
Diseño VLSI

Generación y distribución de la señal de clock

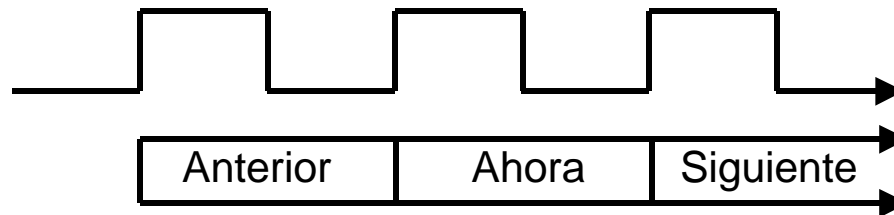
Enric Pastor

Dept. Arquitectura de Computadors

UPC

El porque de los clocks

- Necesitamos clocks ya que queremos que las salidas del sistema dependan no solo de las entradas, sino tambien de las salidas anteriores.
- Estas salidas anteriores son los denominados “*bits de estado*”.
- Los bits de estado pueden generar multitud de problemas si no se gestionan correctamente.
- Es preciso definir una “*política*” para definir que quiere decir “estado anterior”.



Contenido

- El funcionamiento de los clocks
- Esquemas de sincronización
- El clock en un sistema *edge triggered*
 - Retardo en una conexión del clock
 - Restricciones debidas al *skew*
- Distribución del clock
 - Ejemplo: DEC Alpha 21164 / 21064
- El clock en un sistema *single-phase*
- El clock en un sistema *two-phase*
 - El *skew* en un sistema *two-phase*
 - Importancia de *non-overlapping clocks*
- Generación del clock

El funcionamiento de los clocks

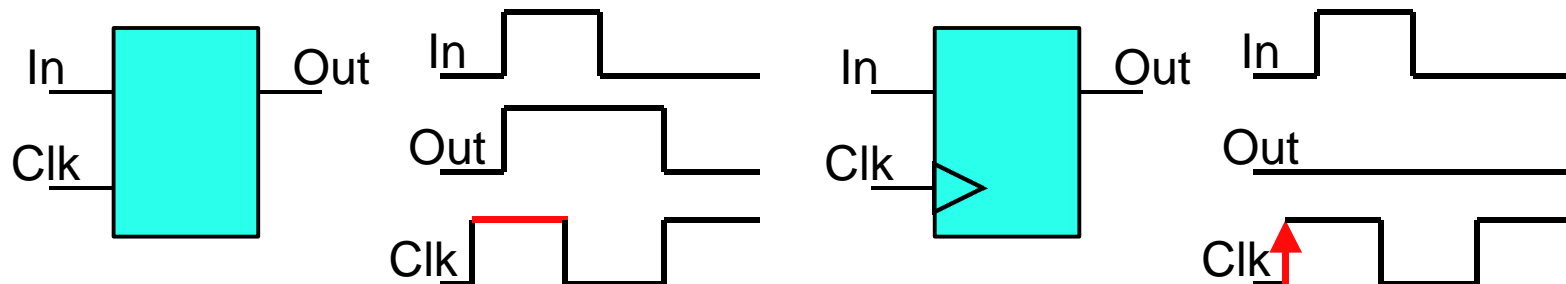
- Trabajan con **latches** y **flip-flops** para almacenar el estado:

Latch, sincronización por nivel:

- Clock **1**, la entrada se transmite a la salida.
- Clock **0**, mantiene el valor en la Entrada cuando el clock bajo.

Flip-Flop, sincronización por flanco:

- En el flanco positivo, transfiere el valor en la entrada a la salida.
- Mantiene el valor en cualquier otro caso.



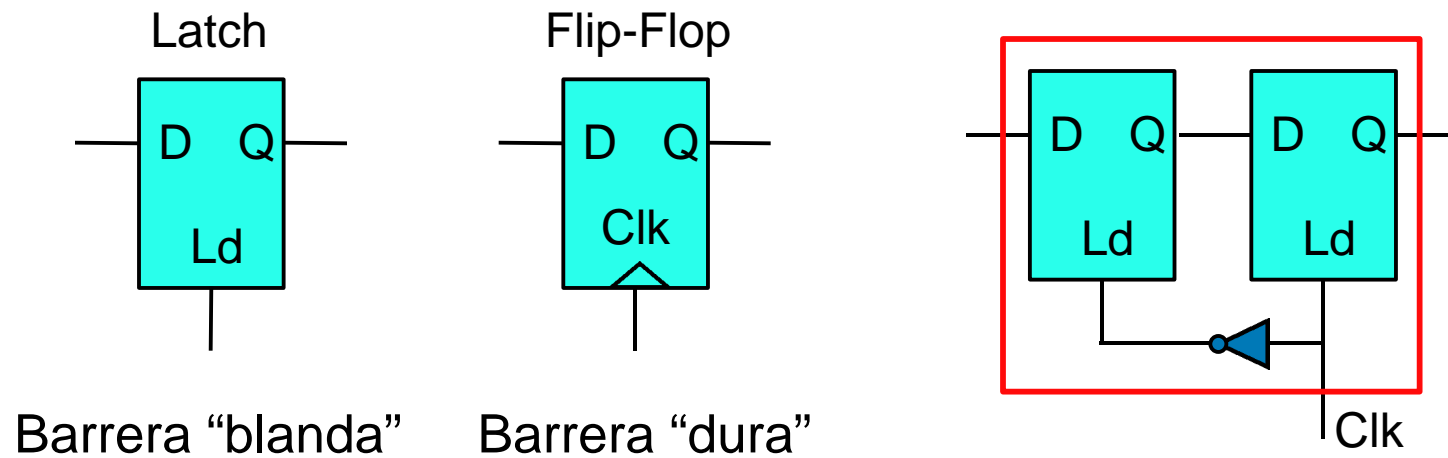
El funcionamiento de los clocks

- El objetivo es mantener las señales correladas en el tiempo:
 - Es necesario garantizar que las señales no se mezclan.
 - Es preciso saber que existe una cota.
- Si el retardo en cada path de la lógica es **exactamente** igual:
 - No se necesitan clocks.
 - La señales están correladas de manera natural.
 - El “estado” se almacena en las conexiones: “*wave pipelining*”.



El funcionamiento de los clocks

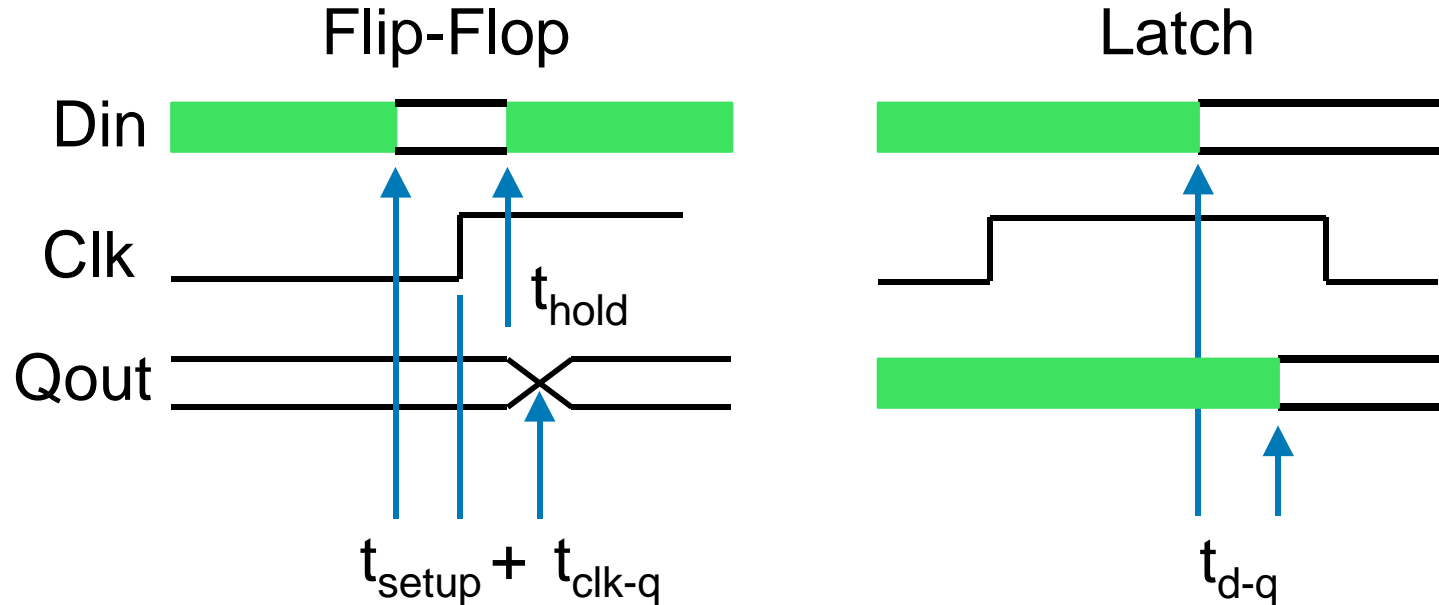
- Los clocks se utilizan para “*ralentizar*” las señales demasiado rápidas. Los flip-flops / latches actúan como barreras:



- En un latch, la entrada solo se propaga cuando el clock es **1**.
- En un flip-flop, la entrada se propaga exclusivamente en el flanco positivo del clock.

El funcionamiento de los clocks

- El problema adicional es que los latches y los flip-flops también ralentizan las señales lentas:



