

TEMA IV: ANÁLISIS DE SISTEMAS SECUENCIALES

En los temas anteriores hemos tratado los fundamentos teóricos de los diagramas de estados, así como los elementos básicos secuenciales. Así que ya estamos en disposición de poder abordar el problema de determinar el comportamiento de un circuito secuencial.

En este tema vamos a tratar el análisis de los sistemas secuenciales, pues ya conocemos todos los componentes de un circuito secuencial sencillo, puertas lógicas y biestables. Para lo cual veremos, en primer lugar, las propiedades que debe cumplir cualquier sistema secuencial, y las principales diferencias existentes entre las operaciones de los tipos básicos de sistemas secuenciales: sistemas síncronos y asíncronos. De este modo sabremos cómo afrontar uno u otro problema.

1. Introducción. Definición.

En primer lugar vamos a dar una definición más exhaustiva de circuito secuencial, ya que cualquier sistema con dependencia de su historia no se puede considerar como tal, ya que para que el sistema sea útil, se debe conocer como va a evolucionar y no presentar ningún comportamiento aleatorio. Por lo tanto, cualquier comportamiento secuencial que satisfaga las siguientes propiedades podrá implementarse con un circuito secuencial.

- **Ninguna dependencia con el futuro.** La salida solamente debe depender del valor de las entradas en el instante actual y/o de su historia pasada, nunca en su historia futura. Esta propiedad es consecuencia directa del principio causa-efecto, el cual nos indica que el efecto (la salida) nunca debe preceder a la causa (las entradas).
- **Requerimientos de memoria finito.** La cantidad de información que es necesaria almacenar, es decir, el número de situaciones por las que pueda pasar dicho sistema, debe ser finita. Esta propiedad implica que el sistema sea realizable, es decir, debemos necesitar un número finito de elementos entre los que se encuentran los elementos de memoria. Por lo tanto, si el número de elementos, y entre ellos los de memoria, debe ser finito, implica que la cantidad de información que se pueda almacenar debe ser finita.
- **Ninguna dependencia analógica.** La secuencia de señales de entrada debe determinar completamente la secuencia de salida, y esta última no debe depender en ningún momento de ninguna variable analógica. Su verificación es consecuencia directa del comportamiento digital de estos sistemas. Así, su comportamiento en estado estacionario (debido a la no idealidad de los elementos) no debe depender de variables analógicas, como puede ser el retraso *exacto* de una puerta.

- **Operación determinista.** La salida no debe depender de ningún tipo de variables internas, cuyo comportamiento sea aleatorio y no bien determinado. La operación debe estar bien determinada, una vez que se haya elegido la implementación. En caso contrario, no estaría controlado el comportamiento de nuestro sistema secuencial, porque no se podría garantizar un comportamiento correcto. Esta situación no debe ser confundida con las inespecificaciones, por las cuales no importa el comportamiento del sistema en dichas situaciones, eligiéndose aquel que obtenga la implementación con un menor coste.

Por lo tanto, el comportamiento que sigue dichas propiedades puede ser modelado por un sistema secuencial formado por un bloque combinacional y una serie de elementos de memoria, ya sean elementos explícitos o implícitos, a través de realimentaciones directas o elementos de retraso.

2. Problemas de sincronización.

La definición anterior es verificada por todos los circuitos secuenciales, ya sean síncronos o asíncronos. La diferencia entre ambos tipos consiste en las soluciones que se proponen a los problemas de sincronización, externas o internas. Para nosotros, estos problemas serán aquellos derivados de la secuenciación de las entradas y/o estados que pueden provocar un comportamiento incorrecto.

Entre los problemas más importantes podemos encontrar los cambios simultáneos de señales de entrada, los cambios consecutivos de señales de entrada, las carreras y los ciclos de estado. A continuación vamos a tratarlos en mayor detalle.

2.1. Cambios simultáneos de señales de entrada.

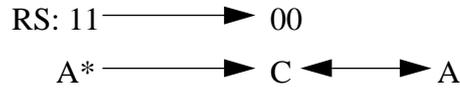
Este problema es consecuencia directa de los azares de función, que veíamos en el análisis de sistemas combinacionales. Por ejemplo, si tratamos la tabla de estados del biestable RS, mostrada en la tabla 4.1, y estudiamos las transición desde RS=11 a RS=00.

