clock tree*)



Hasta ese punto el retardo siempre va a ser igual para cualquier FF que dependa de este reloj en las ramas de la derecha (aunque, para todos, dependa de PVT)

Clock pessimism

Red de distribución de reloj a los FFs (*clock tree*)

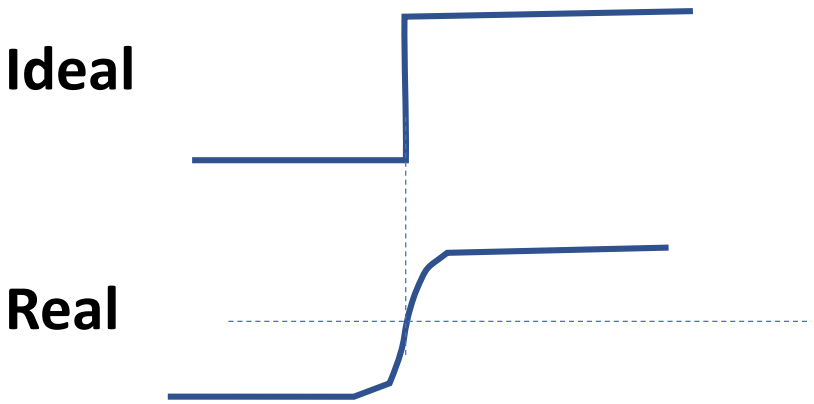


Hasta ese punto el retardo siempre va a ser igual para cualquier FF que dependa de este reloj en las ramas de la derecha (aunque, para todos, dependa de PVT)

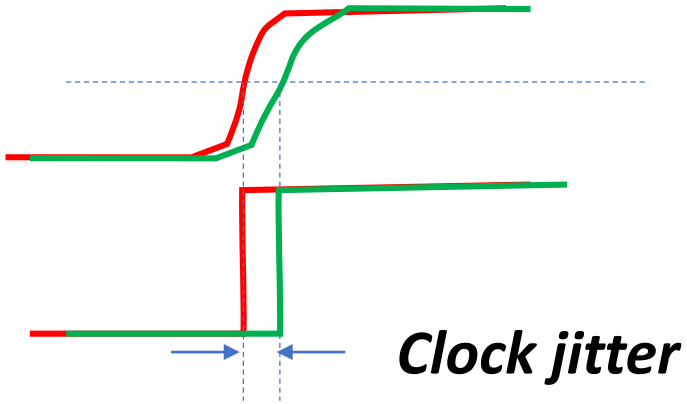
- Vivado, cuando verifica los *setup*, asigna primero un retardo mayor (máx.) hasta el FF origen y uno menor (mín.) hasta el FF destino
- Luego corrige la parte que ha de ser igual: *clock pessimism*

Clock uncertainty

- Realmente los flancos de reloj no son perfectos



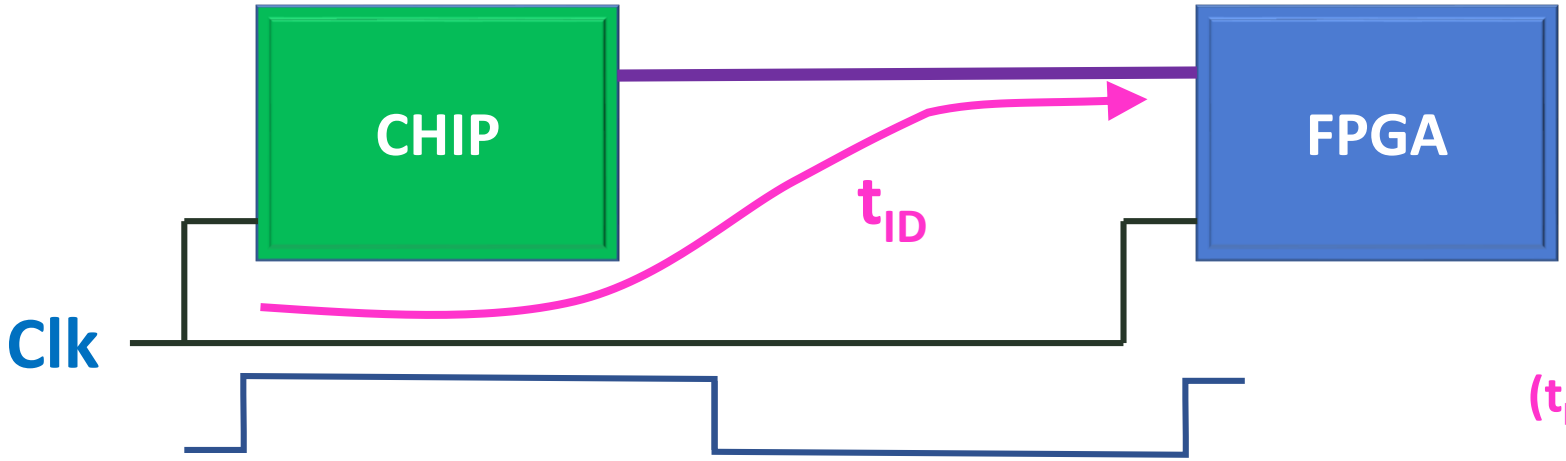
- La detección del cambio de 0 a 1 tendrá lugar a un cierto voltaje
- La señal de reloj se ve influida por el ruido de la alimentación y de pistas cercanas