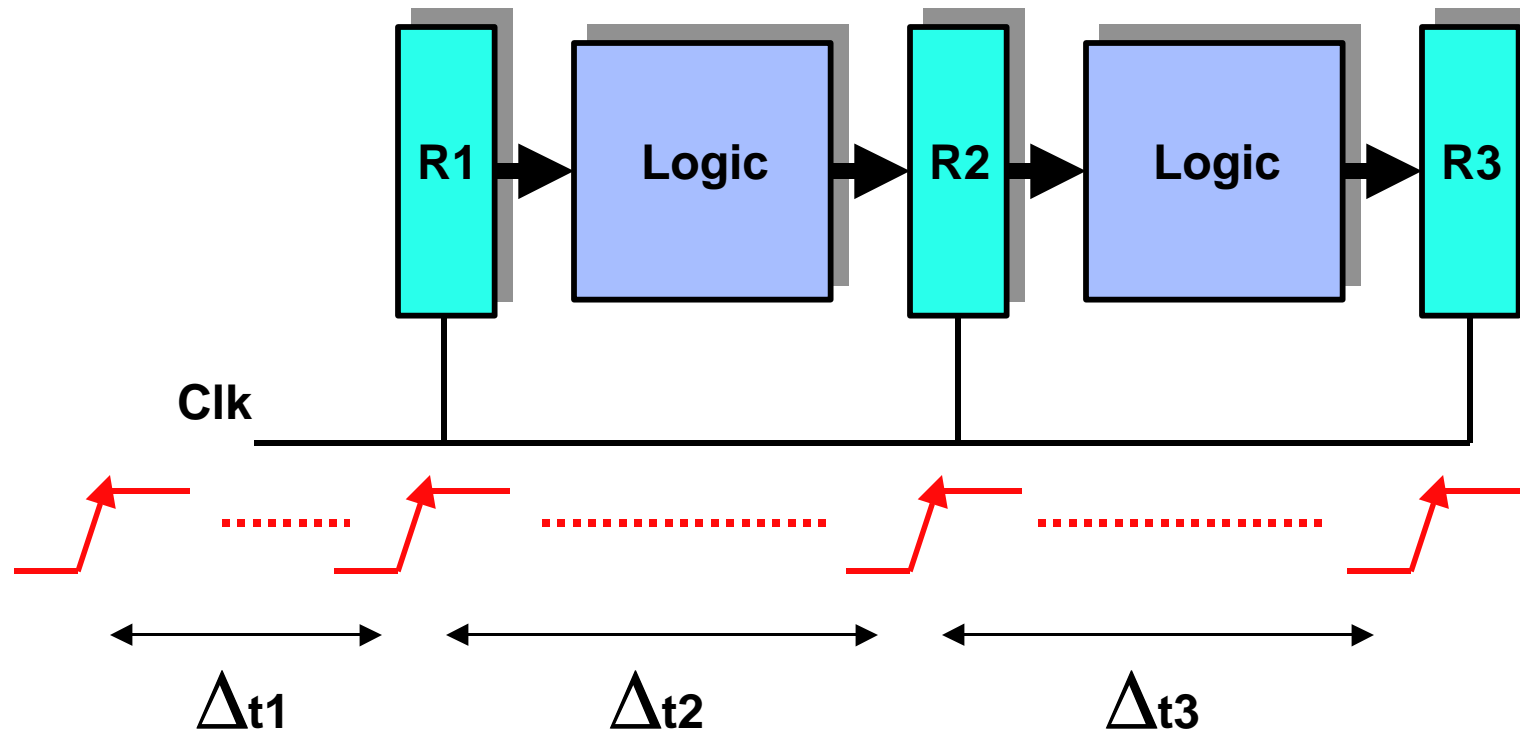
- Flip-Flops retardan el tiempo de **setup** más el de **clk-q**.
- Latches retardan el tiempo de **d-q**.

Esquemas de sincronización

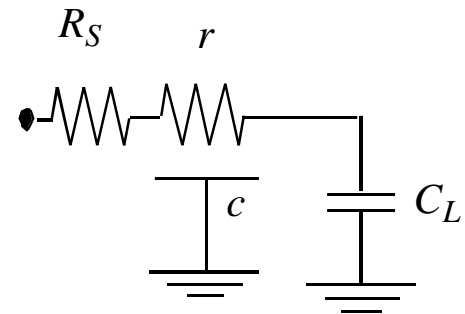
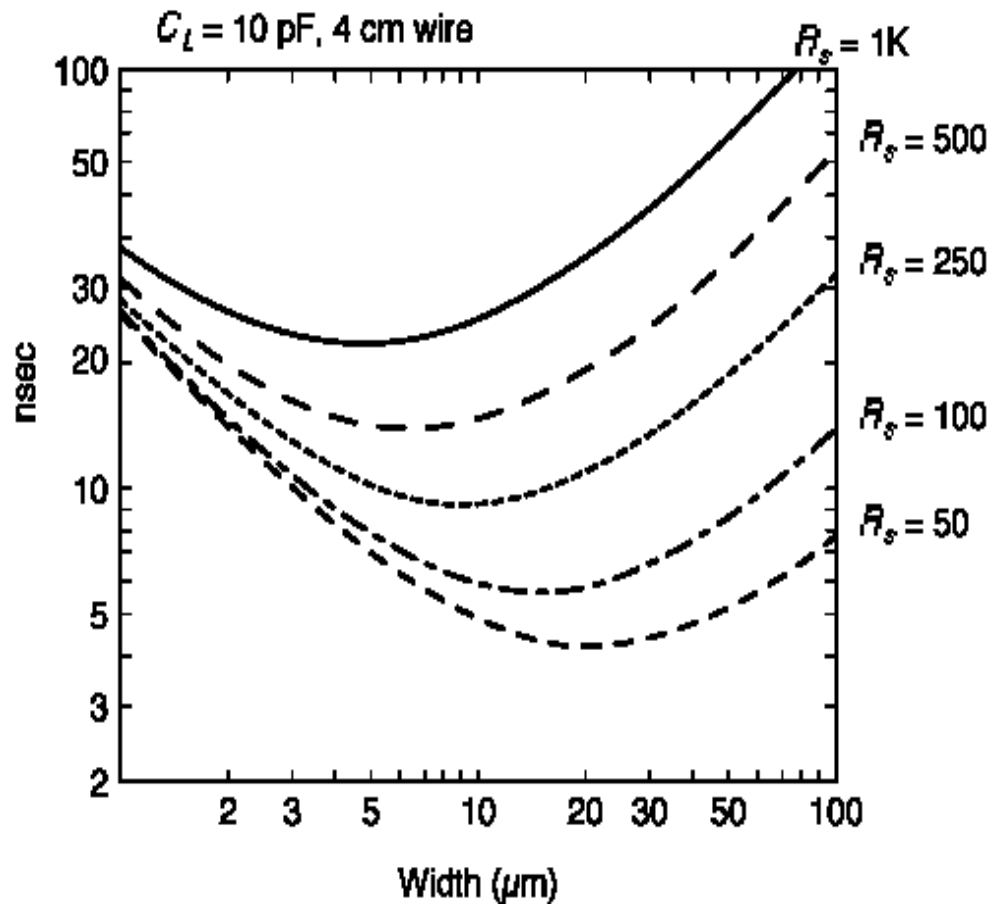
- La mayoría de los circuitos digitales son “*síncronos*”, por tanto, requieren una o más señales de clock:
 - Las distintas fases del clock deben generarse.
 - Cada fase debe distribuirse reduciendo el skew.
- El skew complica el diseño de los circuitos:
 - Reduce el tiempo de ciclo disponible.
 - Puede forzar a almacenar datos incorrectos.
- Existen distintas estrategias de clocking:
 - Edge triggered clocking
 - Single phase clocking
 - Two phase clocking
 - Pulse mode clocking

El clock en un sistema edge triggered

- El clock controla la carga de los latches/registros:
No es posible garantizar su distribución instantánea.



Retardo en una conexión del clock

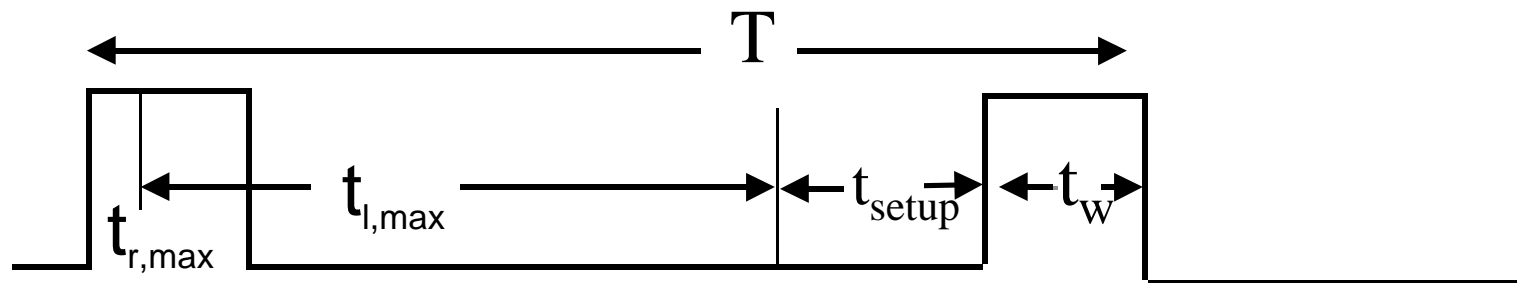


$r = 0.07 \text{ } \Omega/\text{q}$, $c = 0.04 \text{ fF}/\mu\text{m}^2$
 (Tungsten wire)

El clock en un sistema edge triggered

- Parámetros del clock

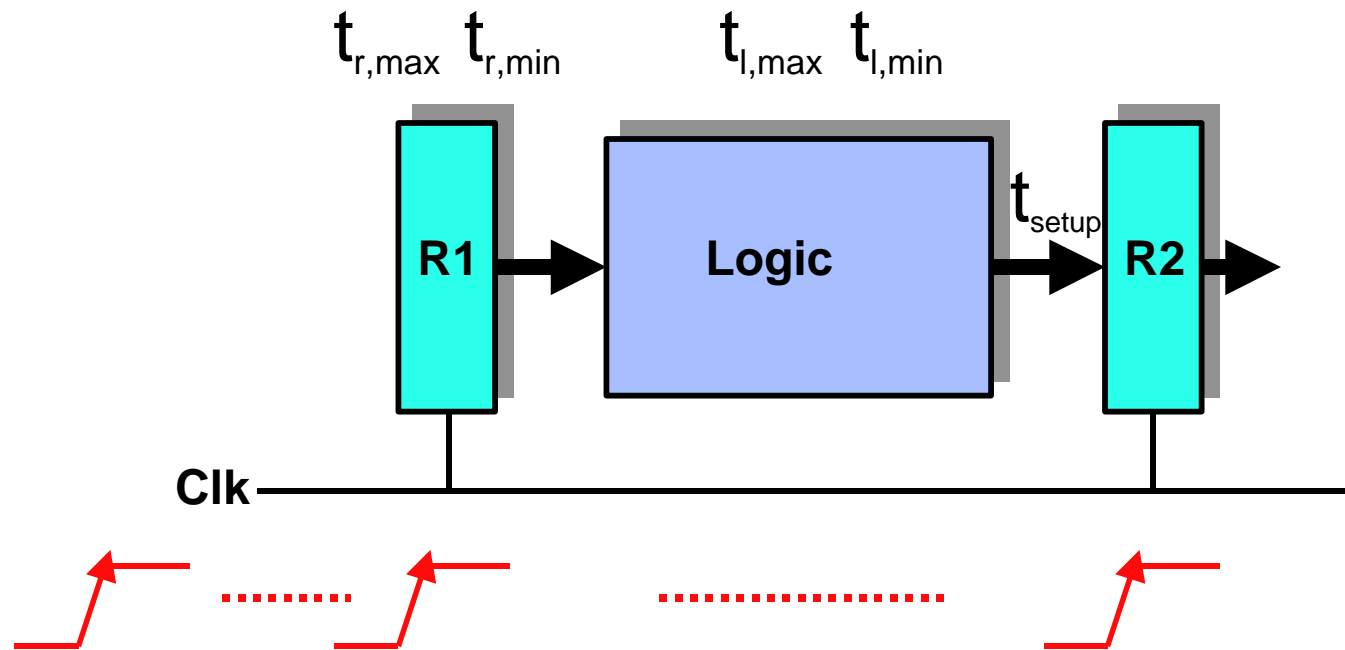
- $t_{l,max}/t_{l,min}$ - retardo en la lógica combinacional
- $t_{r,max}/t_{r,min}$ - retardo a la salida del registro
- T - periodo del reloj
- t_w - amplitud del pulso de reloj
- t_{setup} - tiempo de setup
- C - *overhead* total en el clock