		RS			
		00	01	11	10
A	C	D	A*	B	
B	B*	A	A	B*	
C	A	A	A	A	
D	D*	D*	A	A	

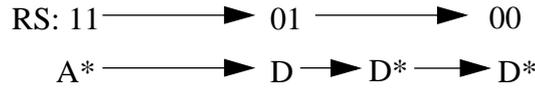
Tabla 4.1. Tabla de estados correspondiente al biestable RS.

Cuando nos encontramos en la combinación de entradas RS=11, partiremos del estado estable A, por ser el único estado estable para dicha combinación. Nos podemos encontrar ante tres situaciones posibles, ante la complejidad de realizar un cambio realmente simultáneo:

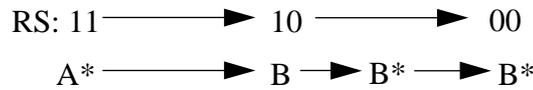
- Si el cambio es realmente simultáneo, pasaremos directamente a la columna RS=00 en el estado A. De este estado, pasaremos al estado C y desde ahí, de nuevo, al estado A. Por lo tanto entraremos en un ciclo sin fin, del que saldremos al cambiar nuevamente la combinación de señales de entrada.



- Si el cambio no es realmente simultáneo, sino que pasamos por una combinación intermedia, puede ser RS=01 ó RS=10. Si pasamos por RS=01, en esta combinación se llegará al estado estable D, el cual se mantendrá en la combinación RS=00, por ser también estable.



- Si, por contra, pasamos por RS=10, en esta combinación se llegará al estado estable B, el cual se mantendrá para la combinación RS=00, por ser también estable.



Resumiendo, obtendremos una dependencia con el valor relativo de los retrasos de las puertas, teniendo como consecuencia una operación no determinista.

2.2. Cambios consecutivos de entrada.

Los cambios consecutivos solamente son problemáticos cuando son lo suficientemente rápidos como para no dar el suficiente tiempo para garantizar la estabilización del sistema. De esta forma podemos encontrar un comportamiento diferente al esperado.

Consideremos de nuevo el biestable RS, y las transiciones RS: 00 -> 01 -> 00, para estudiar la influencia del intervalo temporal existente en la combinación RS=01 sobre el comportamiento de dicho biestable.

- Si partimos del estado estable B y consideramos un intervalo temporal grande, el sistema tiene el tiempo suficiente para estabilizarse. Por lo tanto, se pasará del estado B al A. Durante dicho intervalo se alcanzará el estado estable D, en el que se seguirá con la combinación RS=00. Este comportamiento se puede ver en la tabla 4.2.

		RS			
		00	01	11	10
A	C	D	A*	B	
B	B* →	A	A	B*	
C	A	A	A	A	
D	D* ←	D*	A	A	

Tabla 4.2. Tabla de estados correspondiente al biestable RS en la combinación RS:00 -> 01 -> 00, con un intervalo temporal grande en las transiciones.

- Si partimos del estado estable B y consideramos un intervalo temporal pequeño, el sistema no tiene el tiempo suficiente para estabilizarse pero si para cambiar de estado. Por lo tanto, se pasará del estado B al A. Desde este estado A se producirá la nueva

transición de entradas, por lo que llegaremos al estado C de la columna RS=00. Una vez en este estado y esta columna, entraremos en un comportamiento cíclico entre los estados A y C, del cual solamente se saldrá con una nueva transición en las señales de entrada. Este comportamiento se puede ver en la tabla 4.3.

		RS			
		00	01	11	10
A	C	D	A*	B	
B	B*	A	A	B*	
C	A	A	A	A	
D	D*	D*	A	A	

Tabla 4.3. Tabla de estados correspondiente al biestable RS en la combinación RS:00 -> 01 -> 00, con un intervalo temporal grande en las transiciones.

- Si partimos del estado estable B y consideramos un intervalo temporal pequeño, el sistema no tiene el tiempo suficiente para estabilizarse ni tampoco para cambiar de estado. Por lo tanto, el sistema se comportará como si no se hubiese producido ninguna transición en las señales de entrada, es decir, se mantendrá en el estado de partida estable, B. Este comportamiento se puede ver en la tabla 4.4.

		RS			
		00	01	11	10
A	C	D	A*	B	
B	B*	A	A	B*	
C	A	A	A	A	
D	D*	D*	A	A	

Tabla 4.4. Tabla de estados correspondiente al biestable RS en la combinación RS:00 -> 01 -> 00, con un intervalo temporal grande en las transiciones.

