

BLOQUE TEMÁTICO 1**TÍTULO DE LA ACTIVIDAD:** Lectura sobre diseño síncrono**CÓDIGO:**
BT1_A3_L5**FECHA:****NOMBRE:****APELLIDOS:****MODALIDAD:**

Lectura. Individual

TIPO:

Lectura

DURACIÓN:**CALENDARIO:****CRITERIO DE
ÉXITO:**

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:

minutos

AUTOEVALUACIÓN:
[entre 0 y 10 puntos]

No procede

Índice

Parte I. Lectura.....	3
-----------------------	---

Parte I. Lectura

La aplicación de las reglas para el diseño de circuitos síncronos da lugar a restricciones que condicionan el tipo de circuitos que pueden idearse para conseguir una determinada funcionalidad. En muchas ocasiones pueden impedir la realización de la solución funcional encontrada por el diseñador; cuando esto ocurre es necesario encontrar un diseño alternativo que satisfaga las reglas de diseño.

Habilitación de reloj

En un circuito síncrono está prohibido que la salida de un circuito combinacional o de un *flip-flop* pueda conectarse a la entrada de reloj de un *flip-flop*, la cual, por otra parte, ha de estar forzosamente conectada a la señal global de reloj. Esto impide controlar la operación de un circuito secuencial generando un flanco de reloj cuando se da una combinación de valores en un dato o se pasa por determinado estado en un módulo secuencial (figura 33).

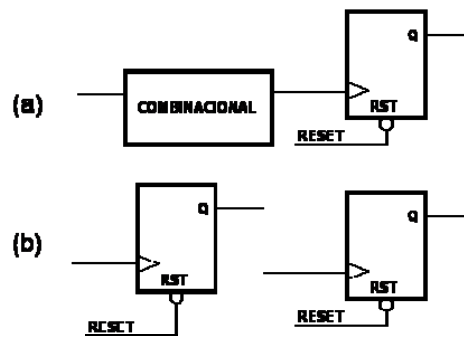


Figura 33. Relojes asíncronos

Las alternativas idóneas para estas situaciones es el empleo de circuitos secuenciales con habilitación de reloj (*clock enable*). En el caso de un *flip-flop* tipo D con *clock enable*, por ejemplo, cuando la entrada de habilitación está activada se captura el dato de entrada, en caso contrario se mantiene el dato almacenado independientemente del nivel lógico presente en la entrada D del *flip-flop*. En la figura 30 se ilustra el modo de operación de este tipo de *flip-flops*.

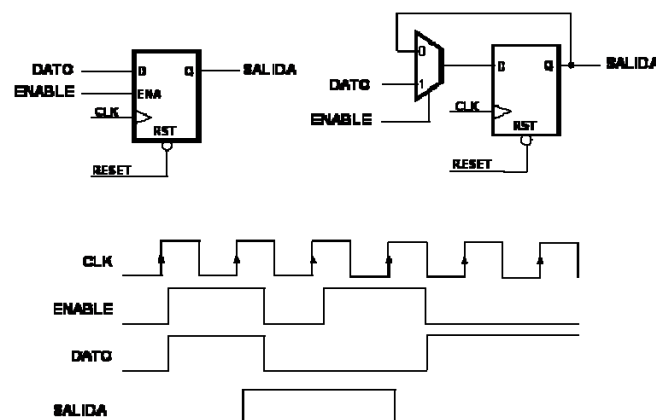


Figura 34. *Flip-flop* con habilitación de reloj

La versión síncrona del circuito de la figura 33a se muestra en la figura 35, donde se compara además el modo de funcionamiento en ambos casos.