$$T = t_{l,max} + C$$

Restricciones debidas al *skew*

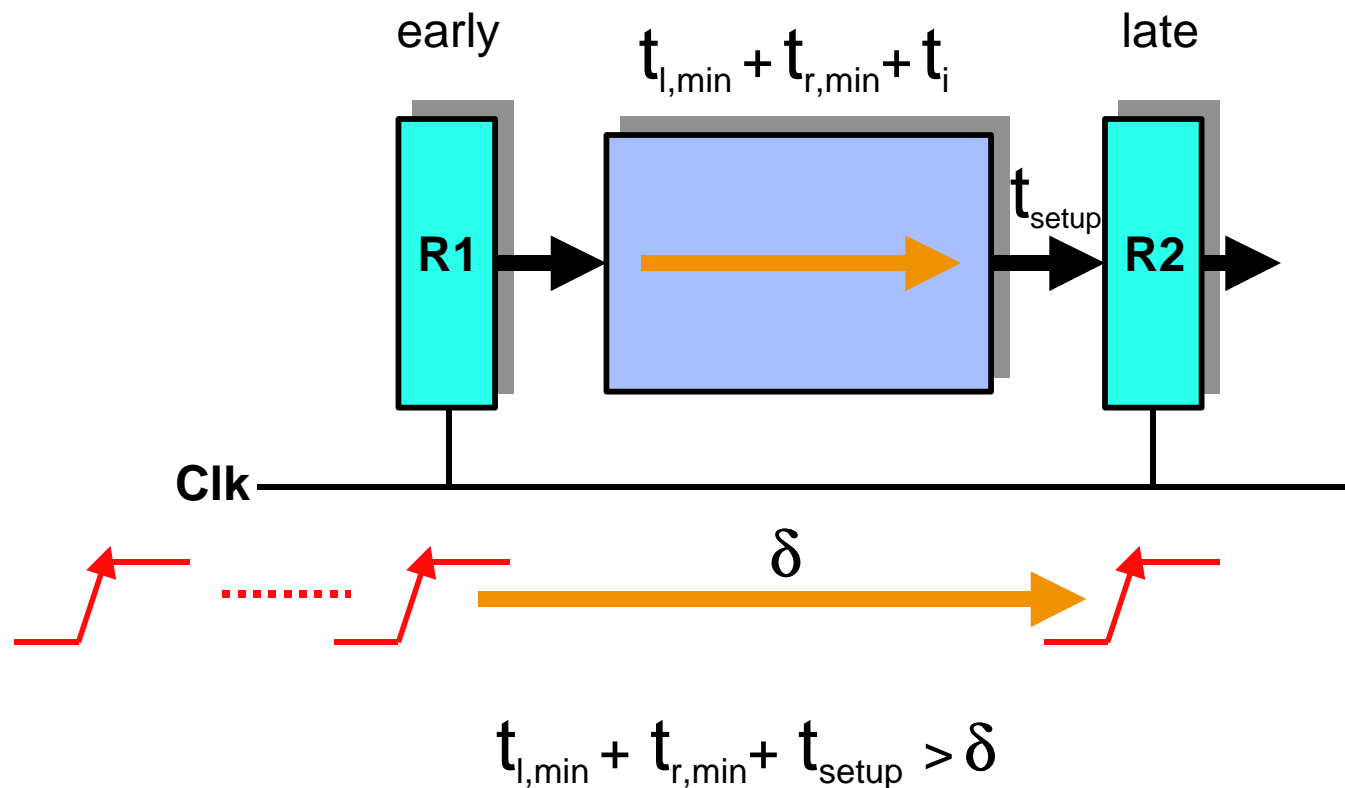
- Los parámetros del clock limitan los tiempos de propagación.



La posición del flanco depende de la posición del registro.

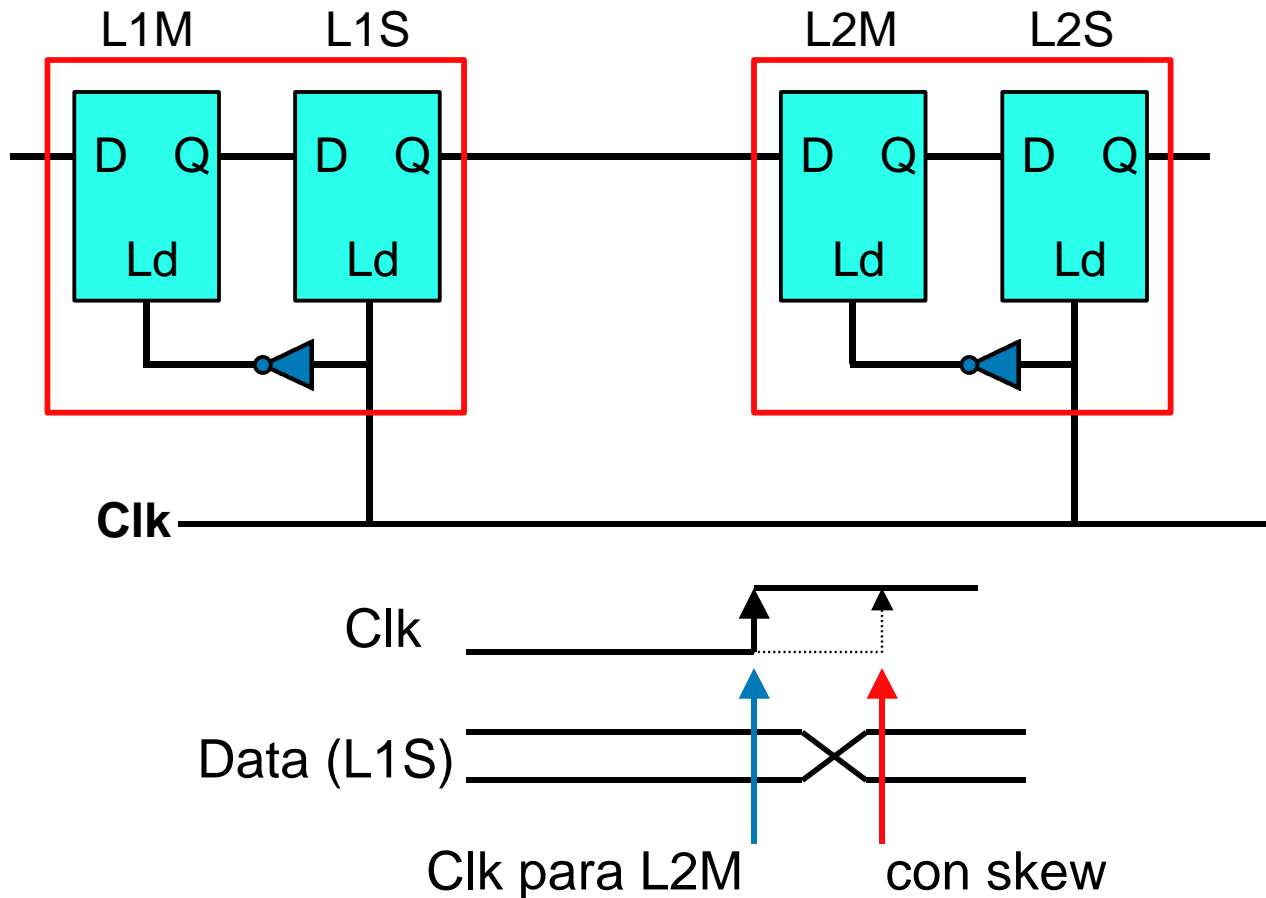
Restricciones debidas al *skew*

- Carrera “race” entre el clock y la propagación de los datos



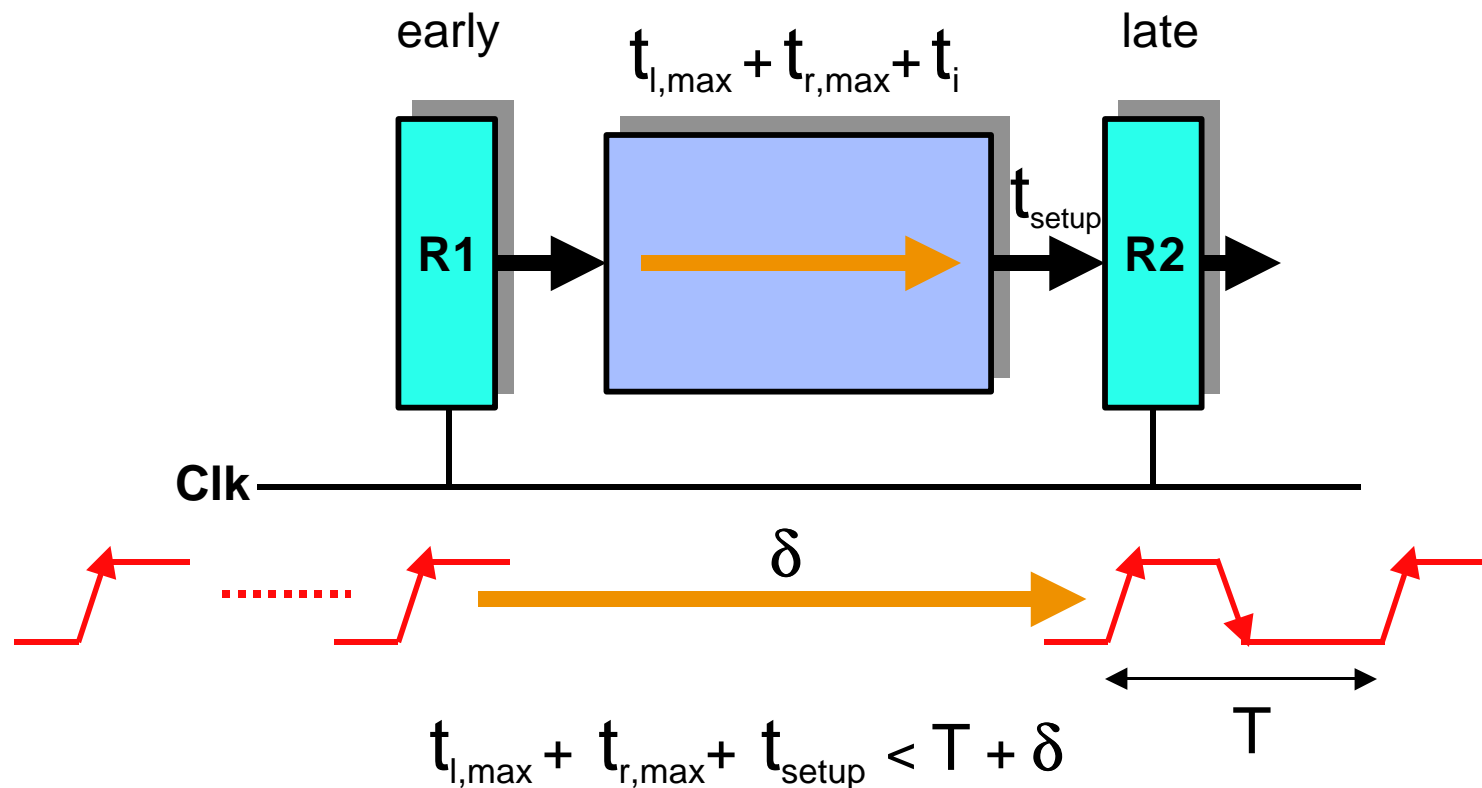
Restricciones debidas al *skew*

- Caso extremo cuando tenemos dos registros consecutivos.



