

Tema 2.6 Sincronización

1

2.6 Sincronización

- 6.1 Estrategias de sincronización
- 6.2 Bases de la sincronización
 - Metaestabilidad y fallos de sincronización
 - Dominios de reloj
 - Clasificación
- 6.3 Diseño de sincronizadores
 - Mesosíncronos
 - Plesiosíncronos
 - Asíncronos periódicos
 - Asíncronos de propósito general

2

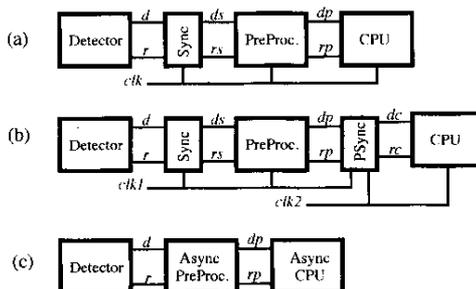
6.1 Estrategias de sincronización

- Sincronización es determinar o forzar un orden de eventos en las señales
 - Muestreo de señales asíncronas en un sistema síncrono
 - Transmisión de señales entre dos dominios de reloj
 - Arbitraje de señales asíncronas
- Siempre que la sincronización debe realizarse en un tiempo limitado existe alguna probabilidad de fallo
- Es posible sincronizar sin ningún fallo siempre que se espere el tiempo suficiente
- La dificultad de sincronización depende de lo previsible que sean los eventos relativos al reloj local
 - Sistemas **mesosíncronos**: los relojes tienen la misma frecuencia pero están desfasados
 - Sistemas **plesiosíncronos**: los relojes tienen casi la misma frecuencia
 - Sistemas **asíncronos periódicos**: los relojes tienen frecuencias diferentes
 - Sistemas **asíncronos**: no existe un reloj que fije la transición de eventos ³

6.1 Estrategias de sincronización

- Si se utiliza lógica asíncrona, la sincronización sólo es necesaria para arbitrar
- En general el diseño asíncrono es más rápido

Ejemplos de sistemas



- a.- Falta de modularidad, Cuidadosa distribución del reloj Dificultad de mejorar los distintos módulos por separado Puede funcionar sólo a ciertas frecuencias
- b.- Se solucionan parcialmente estos problemas, pero todavía hay una latencia de sincronización, cierta probabilidad de error disipación continua

c.- Es la opción que más ventajas proporciona: latencia de sincronización 0, menos latencia de segmentación, probabilidad de fallo 0, se disipa menos potencia

2.6 Sincronización

- 6.1 Estrategias de sincronización
- 6.2 Bases de la sincronización
 - Metaestabilidad y fallos de sincronización
 - Dominios de reloj
 - Clasificación
- 6.3 Diseño de sincronizadores
 - Mesosíncronos
 - Plesiosíncronos
 - Asíncronos periódicos
 - Asíncronos de propósito general

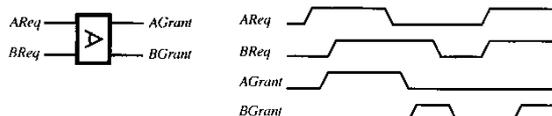
5

6.2 Bases de la sincronización

¿Cuándo es necesaria la sincronización?

- Arbitraje de señales asíncronas
- Muestreo de señales asíncronas en un sistema síncrono
- Transmisión de señales entre dos dominios de reloj

Arbitraje de señales asíncronas



En el segundo ciclo de arbitraje A y B piden casi al mismo tiempo. Debe utilizarse un árbitro para determinar de forma no ambigua el orden de las peticiones y garantizar que el recurso sólo se da a uno de los dos.

6

6.2 Bases de la sincronización

■ Muestreo de señales asíncronas

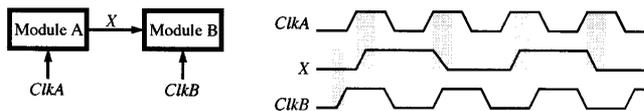


- El sincronizador muestrea X en el flanco de bajada del reloj, y actualiza XS en el flanco de subida
- El primer flanco de X se detecta de forma no ambigua como 1
- El segundo flanco de X puede detectarse como 1, como 0, o hacer que el sincronizador entre en un estado metaestable. En este caso se necesita esperar un cierto tiempo hasta que el estado de metaestabilidad decaiga