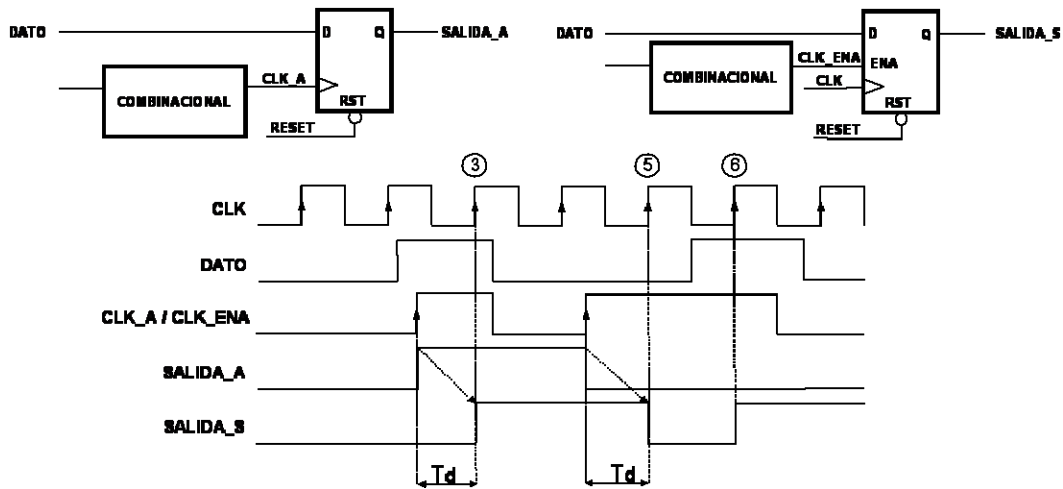


Figura 35

En el cronograma de la figura 35 puede observarse que si la salida del circuito combinacional está activa en un solo flanco (como en el número 3 del cronograma) ambos circuitos son funcionalmente equivalentes, con la diferencia de que en la versión síncrona la salida no cambia hasta que llega el flanco de reloj –esto es lógico, es consecuencia del hecho de que en un sistema síncrono el retardo “efectivo” de cualquier módulo combinacional es de un ciclo de reloj. Esta equivalencia desaparece si la salida del combinacional está activa durante varios flancos (en el cronograma del ejemplo, en los flancos 5 y 6); cuando esto ocurre, para obtener un comportamiento análogo al de la versión asíncrona, es necesario modificar la salida del circuito combinacional para que sólo esté activa en un flanco de reloj. El circuito que suele utilizarse para realizar este acondicionamiento se muestra en la figura 36.

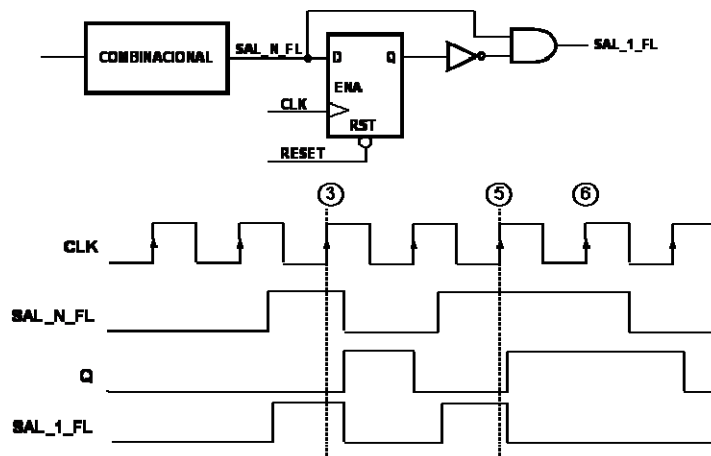


Figura 36. Conformación de pulsos de habilitación de reloj

Este circuito genera, sin retardar la actividad del circuito combinacional, un pulso que está activo un solo flanco de reloj, independientemente de la duración del pulso de salida del circuito combinacional. Intercalándolo en la versión síncrona del circuito, se obtiene ahora un funcionamiento completamente equivalente –que se muestra en el cronograma de la figura 37.

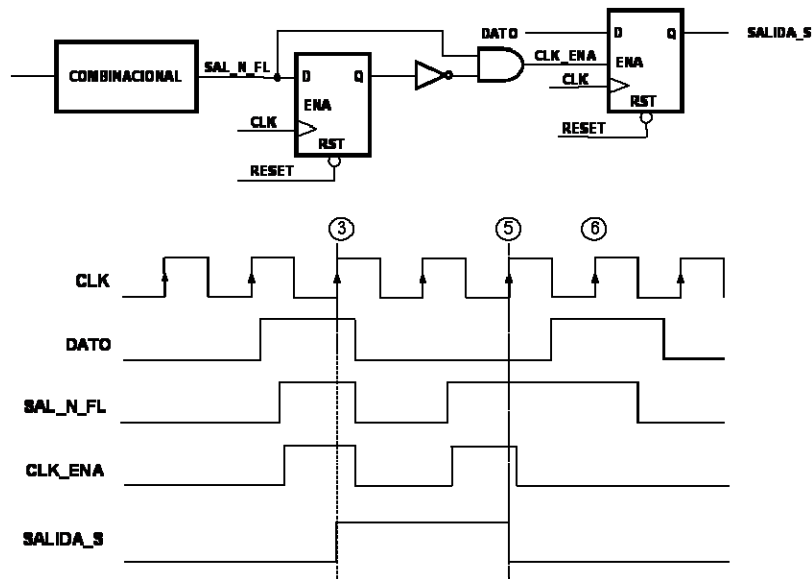


Figura 37

Reset y Preset síncronos

Mientras un circuito digital está funcionando resulta, ocasionalmente, necesario inicializar los *flip-flops* de algunos módulos secuenciales. La forma más simple de fijar un estado lógico en un *flip-flop* consiste en actuar sobre sus entradas de inicialización asíncronas de *reset* o *preset*. Este tipo de acción está prohibida por las reglas de diseño síncrono, salvo para el caso de la inicialización del circuito por la activación del *Reset Global*. Para esta función deben utilizarse *flip-flops* con entradas síncronas de *reset* y/o *preset*. Este tipo de *flip-flops* pueden construirse combinando *flip-flops* tipo D con puertas lógicas.