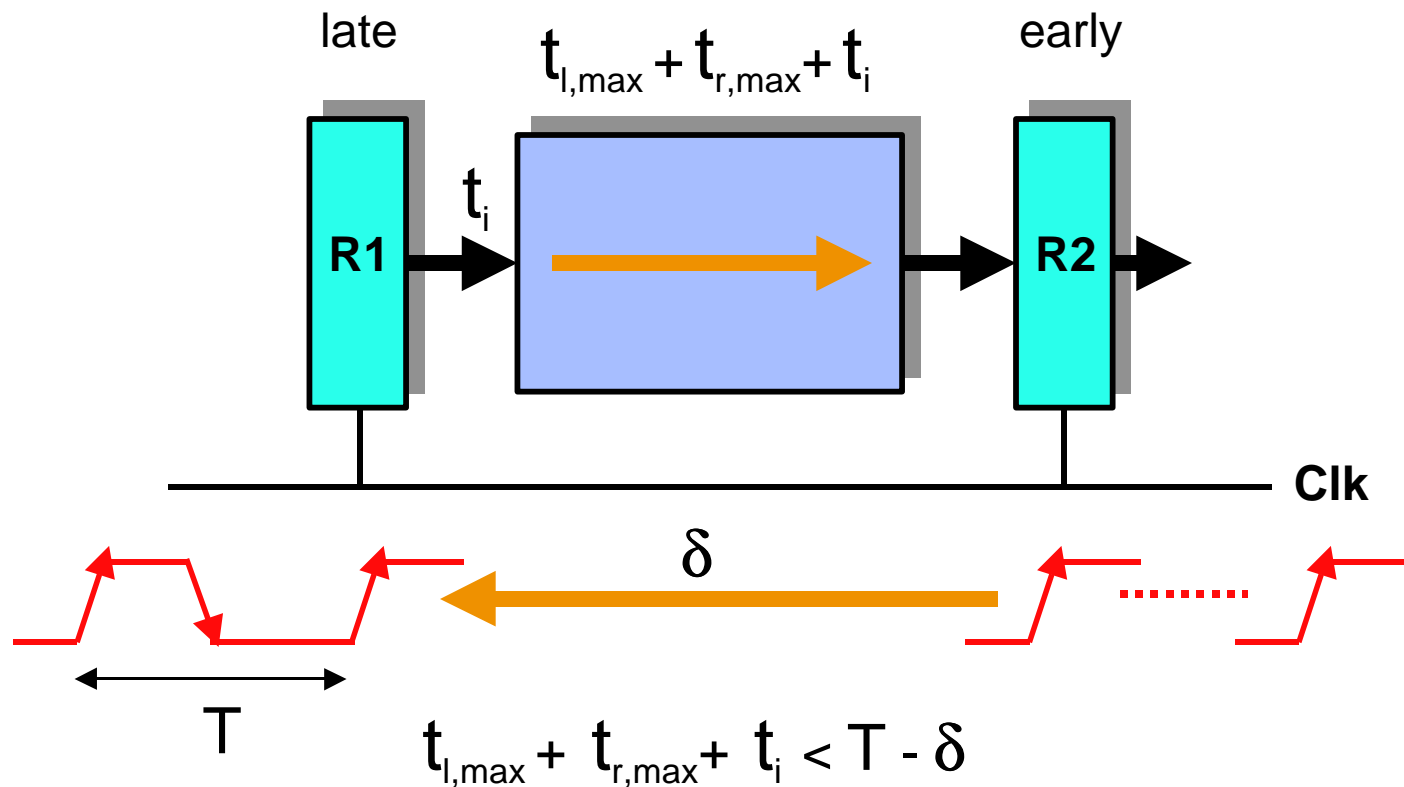
Restricciones debidas al *skew*

- Los datos deben estabilizarse antes del pulso del clock.

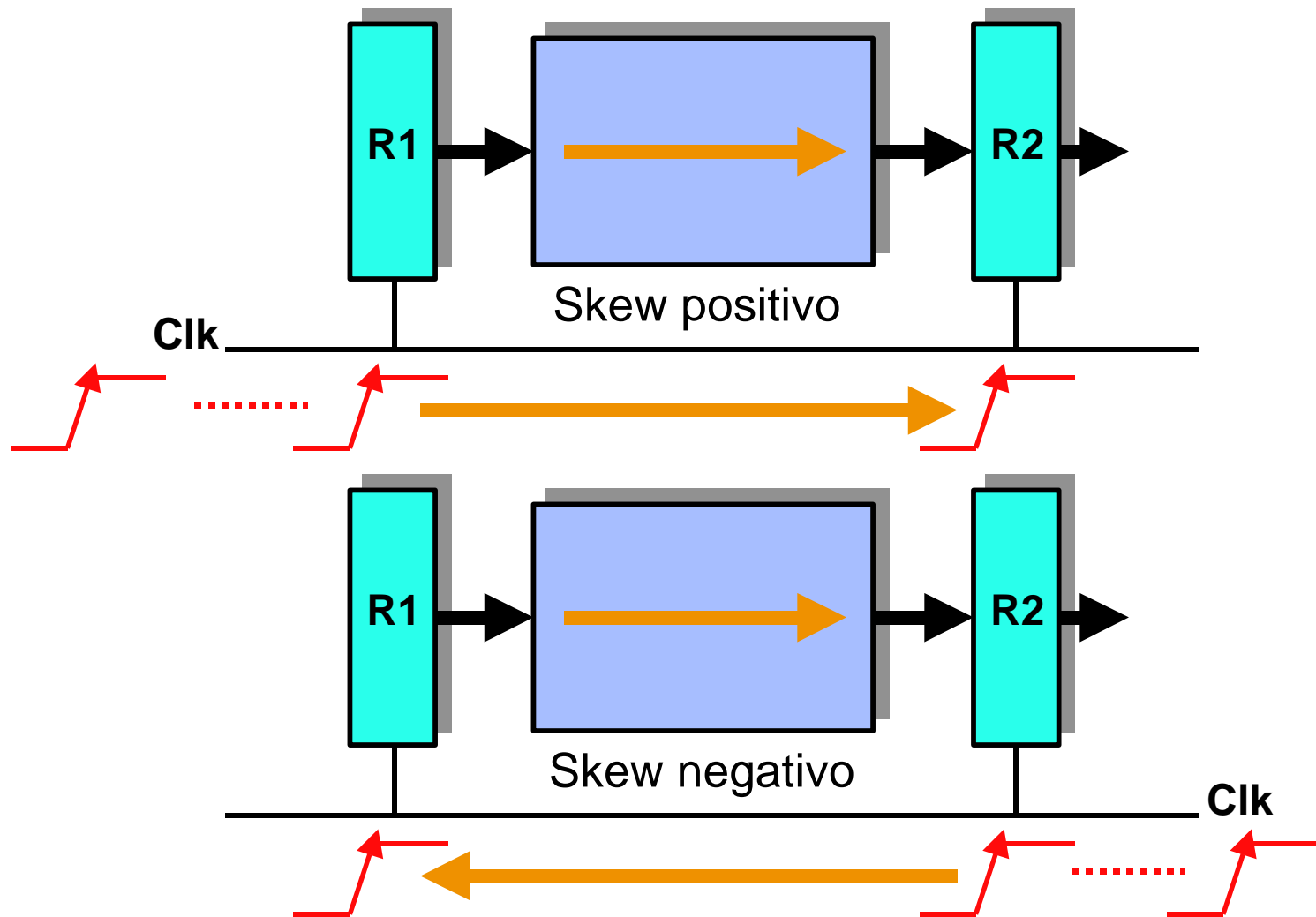


Restricciones debidas al *skew*

- Los datos deben estabilizarse antes del pulso del clock.



Restricciones debidas al *skew*



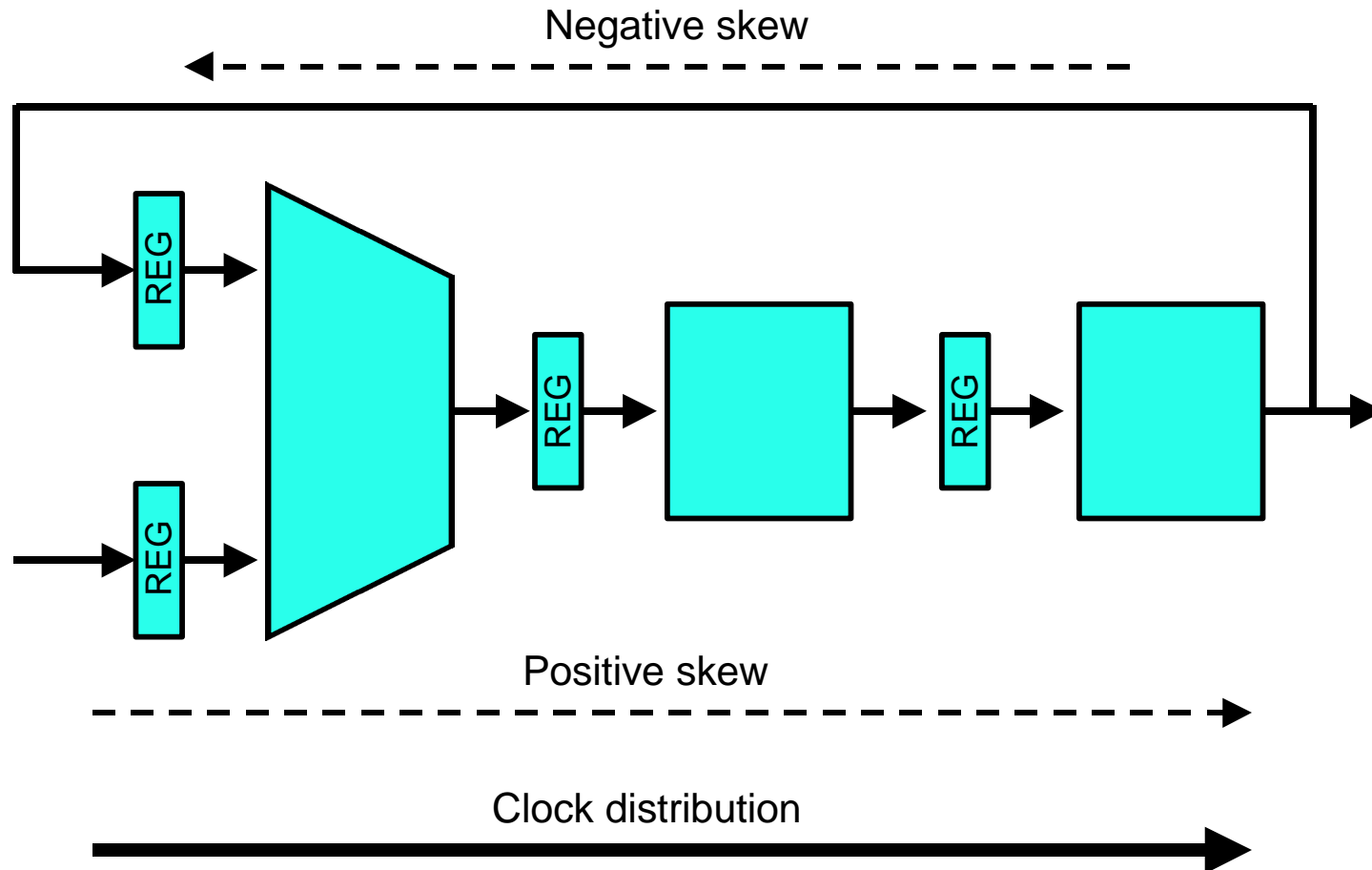
Restricciones debidas al *skew*

- Restricciones en un sistema edge-triggered:

$$t_{l,\min} + t_{r,\min} + t_{\text{setup}} > \delta$$
$$t_{l,\max} + t_{r,\max} + t_{\text{setup}} + \delta < T$$