7

6.2 Bases de la sincronización

■ Transmisión de señales entre distintos dominios de reloj



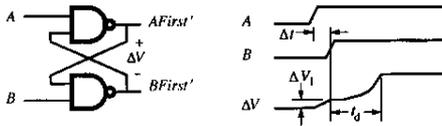
- X puede cambiar durante el tiempo de transición del reloj B
- Si los dos relojes son periódicos puede predecirse las transiciones de X y disminuir el coste de la sincronización

8

6.2 Bases de la sincronización

■ Metaestabilidad y fallos en la sincronización

- Cuanto menor sea la diferencia entre dos eventos, más tiempo se necesita para decidir cuál llegó primero
- Si no se deja el tiempo suficiente puede tener lugar un fallo de sincronización



El latch NAND actúa como árbitro

Si $\Delta V_1 \cong 0$ se entra en un estado metaestable. El ruido suele provocar la salida de este estado

$$\Delta V_1 = ks\Delta t$$

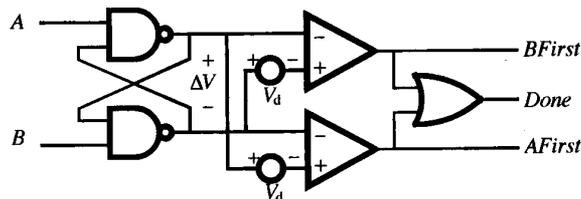
$$ks = \frac{I}{C} \approx \frac{1}{ta}$$

$$\Delta V = \Delta V_1 e^{\frac{t}{\tau s}}$$

$$td = \tau s \log\left(\frac{\Delta V}{\Delta V_1}\right)$$

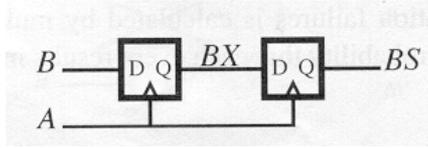
6.2 Bases de la sincronización

Este circuito previene que cualquier salida se actualice antes de que se tome una decisión sobre quién ha llegado antes



6.2 Bases de la sincronización

■ Sincronizador general



A tiene una frecuencia $f_A = 1/tc_A$
 B tiene una frecuencia $f_B = 1/tc_B$
 $f_A > f_B$

■ Se basa en la siguiente idea: si un sistema se diseña para que el tiempo máximo de espera sea t_w , la probabilidad de fallo de sincronización se decrementa exponencialmente con t_w .

■ En este caso t_w es un ciclo completo

■ El valor $\Delta V_1 / \Delta V$ está uniformemente distribuido entre 0 y 1

$$td > t_w$$

$$\tau \log\left(\frac{\Delta V}{\Delta V_1}\right) > t_w$$

$$td = \tau \log\left(\frac{\Delta V}{\Delta V_1}\right)$$

$$\Rightarrow \frac{\Delta V_1}{\Delta V} < e^{-\frac{t_w}{\tau}}$$

$$P(td > t_w) = e^{-\frac{t_w}{\tau}}$$

6.2 Bases de la sincronización

Si las transiciones de B pueden ocurrir en cualquier momento, la probabilidad de que ocurra en el tiempo de apertura de A será

$$Pa = \frac{ta}{tc_A} = ta * f_A$$

Si la transición en B cae dentro de t_a , dará lugar a un valor intermedio ΔV_1 , y el tiempo de decisión será:

$$td = \tau \log\left(\frac{\Delta V}{\Delta V_1}\right)$$

■ La probabilidad de que un flanco de B de lugar a un retardo del sincronizador mayor que t_w está dado por

$$Pfs = Pa * P(td > t_w) \Rightarrow Pfs(t_w) = ta * f_A * e^{-\frac{t_w}{\tau}}$$

La frecuencia de fallos del sincronizador vendrá dada por

$$ffs = f_B * ta * f_A * e^{-\frac{t_w}{\tau}} \quad 12$$

6.2 Bases de la sincronización

Símbolo	Valor	Unidades
f_A	100	MHz
f_B	1	MHz
t_a	200	ps
τ_s	200	ps
t_w	10	ns
Pfs	$3.86 \cdot 10^{-24}$	
Ffs	$3.86 \cdot 10^{-18}$	Hz
TeF	$2.59 \cdot 10^{17}$	s

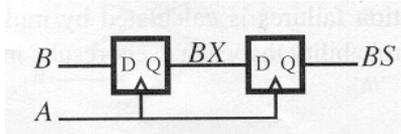
Tiempo entre fallos \swarrow \nwarrow 8.212.836.124 años