Por lo general, estos cambios son pequeños y su origen es muy diverso. Uno de sus posibles orígenes pueden ser los azares debido a la implementación de la lógica combinacional.

2.3. Carreras.

Este problema consiste en el cambio simultáneo de más de una variable de estado, las cuales no las podemos controlar directamente.

Si volvemos a considerar el biestable RS con su codificación de estados vista anteriormente, según la tabla 4.5, y la combinación de entradas RS=01, podemos comprobar que existe una carrera para el estado presente 11 (ya que el próximo estado es el 00 y cambia más de una señal de estado). Como sucede con las entradas, es muy difícil conseguir cambios simultáneos de las variables de estado. Por lo tanto, en dicho cambio de estado se puede pasar de forma temporal por los estados 10 ó 01:

		RS			
		00	01	11	10
$\bar{q}q$	00	11	10	00	01
	01	01*	00	00	01*
	11	00	00	00	00
	10	10*	10*	00	00

Tabla 4.5. Tabla de transición correspondiente al biestable RS.

- Si el cambio de estado es simultáneo y no pasamos por ningún otro estado, el comportamiento es el siguiente. Partiendo del estado 11, se pasará al estado 00. Una vez en este estado, se pasará al estado 10 en el que permanecerá por ser estable.
- Si el cambio de estado pasa por el estado 10, el biestable se quedará en este estado por ser estable y no completará la transición esperada.
- Si el cambio de estado pasa por el estado 01, el biestable pasará al estado 00, de ahí al 10 en el que permanecerá.

Este comportamiento se ve esquematizado en la figura 4.1.

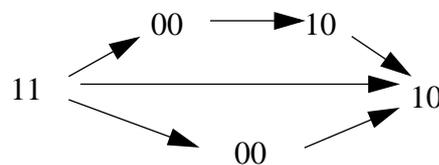


Figura 4.1.- Cambios de estados seguido si partimos del estado presente 11 en la combinación RS=01.

En cambio, si consideramos la combinación RS=00, podemos comprobar que existe una carrera para el estado presente 11. Si estudiamos los cambios de estado, encontraremos el siguiente comportamiento:

- Si el cambio de estado es simultáneo y no pasamos por ningún otro estado, el comportamiento es el siguiente. Partiendo del estado 11, se pasará al estado 00. Una vez en este estado, se pasará al estado 11 entrando en un comportamiento cíclico.
- Si el cambio de estado pasa por el estado 10, el biestable se quedará en este estado por ser estable y no completará la transición esperada.
- Si el cambio de estado pasa por el estado 01, el biestable se quedará en este estado por ser estable y no completará la transición esperada.

Este comportamiento se ve esquematizado en la figura 4.1.

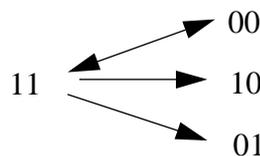


Figura 4.2.- Cambios de estados seguido si partimos del estado presente 11 en la combinación RS=00.

Como podemos ver en los casos anteriores, podemos encontrar dos comportamientos bien diferenciados. En el caso de que el estado final sea el mismo, independientemente de los estados por los que pase de forma temporal, se dice que estamos ante una **carrera no crítica**, que correspondería al primer caso estudiado. Por el contrario, si el estado final depende de los estados por lo que pase de forma temporal, se dice que estamos ante una **carrera crítica**. Obviamente, las carreras no críticas son permisibles ya que la diferencia entre un camino u otro consiste en una diferencia de retraso en alcanzar el estado final en el peor caso. En cambio, las carreras críticas deben ser eliminadas ya que producirán un comportamiento no determinista.

2.4. Transiciones cíclicas de estados.

Este problema se da cuando existe una combinación de entradas y un estado inicial a partir del cual no es posible alcanzar ningún estado estable. El problema consiste en que no sabemos en qué estado se encuentra en cada intervalo temporal. Por lo tanto, no sabemos cuál será el estado desde donde se producirá la siguiente transición de señales de entradas, dando lugar a un comportamiento no determinista.