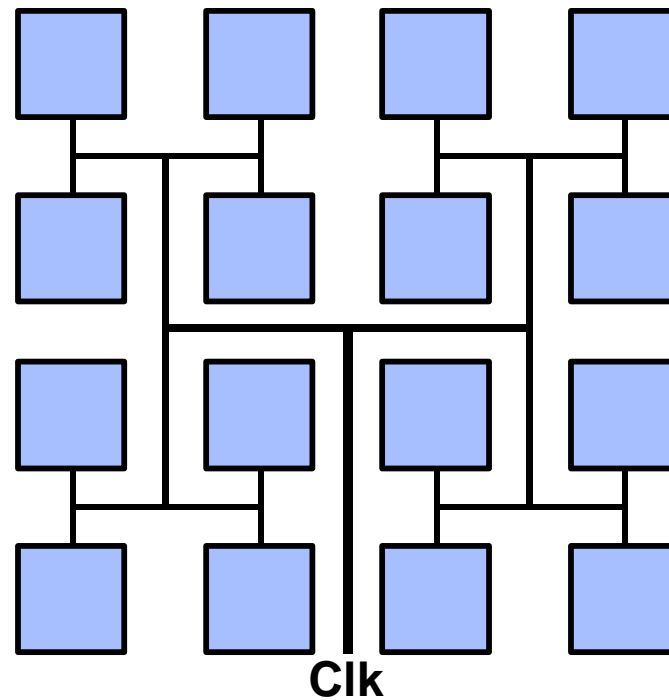
- Clock *skew* máximo:
Determinado por el retardo mínimo entre registros.
- Periodo mínimo del clock:
Determinado por el retardo máximo entre registros.

Restricciones debidas al *skew*



Distribución del clock

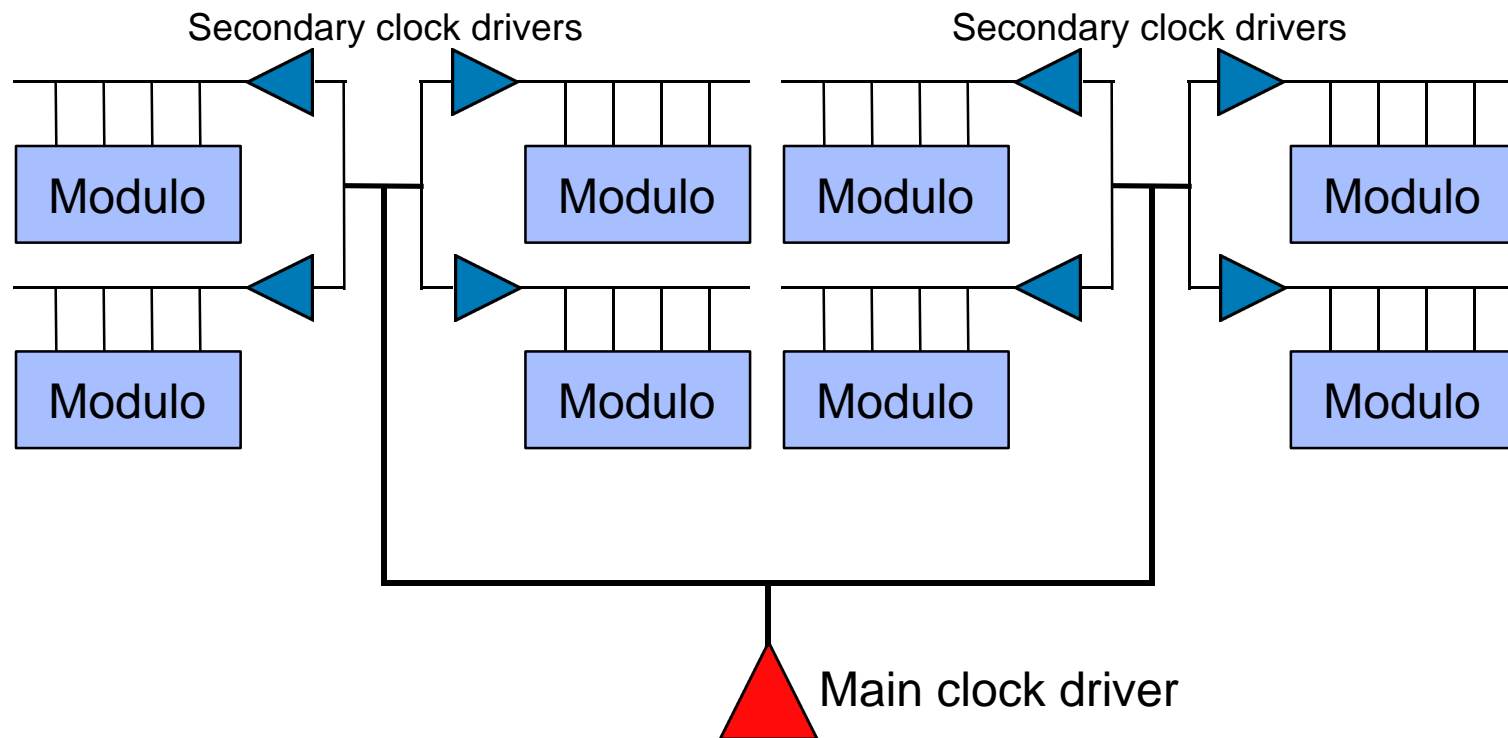
- H-Tree Clock Network:



- Solo el *skew* relativo es relevante :

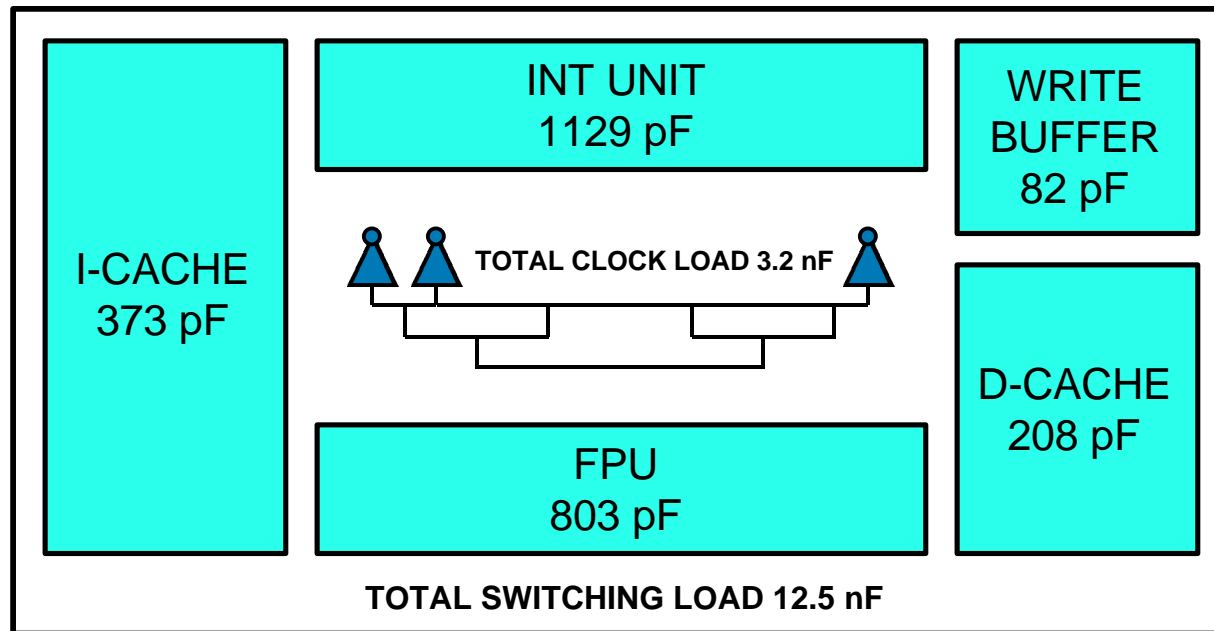
Todos los segmentos deben tener la misma longitud.

Distribución del clock



- Esquema de buffers distribuidos en áreas locales:
 - Reduce el retardo total en cada path.
 - Sensible a las variaciones en el retardo de los buffers.