13

6.2 Bases de la sincronización

Problemas de este sincronizador:

se necesita esperar como mínimo un ciclo completo para sincronizar



El segundo FF espera un ciclo ($t_w=10\text{ns}$) para que el estado metaestable decaiga

El coste de la sincronización son los 10ns adicionales de latencia introducidos por el 2º FF

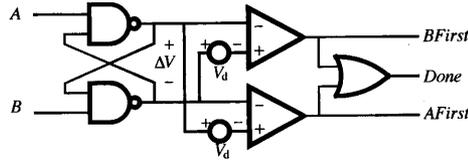
Tiempo medio de sincronización

$$t_z \approx t_{\text{setup}} + t_{dCQ} + t_{cy}/2 + t_{cy}$$

14

6.2 Bases de la sincronización

En un sistema asíncrono no es necesario esperar un tiempo igual al caso peor, sino que en el momento que se toma una decisión el siguiente circuito puede empezar a trabajar



V_d debe ser suficientemente grande para que el ruido no devuelva al sistema al estado metaestable

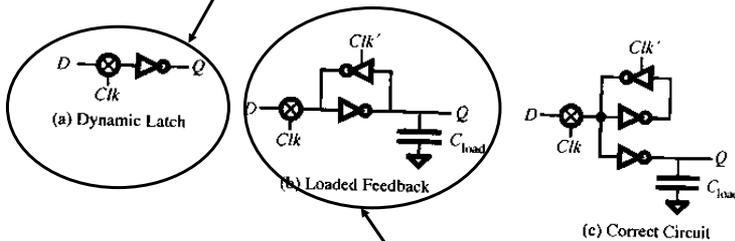
La mayor parte de las veces el circuito responderá inmediatamente, con lo cual es mucho más rápido que el sistema síncrono

15

6.2 Bases de la sincronización

- Un error muy común en sincronización es implementar el sincronizador mediante un latch dinámico (sin regeneración).

En este caso $\tau_s \approx \infty$ y no proporciona sincronización.



Otro error común es utilizar un latch estático pero de regeneración lenta, debido a que el lazo viene de la capacidad de carga

16

2.6 Sincronización

- 6.1 Estrategias de sincronización
- 6.2 Bases de la sincronización
 - Metaestabilidad y fallos de sincronización
 - Dominios de reloj
 - Clasificación
- 6.3 Diseño de sincronizadores
 - Mesosíncronos
 - Plesiosíncronos
 - Asíncronos periódicos
 - Asíncronos de propósito general

17

6.2 Bases de la sincronización

- Un ***dominio de reloj*** es un conjunto de señales sincronizadas con el mismo reloj.
- Todas las señales son estables durante el t_a de cada evento de reloj.
- Las combinaciones lógicas de dos señales de un mismo dominio que cumplan las restricciones de retardo máximo y mínimo también están en dicho dominio.
- La mayor parte de los sistemas se dividen en varios dominios de reloj. Así el comportamiento secuencial del sistema es más complejo.

18

6.2 Bases de la sincronización

- Ventajas de la división en varios dominios de reloj:
 - 1- **Frecuencias independientes** de los distintos módulos. Esto facilita la actualización individualizada
 - 2- Se simplifica la **distribución de reloj** (sólo hay que ajustar retardos dentro de cada dominio).
 - 3- La **temporización segmentada** elimina las restricciones del retardo de los cables
 - 4- Se pueden utilizar **relojes aperiódicos**, disminuyendo el consumo y aumentando la eficiencia.

19

2.6 Sincronización

- 6.1 Estrategias de sincronización
- 6.2 Bases de la sincronización
 - Metaestabilidad y fallos de sincronización
 - Dominios de reloj
 - Clasificación
- 6.3 Diseño de sincronizadores
 - Mesosíncronos
 - Plesiosíncronos
 - Asíncronos periódicos
 - Asíncronos de propósito general

20

6.2 Bases de la sincronización

Clasificación	Periódico	$\Delta\Phi$	Δf	
Síncronos	Sí	0	0	Se puede muestrear directamente
Mesosíncronos	Sí	Φ_c	0	Se puede muestrear con el reloj retardado
Plesiosíncronos	Sí	Varía	$fd < \varepsilon$	Reloj retardado por una cantidad variable
Periódicas	Sí		$fd > \varepsilon$	La naturaleza periódica puede explotarse para predecir
Asíncronos	No			Se necesita un sincronizador completo