Desde el punto de vista funcional, la diferencia entre inicializar asíncrona o síncronamente un *flip-flop* radica en que en la acción asíncrona el efecto de la actividad de la entrada es instantáneo la inicialización, mientras que en la síncrona la inicialización se verifica en el flanco activo de reloj. En la figura 38 se ilustra la diferencia entre ambos modos de funcionamiento.

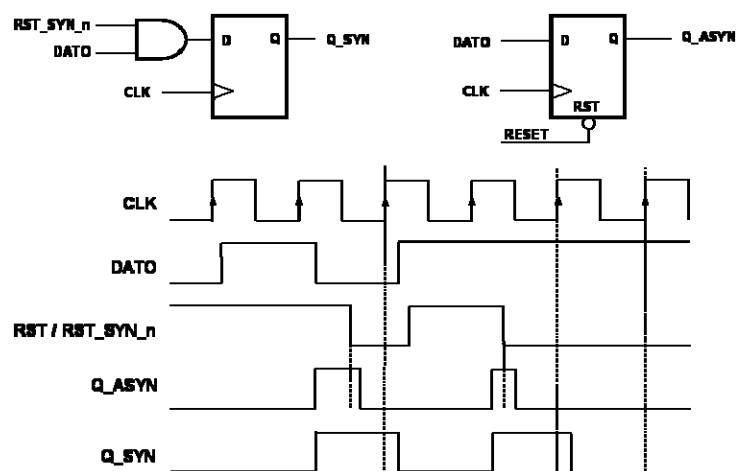


Figura 38. Inicialización síncrona y asíncrona

En el cronograma puede constatarse que la versión síncrona es equivalente a la asíncrona, independientemente de la duración de la señal de *reset*. El retardo derivado de la actuación del *reset* síncrono en los flancos de reloj puede ocasionar problemas que hay que considerar a la hora de calcular la temporización de un sistema síncrono.

Sincronización de Entradas

Las reglas de diseño síncrono exigen que deba registrarse cualquier entrada asíncrona del circuito (cualquier señal manejada desde fuera del dominio del reloj del sistema síncrono) con el fin de que sus conmutaciones queden sincronizadas con el reloj del circuito. El cumplimiento de esta regla conlleva una serie de problemas que vamos a analizar en detalle en este apartado.

En los comentarios de la figura 23 (que se repite aquí para comodidad del lector) se describían dos de los efectos derivados del registro de las entradas asíncronas: los cambios de estado en las entradas se detectan, en el circuito síncrono, con cierto retardo –de valor aleatorio y acotado por el valor del periodo de reloj– y la forma de la señal (la duración de sus niveles lógicos) se modifica, ajustándose a un número entero de periodos de reloj. En el caso de que la velocidad de respuesta del sistema a los cambios en las entradas sea un requisito de funcionamiento, o cuando sea necesario evaluar la duración de los estados de la entrada con gran precisión, los efectos de la sincronización pueden repercutir seriamente sobre el diseño del circuito; pueden requerir, en concreto, que la frecuencia mínima de funcionamiento del circuito sea muy grande, un requisito que, quizás, pueda ser difícil de alcanzar si la lógica de procesamiento tiene asociados retardos de propagación altos.

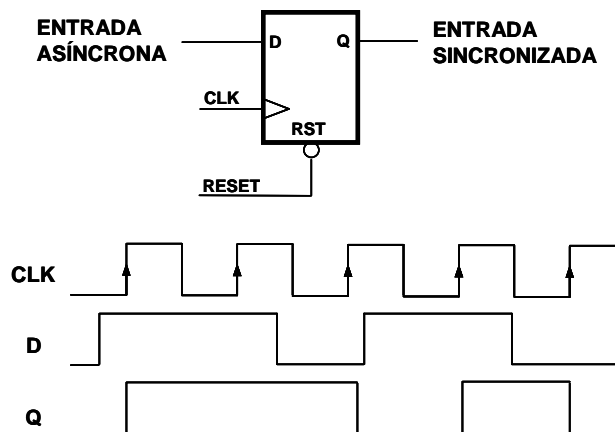


Figura 23 (repetida)

Otro problema derivado de la necesidad de sincronizar las entradas es el que nos encontramos cuando los niveles lógicos de una entrada asíncrona tienen –o pueden a veces tener– una duración menor que el periodo del reloj del circuito síncrono –bien porque sean salidas de otro sistema síncrono que funciona con una frecuencia de reloj mayor, o bien porque su duración se derive del mecanismo particular de generación de la entrada en cuestión. Este caso se ilustra en la figura 39. Como puede verse, la detección de pulsos es aleatoria –depende de si coinciden con los flancos de reloj– y no es posible, en general, discriminar la ocurrencia de trenes de pulsos.