Ejemplo: DEC Alpha 21164

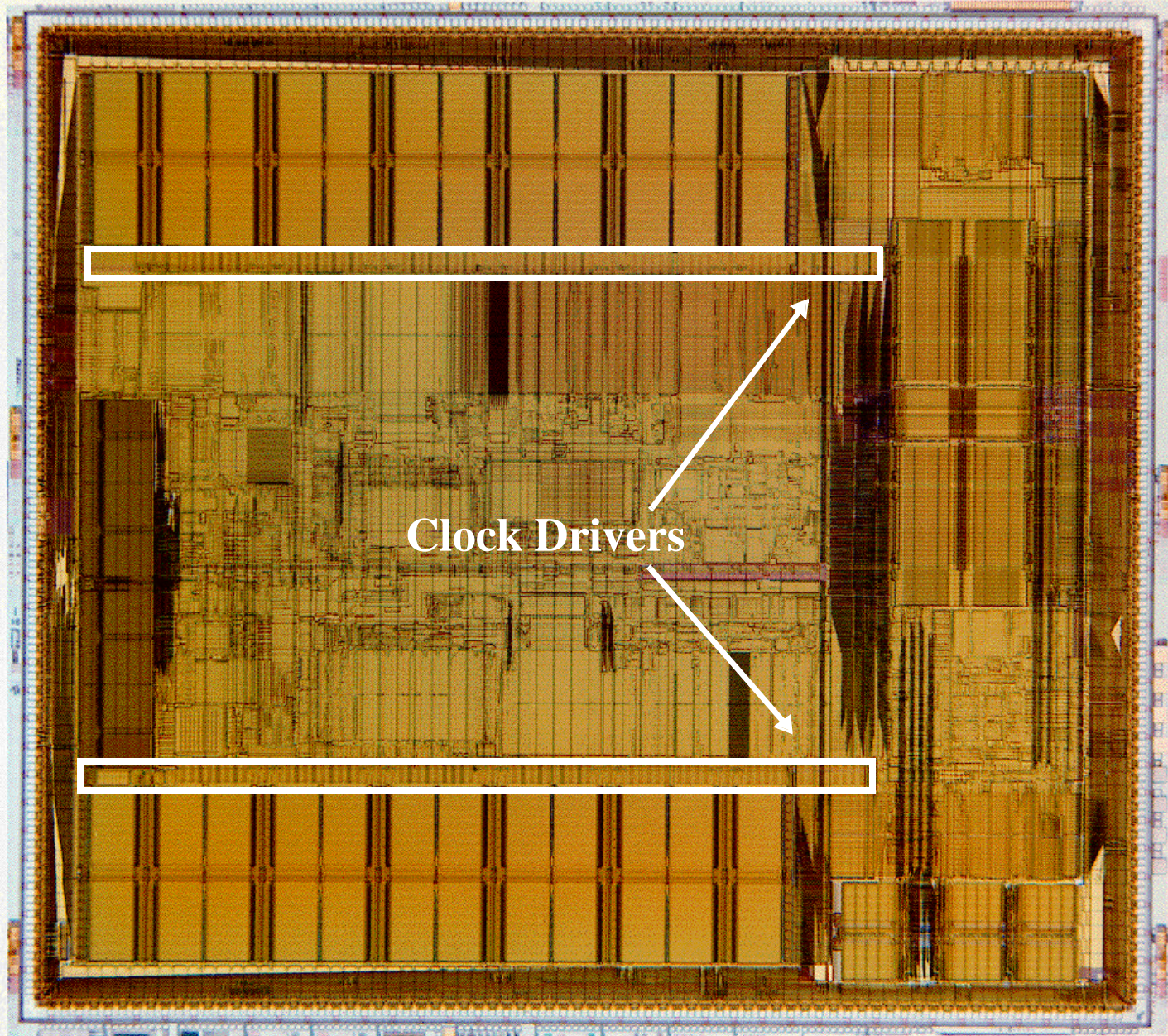


DECOUPLING CAPACITANCE 128 nF USED TO PROVIDE di/dt OF 2×10^{11}

- Los bloques reales complican la distribución en H :
 - La capacidad en los registros depende de la configuración.
 - Variaciones de temperatura y corriente.

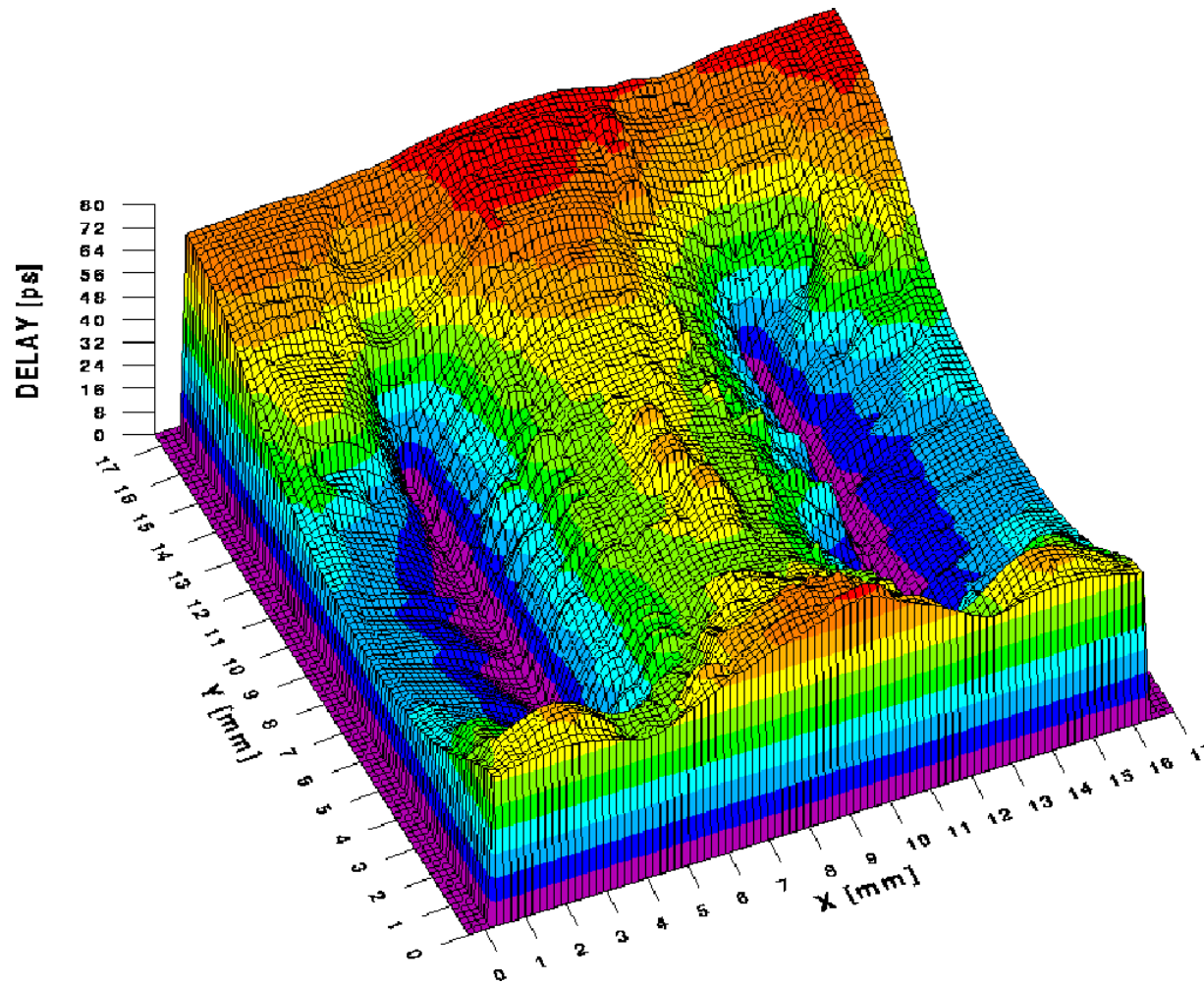
Ejemplo: DEC Alpha 21164

- Clock frequency: **300 MHz** con **9.3** millones de transistores
- Carga total en el clock: **3.75 nF**
- Consumo en la red de distribución: **20W** (de **50W** totales)
- Utiliza dos niveles de distribución:
 - Driver de 6 etapas único en el centro del chip.
 - Buffers secundarios distribuyen el clock en la maya izquierda y derecha (Metal3 y Metal4).
 - Tamaño total de los drivers 58cm.



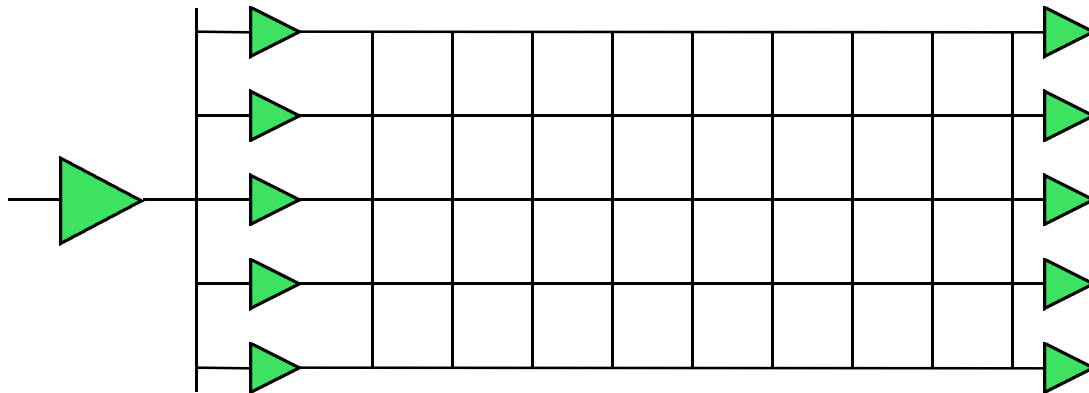
Clock Drivers

Ejemplo: DEC Alpha 21164



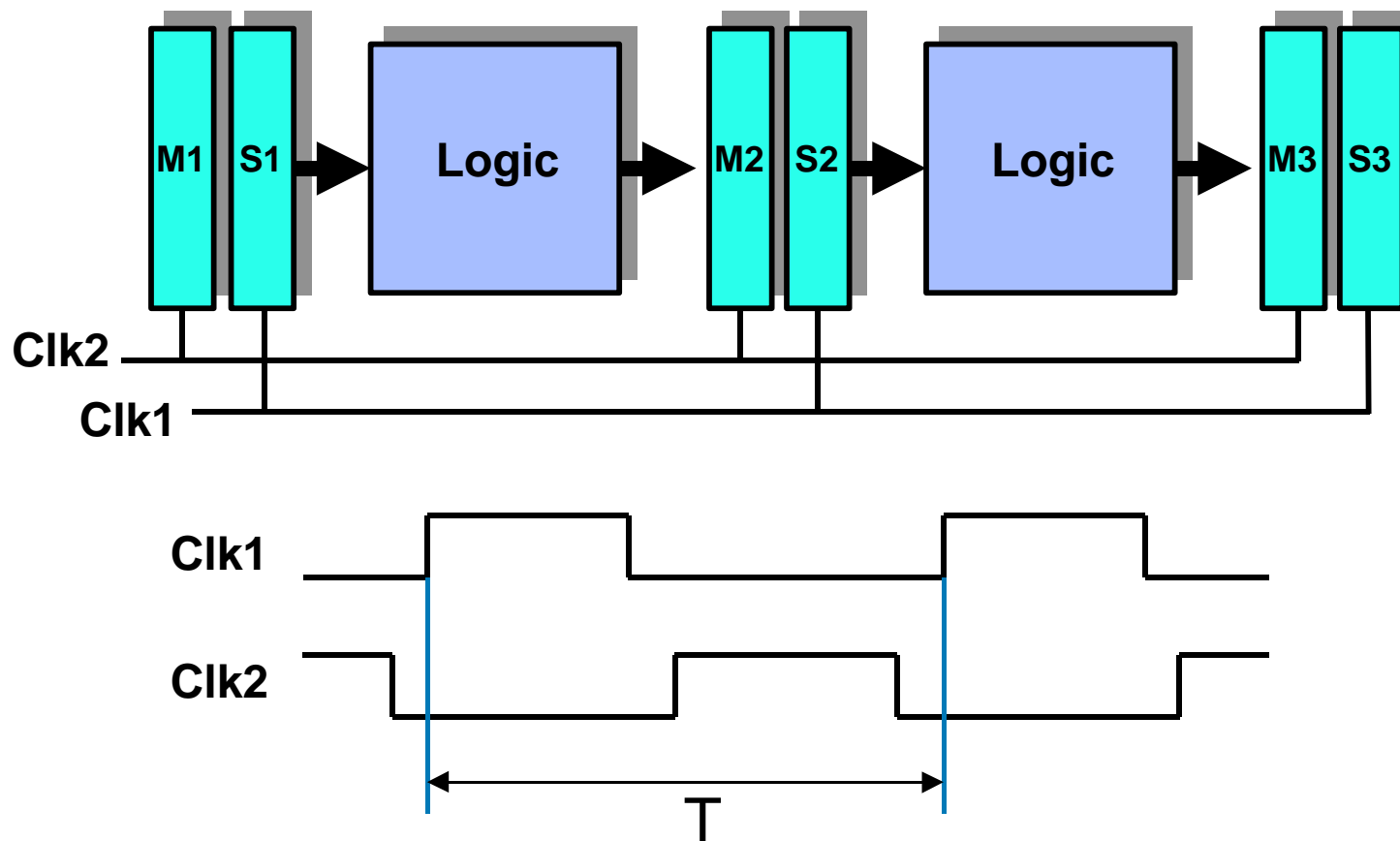
Ejemplo: DEC Alpha 21064

- Estructura en forma de *grid* de la red de distribución:
 - Clock buffer grande: toda la anchura del chip.
 - Estructura en forma de árbol
 - Clocks locales “corto-circuitados” para minimizar las diferencias entre dispositivos y el *skew* local.