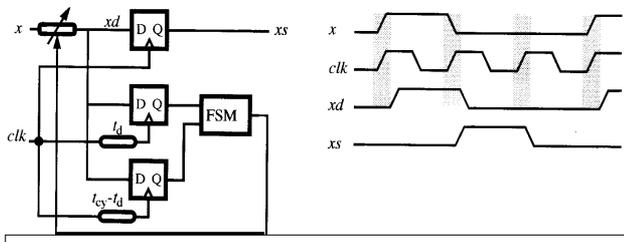
Cuanto mayor es el nivel de sincronía, más fácil es sincronizar²¹

2.6 Sincronización

- 6.1 Estrategias de sincronización
- 6.2 Bases de la sincronización
 - Metaestabilidad y fallos de sincronización
 - Dominios de reloj
 - Clasificación
- 6.3 Diseño de sincronizadores
 - Mesosíncronos
 - Plesiosíncronos
 - Asíncronos periódicos
 - Asíncronos de propósito general

6.3 Diseño de sincronizadores

- Una señal mesosíncrona está sincronizada con el reloj local, pero desfasada



$t_a < t_d$

Se necesita una secuencia de inicialización

La FSM ajusta el retardo de la línea de retardo.
Debe considerar la posibilidad de que los FF entren en estado metaestable (esperar a que los estados metaestables decaigan)

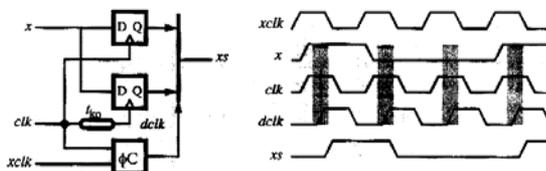
El sincronizador tiene un retardo medio de $t_z = t_{vmin} + t_{setup} + t_{dCQ} + t_{cy}/2$

Retardo variable mínimo

Tiempo mínimo de sincronización

6.3 Diseño de sincronizadores

- Este sincronizador resulta muy caro para sincronizar una señal multibit. Una solución más barata se muestra en la siguiente figura:



La entrada será muestreada OK por uno de los dos FF si $t_a < t_{cy}/2$

tko: retardo igual al ancho de apertura

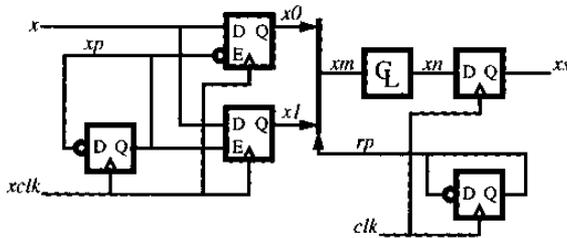
El control del mux es una cte que se obtiene durante la inicialización del sistema (se compara la fase relativa de clk y xclk)

$t_z = t_{ko}^2 / t_{cy} + t_{setup} + t_{dCQ} + t_{cy}/2$ Retardo de FF de abajo por la probabilidad de que se seleccione

- Si se selecciona el FF inferior, xs está retardada en tko respecto al resto de FF.
- Se limita la lógica que puede colocarse detrás del sincronizador.

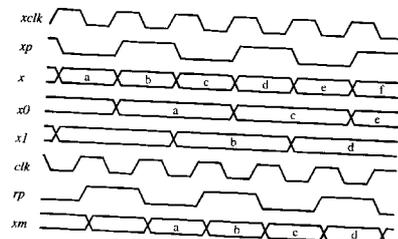
6.3 Diseño de sincronizadores

Otra solución: el Sincronizador FIFO



Formas de onda para un desfase de 180°

La fase de rp está preparada para muestrear siempre el FF que está estable durante todo el ciclo
 xm está sincronizada con clk y puede utilizarse con lógica combinacional y muestrearse con el reloj local



6.3 Diseño de sincronizadores

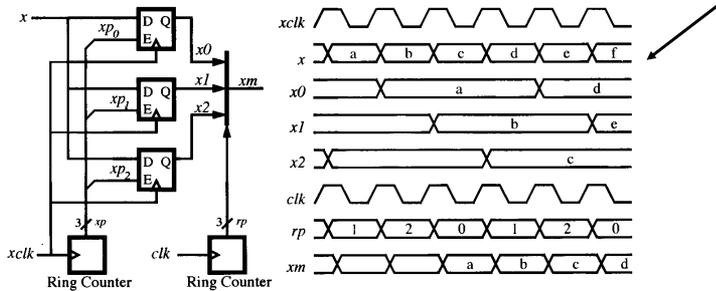
$$tz = t_{mux} + t_{dCQ} + t_{cy}/2$$

- No hay penalty debido a t_a de los FF porque el reloj del transmisor se usa para excitar los FF
- Si el reloj local y el del transmisor están casi en fase, este sincronizador no puede asegurar que xm esté estable durante todo el ciclo
- Se suele ajustar rp para que la inestabilidad sólo ocurra al principio del ciclo
- Otra solución: añadir más elementos de almacenamiento al FIFO