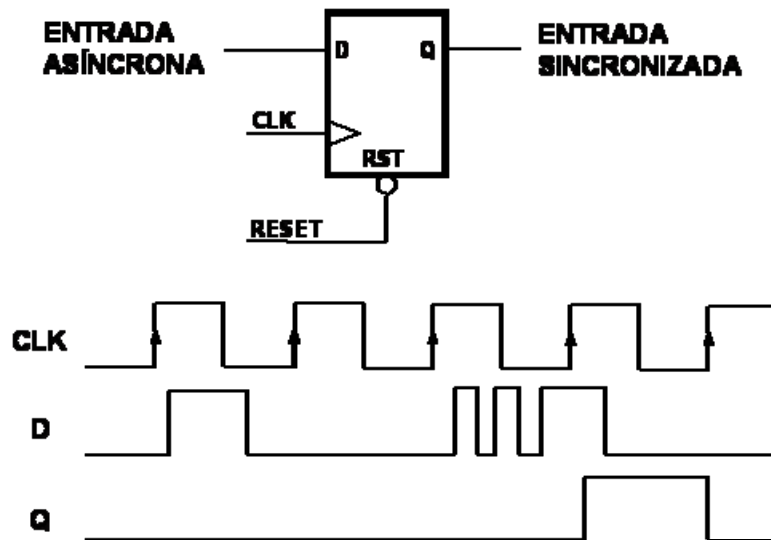


Figura 39. Pulsos de duración menor que un periodo de reloj

Una solución parcial, aunque suficiente en la mayoría de los casos, para este problema se muestra en la figura 40.

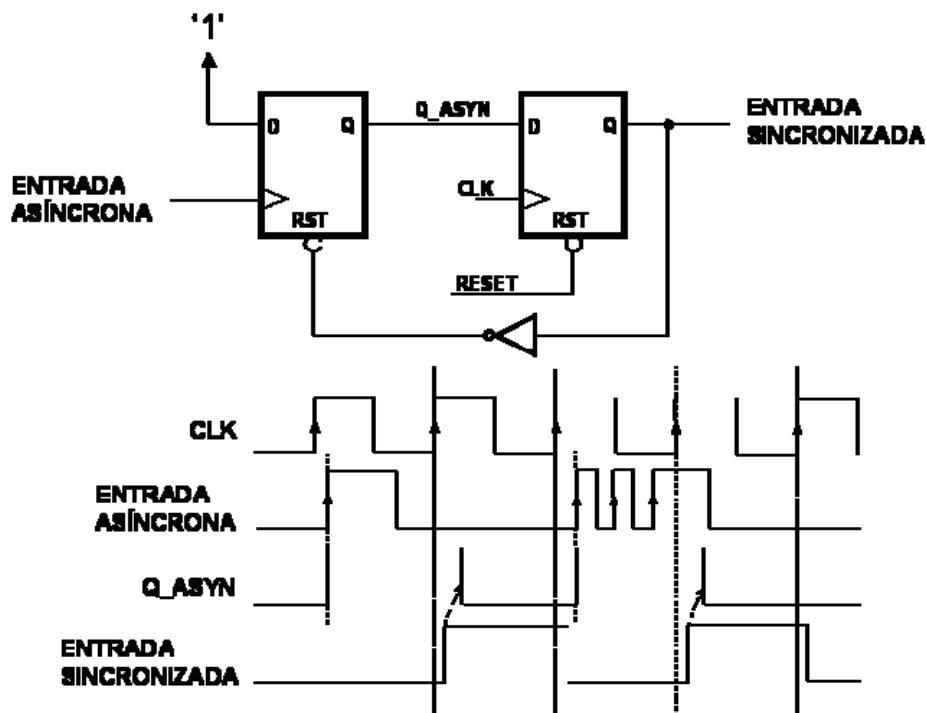


Figura 40. Sincronización de señal de duración menor que TCLK

Es evidente que este circuito viola flagrantemente las reglas de diseño síncrono: el reloj del primer *flip-flop* no es el reloj global del sistema y, además, su reset asíncrono es manejado por la salida del segundo *flip-flop*. Esta infracción puede justificarse con dos argumentos: el primero, de carácter formal, es que podemos considerar que el primer *flip-flop* no es parte del circuito síncrono, sino que es una especie de adaptador externo que acondiciona la señal de entrada asíncrona e interacciona con el sistema por medio de una salida síncrona y registrada de éste que actúa sobre su *reset* asíncrono; el segundo argumento, más prosaico, pero más interesante, es que este circuito, aunque no se atiene a las reglas de diseño, funciona y nos permite realizar el resto del sistema sin perder las ventajas que

acarrea el uso de las técnicas de diseño síncrono, porque es una solución aislada (su uso está limitado a solucionar el problema de sincronización) con una interfaz hacia el circuito (la salida del *flip-flop* manejado por el reloj global) que satisface los requisitos de operación síncrona (disponemos de una entrada sincronizada).