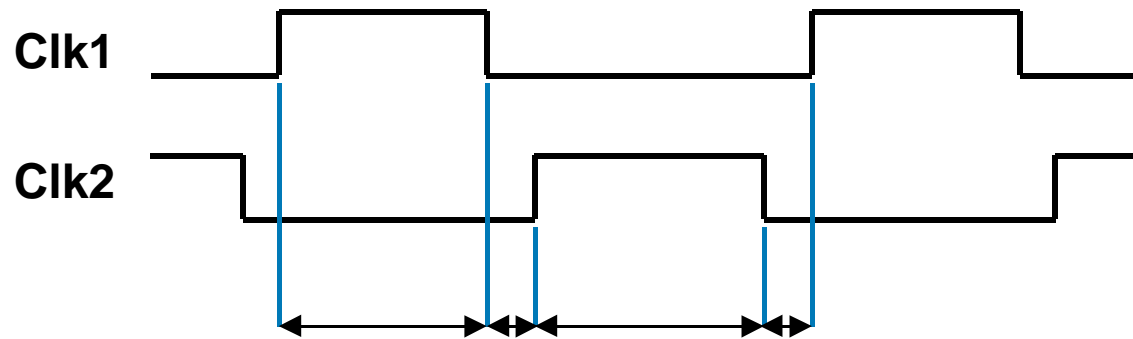
El clock en un sistema “two phase”

- Esquema de clock con dos fases alternadas:



El clock en un sistema “*two phase*”

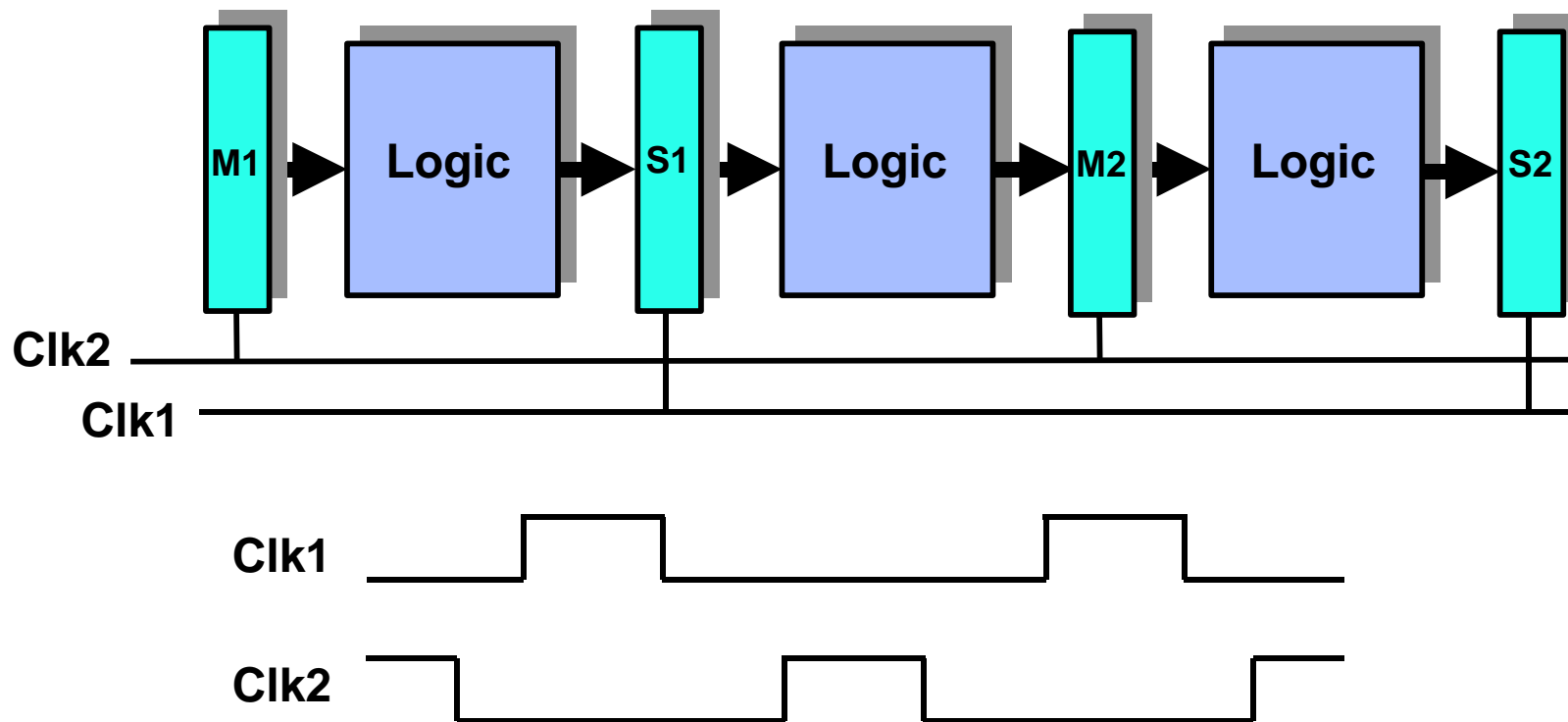
- Flancos distintos guardan los datos y modifican la salida.



- Existen cuatro sub-fases distintas:
 - **Fase 1:** Clk1 activo
 - **Fase 2:** Flanco negativo de Clk1 a flanco positivo de Clk2
 - **Fase 3:** Clk2 activo
 - **Fase 4:** Flanco negativo de Clk2 a flanco positivo de Clk1

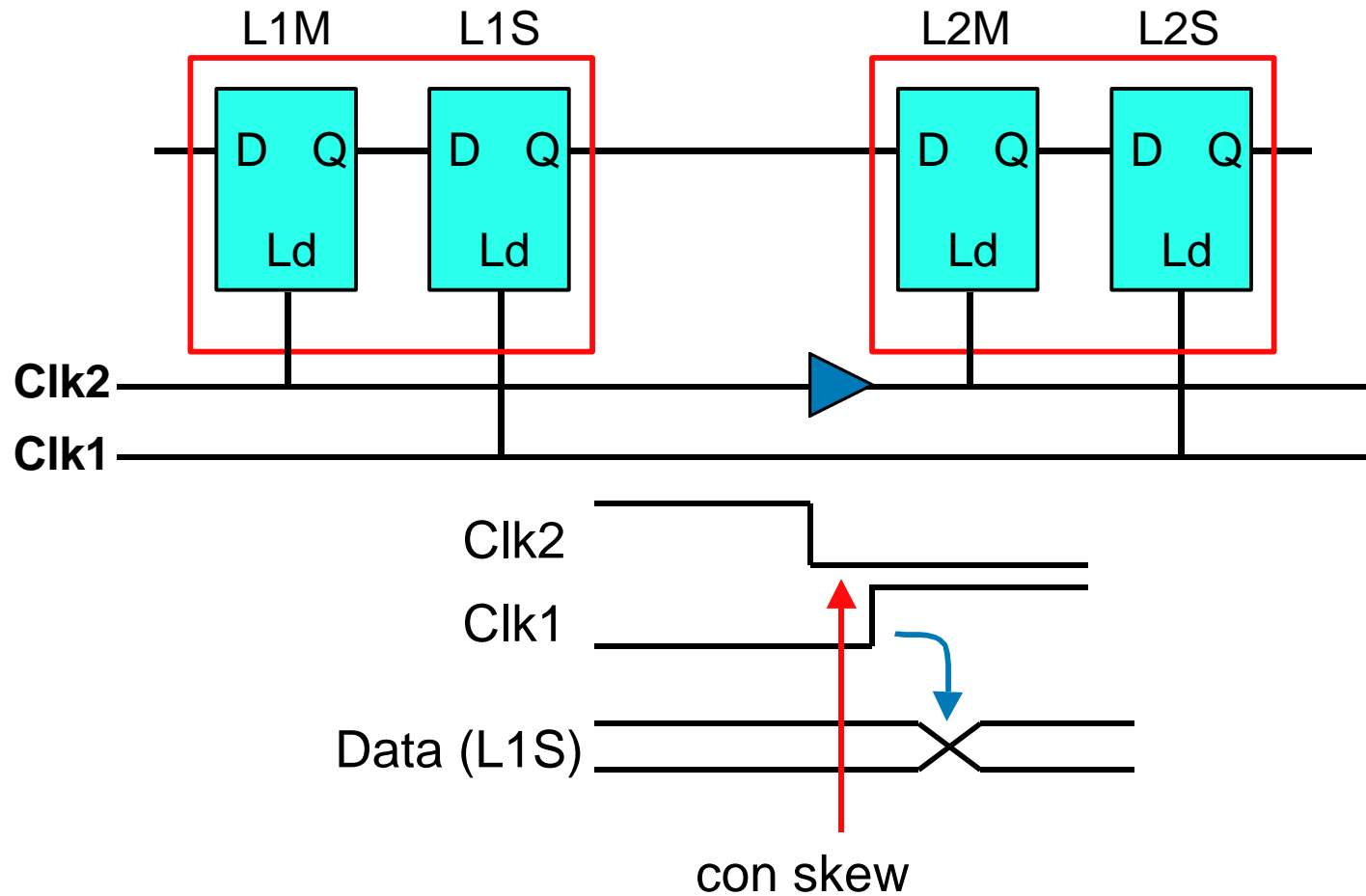
El clock en un sistema “two phase”

- Este esquema permite redistribuir la lógica:

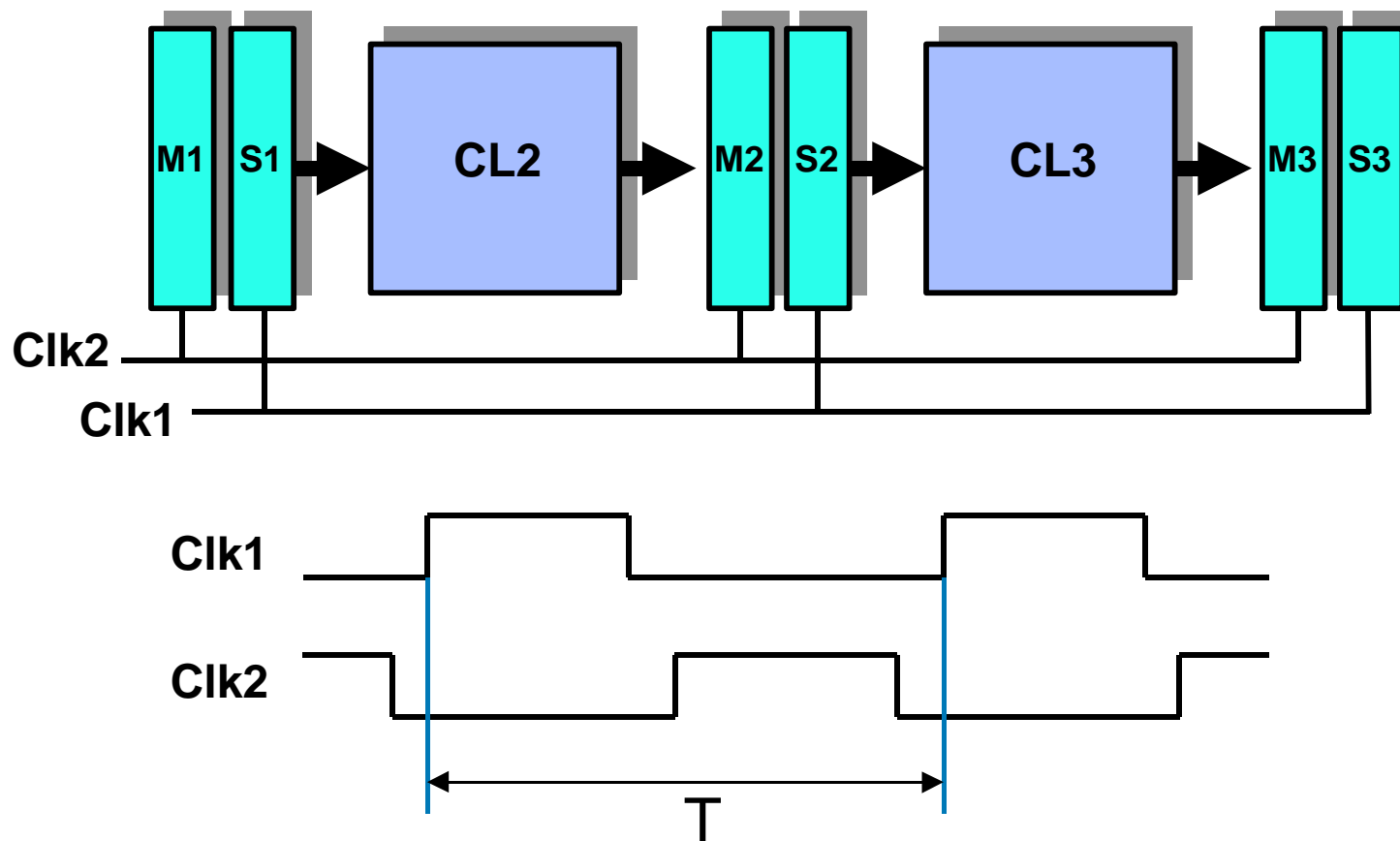


Skew en un sistema “two phase”

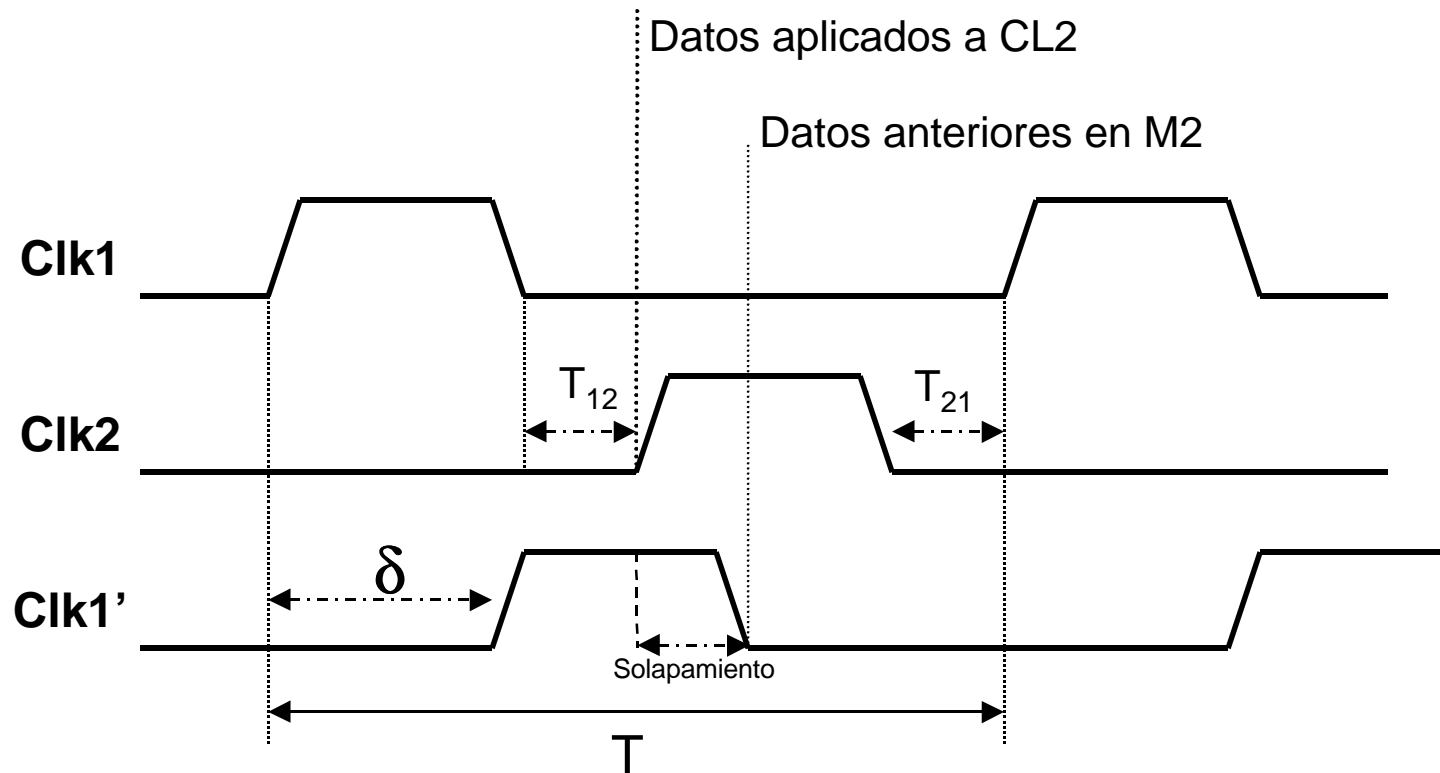
- Caso extremo cuando tenemos dos registros consecutivos.



Skew en un sistema “two phase”



Skew en un sistema “two phase”

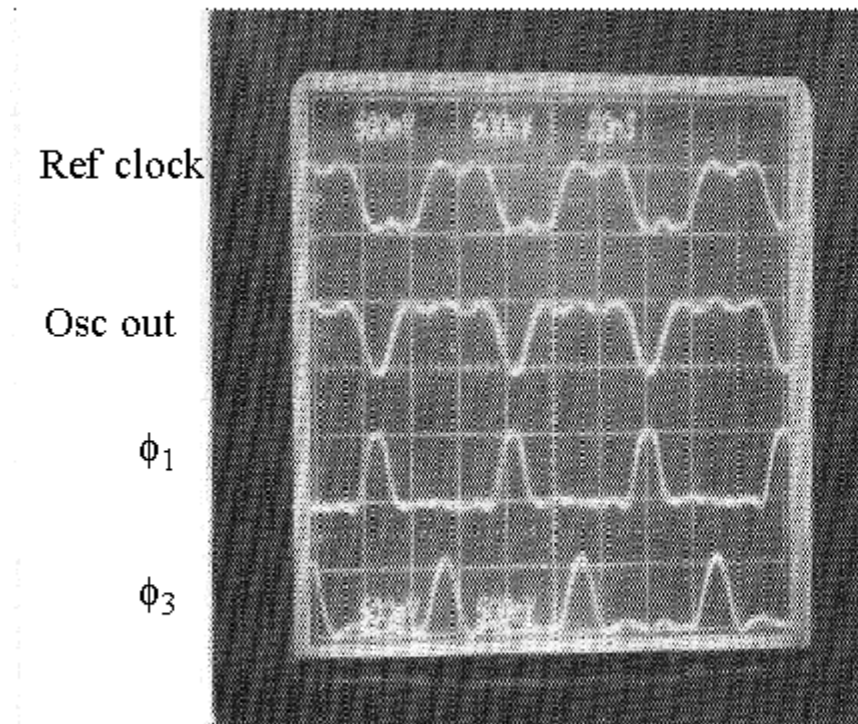


$$t_{\min} > \delta - T_{12}$$
$$t_{\max} > T + \delta - T_{12}$$

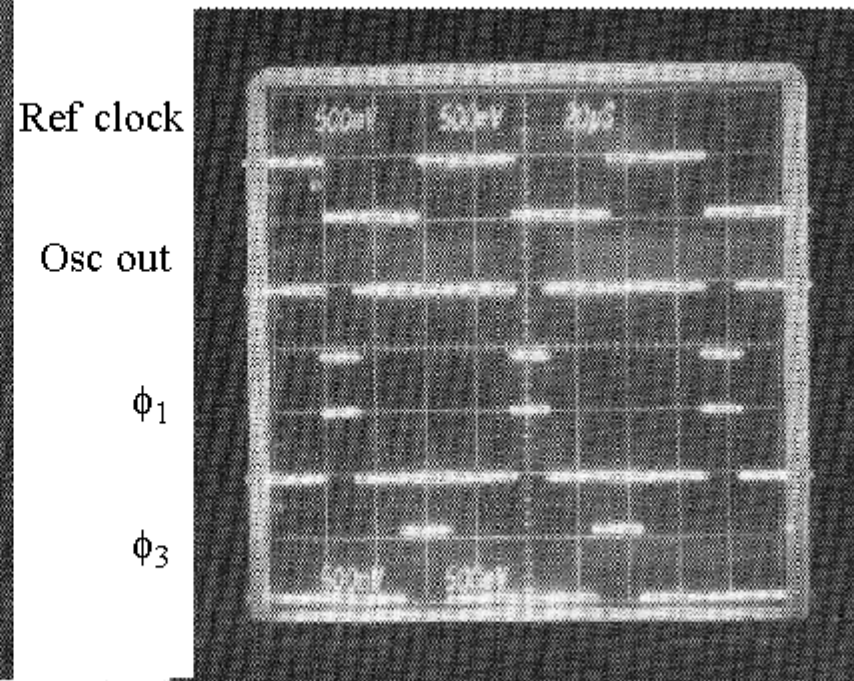
PLL-generated clock

Clock sin PLL

Clock soportado por un PLL



(a) Clock generator output at 18 MHz.



(b) Clock generator output at 15 kHz.