6.3 Diseño de sincronizadores

FIFO de 3 elementos

Formas de onda cuando los dos relojes están en fase



Los 3 sincronizadores mesosíncronos vistos eliminan la probabilidad de fallo explotando la naturaleza periódica de los eventos de las entradas.

Tienen menos retardo que el caso general

Necesitan una inicialización

27

2.6 Sincronización

- 6.1 Estrategias de sincronización
- 6.2 Bases de la sincronización
 - Metaestabilidad y fallos de sincronización
 - Dominios de reloj
 - Clasificación
- 6.3 Diseño de sincronizadores
 - Mesosíncronos
 - Plesiosíncronos
 - Asíncronos periódicos
 - Asíncronos de propósito general

28

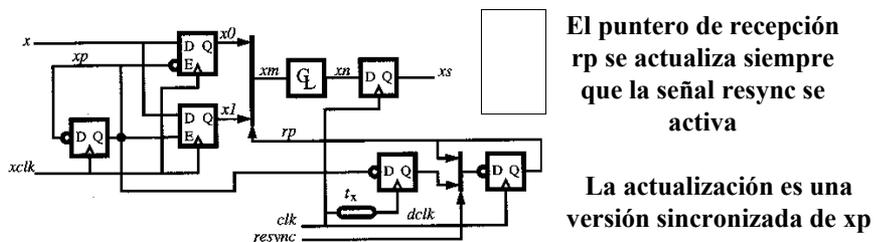
6.3 Diseño de sincronizadores

- En este caso **la fase cambia lentamente** con respecto al reloj local.
- Se pueden utilizar los sincronizadores mesosíncronos con un **reajuste periódico**
- Si el reloj del transmisor es más rápido que el del receptor puede haber pérdida de datos. Solución: cuando se realice un reajuste el transmisor debe incluir un **NULL**
- Si el reloj del receptor es más rápido que el del transmisor puede haber duplicación de datos. Solución: al realizar el reajuste **eliminar un dato**

29

6.3 Diseño de sincronizadores

- FIFO modificado

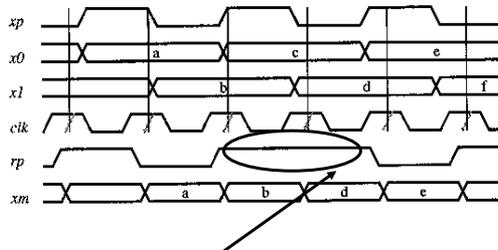


Si $resync$ es 1, el circuito actualiza rp todos los ciclos

30

6.3 Diseño de sincronizadores

- Temporización para el caso de clk más lento que xclk



Las líneas verticales señalan los pto donde dclk muestrea xp

Debido al desajuste, el punto de muestreo se desplaza y rp tiene el mismo valor durante 2 ciclos

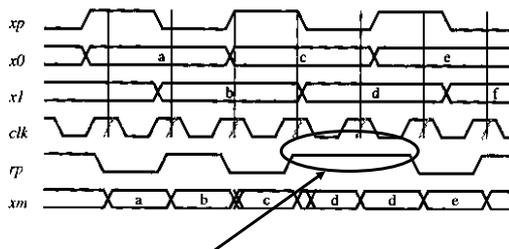


c se pierde

31

6.3 Diseño de sincronizadores

- Temporización para el caso de clk más rápido que xclk



Los puntos de muestreo se van retrasando a través de rp

Debido al desajuste, el punto de muestreo se desplaza y rp tiene el mismo valor durante 2 ciclos