Esta segunda justificación pone de manifiesto un hecho del que se debe ser consciente: en ocasiones resulta inevitable infringir las reglas de diseño síncrono -para realizar interfaces con elementos externos al sistema (memorias asíncronas, buses asíncronos, etc.); esto debe ser considerado por el diseñador como un “último recurso”, pero un recurso al fin y al cabo. Cuando tenga que recurrir a este tipo de soluciones deberá asegurarse de que el bloque asíncrono quede aislado del resto del sistema, permitiendo que el análisis y diseño de la mayor parte del circuito pueda realizarse de acuerdo con la metodología de diseño síncrono.

Debe evitarse también el uso de soluciones asíncronas donde existan versiones síncronas que respeten las reglas de diseño, aunque le parezca que la solución asíncrona es más simple o ahorra recursos lógicos; el circuito de la figura 40, por ejemplo, sólo debe ser usado si es necesario sincronizar una entrada asíncrona con pulsos cuya duración sea menor que un periodo de reloj, y nunca para resolver ninguna otra funcionalidad que pueda ser necesario incorporar al sistema síncrono.

Centrándonos en cómo funciona el circuito de la figura 40, debemos notar que garantiza que la salida del primer *flip-flop* (que es aún una señal asíncrona), va a estar activa, tras un pulso en la señal asíncrona, cuando llegue el flanco de reloj al segundo *flip-flop* (que es el que propiamente sincroniza la entrada), pero no permite distinguir la ocurrencia de más de un pulso en la entrada durante un ciclo de reloj.

Si se desea que cada pulso de entrada se corresponda con otro, síncrono y de duración igual a un ciclo de reloj, la frecuencia de la entrada asíncrona ha de ser menor o igual a la mitad de la frecuencia del reloj del sistema síncrono (figura 41).

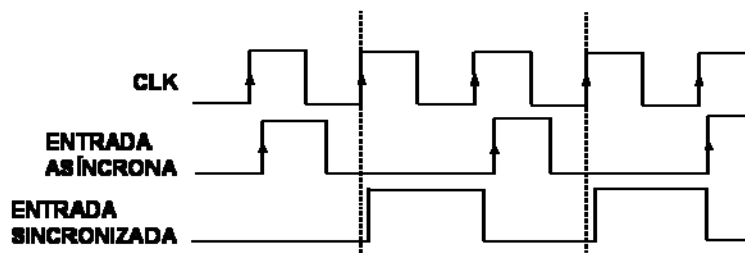


Figura 41. Funcionamiento de la interfaz asíncrona

El último problema derivado de la sincronización es que existe el riesgo de que los cambios de estado de las entradas violen los tiempos de *set-up* o *hold* del *flip-flop* de sincronización. Este *flip-flop* está controlado por el reloj del circuito síncrono, y las entradas asíncronas pueden conmutar en cualquier instante de tiempo, luego existe cierta probabilidad de que el cambio de nivel ocurra en la ventana de tiempo delimitada por el tiempo de *set-up* y *hold* en torno al flanco activo de reloj (figura 42).

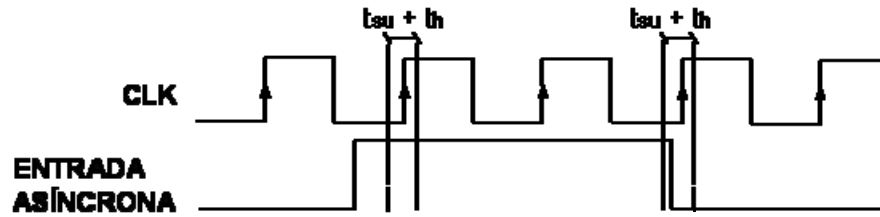


Figura 42. Sincronización (1)

La probabilidad de que no se respeten los tiempos del *flip-flop* es directamente proporcional a la frecuencia de reloj del circuito –la ventana de tiempos a respetar es más grande respecto al periodo de reloj cuanto menor sea éste y, en consecuencia, es más probable que el instante de conmutación de la entrada se produzca dentro de ella- y a la frecuencia de conmutación de la entrada asíncrona – lógicamente, cuantas más veces conmute la entrada, más posibilidades habrá de que alguna vez lo haga en la ventana de tiempos. Cuando no se respetan los tiempos del *flip-flop* podemos encontrarnos con dos respuestas diferentes. La primera y más frecuente es que el *flip-flop* establezca en su salida, con un retardo compatible con su tiempo de propagación, un nivel lógico estable que se corresponda con el de conmutación de la entrada o el anterior a ésta (figura 43).