***Jitter* = Incertidumbre**

Constraint de entrada: *set_input_delay*

- Si en una entrada recibimos una señal desde otro chip con el mismo reloj, Vivado no puede saber cuánto ha tardado ese chip en proveer el nuevo dato desde el flanco de reloj



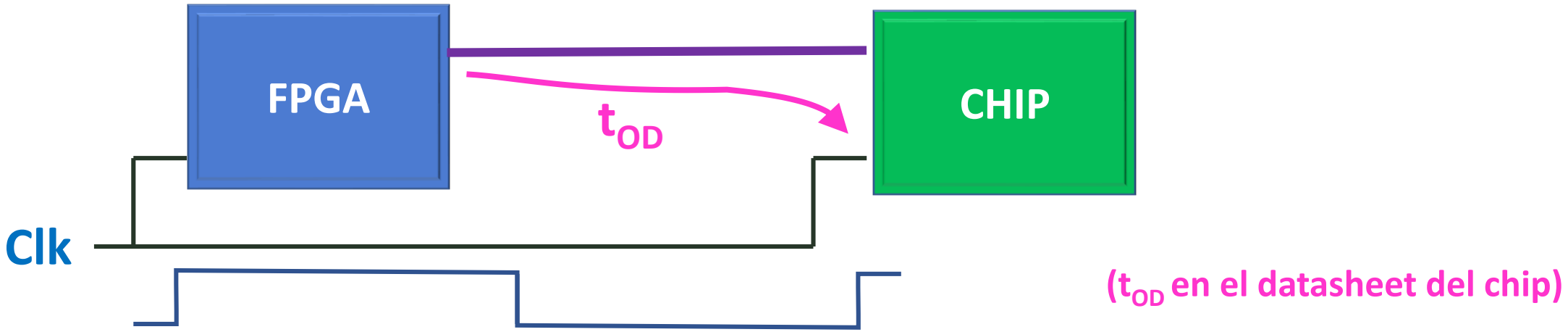
(t_{ID} en el datasheet del chip)

- Consideramos el otro chip como un FF y le decimos a Vivado cuánto ha tardado en darnos el dato desde el flanco del reloj en la placa:

set_input_delay -clock **reloj** **t_{ID}** **puerto_de_entrada**

Constraint de salida: *set_output_delay*

- Si damos una salida a otro chip con el mismo reloj, Vivado no puede saber cuánto tiempo antes del flanco de reloj necesita ese chip tener el dato estable



- Consideramos el otro chip como un FF y le decimos a Vivado que nuestra salida tiene un recorrido externo igual al *setup* especificado en la correspondiente entrada del chip:

set_output_delay -clock **reloj** **t_{OD}** **puerto_de_salida**

Regla de diseño: registrar las salidas

Distintos diseñadores pueden implementar distintos bloques (o “IPs“) de un diseño más grande.

¿Qué pasa si varios bloques se conectan en cadena?

- Si varios diseñadores dejan un *path* combinacional entre entradas y salidas, se tendrá un *path* combinacional más largo -> menor frecuencia máxima
- Si cada diseñador se preocupa de registrar sus salidas, los caminos combinacionales quedarán limitados a un bloque.

Seguir esta regla permite que los diseñadores analicen independientemente la velocidad de sus bloques tras síntesis (también puede ser tras implementación física)