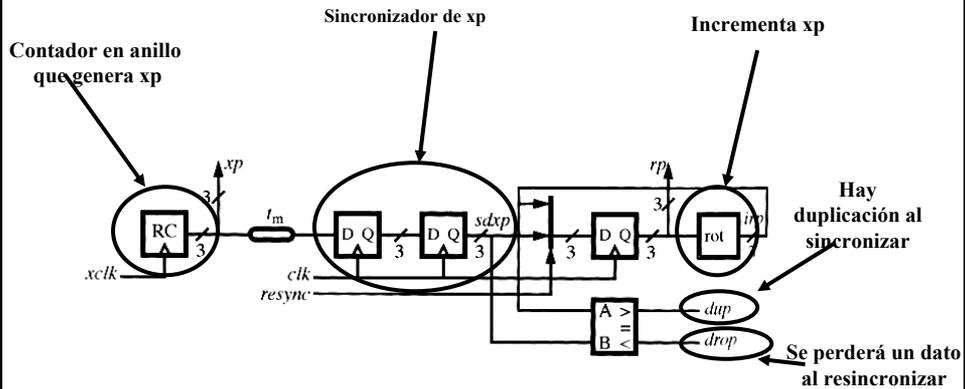


d se duplica

32

6.3 Diseño de sincronizadores

El circuito siguiente detecta cuando y en qué dirección debe ajustarse rp para un buffer de 3 elementos



33

6.3 Diseño de sincronizadores

- Para solucionar el problema de la duplicación y pérdida de datos se usan los **símbolos nulos**.
- Si la fase de rp se retrasa, repitiendo el mismo estado estado durante 2 ciclos en una fila, la duplicación puede evitarse insertando un **NULL** durante el ciclo que sigue al ajuste.
- Para evitar la pérdida de datos, la línea $resync$ se mantiene a 0, postponiendo el ajuste a la aparición de un **NULL**.
- Lo ideal es una solución en lazo cerrado para asegurar que aparecerá un símbolo **NULL** dentro del tiempo requerido.

34

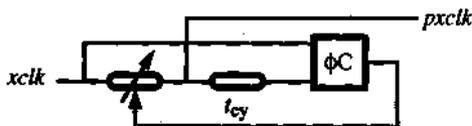
2.6 Sincronización

- 6.1 Estrategias de sincronización
- 6.2 Bases de la sincronización
 - Metaestabilidad y fallos de sincronización
 - Dominios de reloj
 - Clasificación
- 6.3 Diseño de sincronizadores
 - Mesosíncronos
 - Plesiosíncronos
 - Asíncronos periódicos
 - Asíncronos de propósito general

35

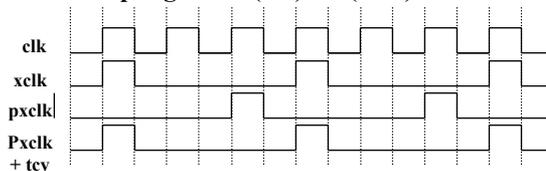
6.3 Diseño de sincronizadores

- Si las frecuencias de los relojes del transmisor y del receptor son muy diferentes, se puede diseñar un sincronizador que **explote la naturaleza periódica** de las señales para predecir las transiciones arbitrariamente lejanas y evitar fallos en la sincronización



Circuito para predecir el valor de $xclk$ un ciclo de reloj (local) t_{cy} después

Supongamos $f(\text{clk})=3 f(xclk)$

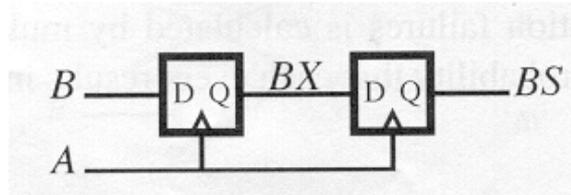


Puede utilizarse la misma técnica para predecir $xclk$ varios ciclos en el futuro

36

6.3 Diseño de sincronizadores

- En este caso no hay predicción posible.
- La sincronización debe realizarse en respuesta a un evento introduciendo una espera

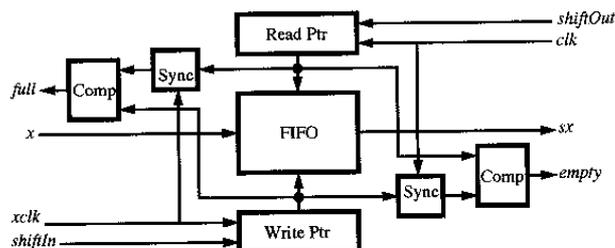


Este sincronizador introduce un retardo de uno o más ciclos y tiene una cierta probabilidad de fallo

39

6.3 Diseño de sincronizadores

- Sincronizador FIFO
 - Los datos se meten usando $xclk$
 - Los datos se sacan usando clk



La sincronización se realiza en los punteros de transmisión y recepción para generar las señales de vacío en el dominio del reloj local y lleno en el del transmisor

40