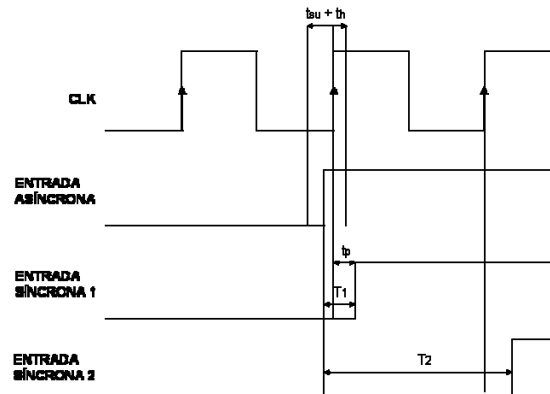


Figura 43. Sincronización (2)

Ninguno de los casos ilustrados en la figura 43 resulta problemático, ya que en ambos se detecta, aunque con diferente retardo, el cambio de estado. La situación alternativa, más infrecuente y grave, es que el *flip-flop* entre en un estado de metaestabilidad. En metaestabilidad la tensión de la salida del *flip-flop* se establece en una zona intermedia a la correspondiente a los niveles lógicos válidos (cero y uno), dando lugar a una indeterminación lógica (no es ni un '0' ni un '1') y ocasionando un aumento sustancial en el consumo de potencia del *flip-flop*.

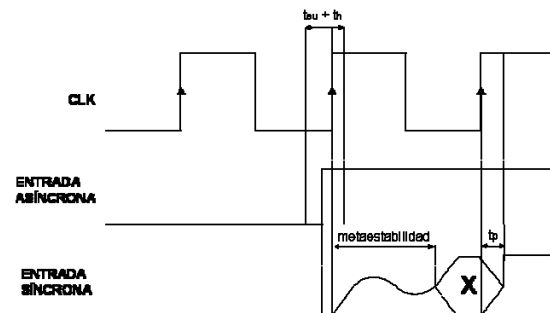


Figura 41

Cuando la salida de un *flip-flop* alcanza un estado metaestable termina, al cabo de un tiempo denominado tiempo de recuperación (*recovery time*), t_R , indeterminado pero susceptible de ser caracterizado probabilísticamente, por bascular aleatoriamente (figura 44) a un estado estable ('0' ó '1') –en caso de que esto no ocurra pasará a '0' ó '1' cuando llegue un flanco activo de reloj con las entradas síncronas estables.

La metaestabilidad puede propagarse a elementos del circuito conectados a la salida del *flip-flop* de sincronización. Si se trata de lógica combinacional, el “contagio” es inmediato, en el caso de que se trate de entradas síncronas de *flip-flops*, se produce si llega un flanco activo de reloj. En ambos casos los efectos perniciosos de la metaestabilidad se extienden por el circuito, pudiendo producir errores funcionales o, incluso, dar lugar a un deterioro del *hardware* por el exceso de consumo.

Desde un punto de vista práctico, la entrada de un *flip-flop* en estado metaestable se manifiesta como un incremento del tiempo de propagación síncrono del *flip-flop* –es decir, del tiempo de propagación desde la entrada de reloj a la salida. Este incremento, que es el tiempo de recuperación, t_R , al que antes nos hemos referido, tiene un valor aleatorio que en la mayoría de los casos es muy pequeño, aunque, eventualmente, puede tomar valores que produzcan fallos en el circuito –para que se produzca un fallo es necesario que el estado metaestable se prolongue durante un ciclo completo de reloj.

La probabilidad de que el *flip-flop* entre en metaestabilidad, y una vez en ella cause un fallo, debe ser tan baja como exijan los requisitos del circuito que se está desarrollando –evidentemente no tiene la misma importancia que fallen, por ejemplo, circuitos digitales en un satélite o que lo hagan los del mando a distancia de un televisor. En relación con este problema, lo primero que hay que tener claro es que aunque resulta imposible garantizar absolutamente la inmunidad de un circuito frente al problema de la metaestabilidad, sí es posible reducir la probabilidad de fallo a niveles despreciables. Los factores de los que depende la probabilidad de fallo son:

1. La frecuencia del reloj del circuito y la frecuencia media de conmutación de la entrada asíncrona –cómo ya se justificó anteriormente.
2. La tecnología empleada en la realización del sistema y las condiciones ambientales de funcionamiento.

Los fabricantes de chips ofrecen gráficas, tablas y fórmulas que permiten calcular el tiempo medio entre fallos (MTBF) de un circuito en función del tiempo de recuperación que tolere. La figura 45, por ejemplo, es una gráfica de ALTERA que caracteriza el tiempo medio entre fallos para dos de sus familias de PLDs y para diferentes frecuencias de reloj en función del tiempo de recuperación que pueda tolerar el circuito (t_{MET} , en la figura). De acuerdo con la gráfica, en un circuito síncrono con una frecuencia de reloj de 40 MHz, en el que se pueda tolerar, cuando surja un problema de metaestabilidad, un incremento en el tiempo de propagación del *flip-flop* de sincronización de 4 ns, el tiempo medio entre fallos (MTBF) es de 100 años.

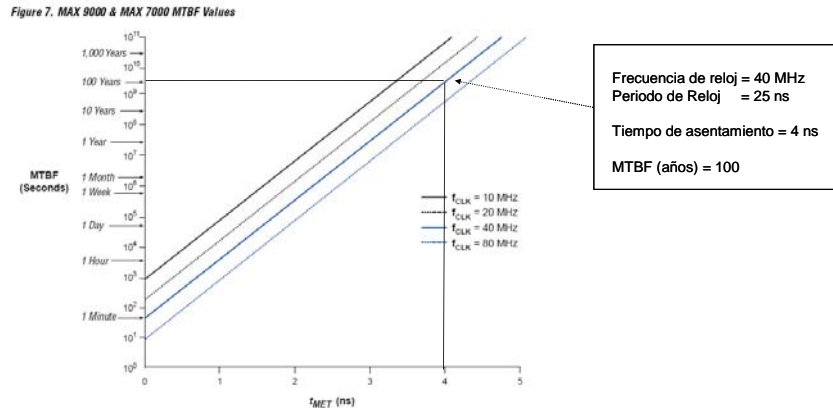


Figura 45. Gráfica del MTBF de ALTERA

La fórmula y la tabla de la figura 46 nos permiten realizar un cálculo del MTBF para diferentes familias de dispositivos lógicos programables de ALTERA –la mayor parte de los fabricantes dan una información análoga a la que se muestra aquí.

$$MTBF = \frac{e^{(C_2 \times t_{MET})}}{C_1 \times f_{CLOCK} \times f_{DATA}}$$

Table 1. Metastability Equation Constants		
Device	C ₁	C ₂
FLEX 10K	1.01 × 10 ⁻¹³	1.268 × 10 ¹⁰
FLEX 8000	1.01 × 10 ⁻¹³	1.268 × 10 ¹⁰
FLEX 6000	1.01 × 10 ⁻¹³	1.268 × 10 ¹⁰
MAX 9000	2.98 × 10 ⁻¹⁷	5.023 × 10 ⁹
MAX 7000	2.98 × 10 ⁻¹⁷	5.023 × 10 ⁹

Figura 46. Cálculo del MTBF

En la fórmula, C₁ y C₂ son dos parámetros que se obtienen a partir de medidas del tiempo de recuperación realizadas sobre muestras de los circuitos fabricados por ALTERA. Haciendo uso de ellos se puede estimar el MTBF para unas condiciones de funcionamiento dadas o las condiciones de funcionamiento que deben garantizarse para obtener un determinado valor del MTBF.

Ejemplo

Vamos a calcular el tiempo de recuperación que debe tener el flip-flop de sincronización de un circuito síncrono que se desea realizar con una FPGA de ALTERA de la familia 10K, y que debe funcionar con un reloj de 20 MHz, si la frecuencia media de la entrada asíncrona es de 10 MHz y se desea conseguir un MTBF de 500 años (10¹⁰ segundos).

Despejando de la ecuación de la figura 45, obtenemos:

$$t_{MET} = \frac{\ln(MTBF \times C_1 \times f_{CLK} \times f_{DATA})}{C_2} = \frac{\ln(10^{10} \times 1.01 \times 10^{-13} \times 2 \times 10^7 \times 10^7)}{1.268 \times 10^{10}} \approx 2 \text{ ns}$$

Normalmente no es necesario realizar este tipo de cálculos, porque si un MTBF de decenas de años no resulta aceptable, basta con añadir al circuito de sincronización un segundo *flip-flop* (figura 47). Este *flip-flop* extra actúa como barrera de seguridad ante un fallo: cuando el primer *flip-flop* de

sincronización permanece en estado metaestable durante un ciclo de reloj y lo propaga al segundo – supongamos, por ejemplo, que una vez cada año de funcionamiento del circuito–, la probabilidad de que el segundo *flip-flop* propague a su vez este suceso es extraordinariamente remota –la misma que la anterior, del orden de 1 entre millones (o decenas o cientos de millones) de casos–, consiguiéndose valores enormes de MTBF.

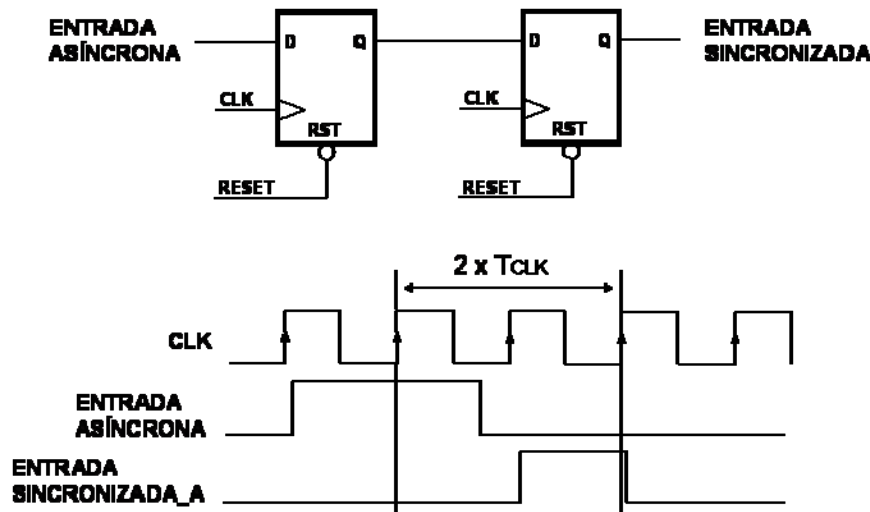


Figura 47. Sincronización con 2 *flip-flops*

El problema que presenta este circuito es que introduce un retardo adicional, de un ciclo de reloj, en la detección del cambio de estado de la entrada asíncrona. En las aplicaciones en que este retardo no resulta crítico, y en cambio la seguridad es esencial, puede llegar a incluirse un tercer *flip-flop*.

El problema se complica si es necesario detectar muy rápidamente los cambios de nivel de la entrada y, al mismo tiempo, se desea tener valores de MTBF altos. Una solución de compromiso para estas situaciones es utilizar en el circuito de sincronización un reloj con una frecuencia múltiplo de la del sistema síncrono –se puede generar fácilmente si se dispone de un *clock manager* (un PLL o un DLL)– y en fase con él; alternatively se pueden utilizar flancos alternos de reloj en el *flip-flop* de sincronización y el de barrera. En la figura 48 se ilustra el funcionamiento de estas soluciones. El circuito que funciona con un reloj cuya frecuencia es el doble que la del sistema síncrono, al estar en fase con éste sincroniza la señal en el flanco activo del reloj del sistema. El segundo circuito captura la señal asíncrona en el flanco de bajada y la sincroniza en el flanco de subida del reloj. Ambos circuitos consiguen un MTBF mejor que el circuito de sincronización con un solo *flip-flop*, y peor que el que utiliza un *flip-flop* adicional como barrera para la metaestabilidad –porque en estos circuitos los *flip-flops* tienen un tiempo de recuperación menor (aproximadamente la mitad).

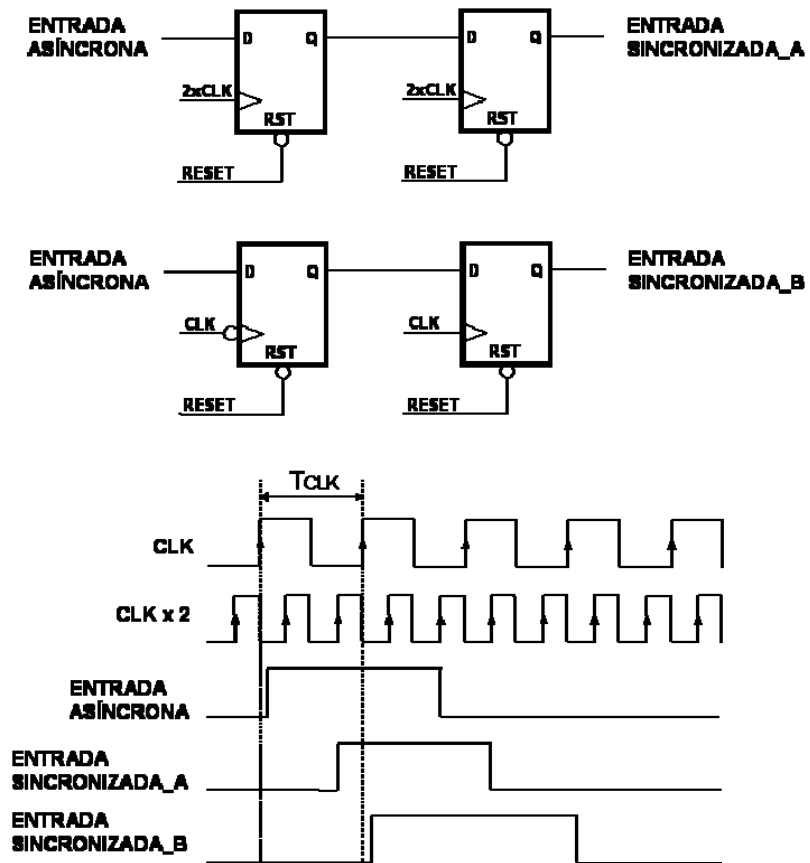


Figura 48. Circuitos de sincronización