Si consideramos el biestable RS en su combinación RS=00, encontramos un ciclo entre los estados A y C (como se puede ver en la tabla 4.1). El tiempo de permanencia en cada estado depende del sistema, más concretamente del retraso de una puerta NOR en este caso particular. Este retraso depende de numerosos factores como pueden ser la combinación de las señales de entrada, la tecnología de fabricación, la vida de los circuitos, etc. Para este caso particular, esta situación no es problemática ya que siempre se llega al mismo estado estable de forma independiente del estado en el que se dé la transición, como vimos en el apartado anterior. La única diferencia sería un mayor o menor retraso en alcanzar el estado estable. Supongamos la transición RS=00 \rightarrow 01, las secuencias de estado posibles serían:

- C \rightarrow A \rightarrow D \rightarrow D*, para el caso de que partiéramos del estado C cuando se produce la transición.
- A \rightarrow D \rightarrow D*, para el caso de que partiéramos del estado A cuando se produce la transición.

Luego, la única diferencia sería un cambio más de estado que se traduce en un retraso mayor.

Este problema también puede ser visto como una de las condiciones de **metaestabilidad**. Uno de los posibles comportamientos de este fenómeno puede ser el comportamiento cíclico descrito anteriormente. No obstante, si los cambios de estado son lo suficientemente rápidos, las puertas no tendrán el tiempo suficiente para acabar su evaluación. Así, llegaremos a un estado en el que las variables no tienen un valor lógico bien determinado, como se puede ver en la figura 4.3.



Figura 4.3.- Comportamiento de metaestabilidad.

3. Soluciones.

Las diferentes soluciones de los problemas de sincronización (carreras, ciclos, cambios consecutivos y cambios simultáneos de entrada) van a ser diferenciados según el tipo de almacenamiento utilizado: implícito o explícito. El almacenamiento implícito es aquel en el que se utilizan realimentaciones directas, elementos de retraso o biestables transparentes, tal que cualquier cambio en las salida de la lógica combinacional producirá un efecto sobre los biestables transparentes. Por contra, el almacenamiento explícito es aquel que utiliza biestables no transparentes, sino que tienen señales de control, ya sean externas o internas (latches o flip-flops). Esta distinción se muestra en la figura 4.4.

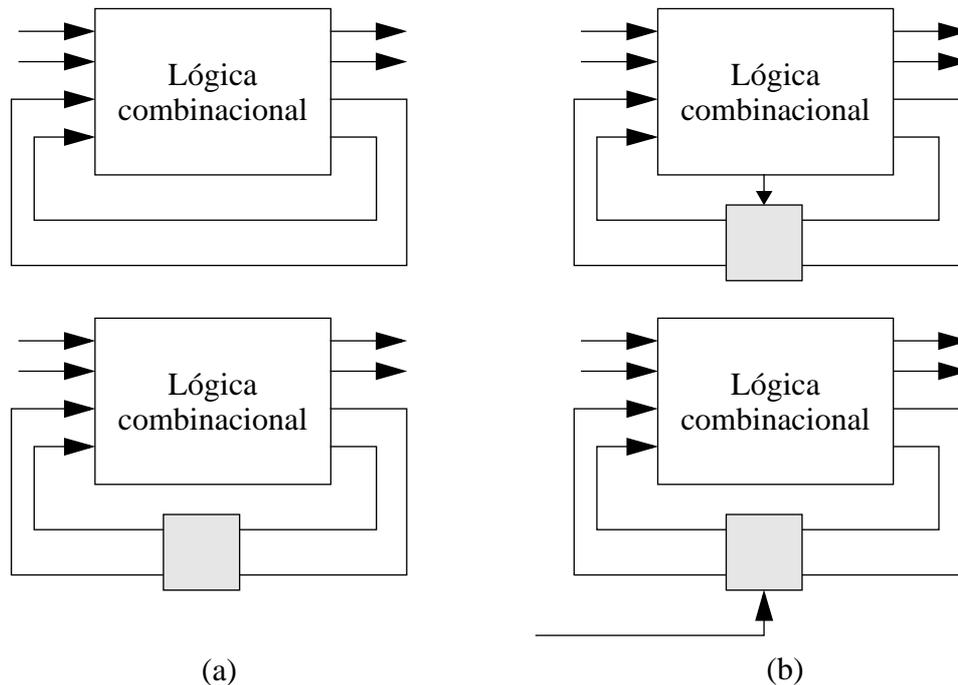


Figura 4.4.- Esquemas secuenciales con almacenamiento (a) implícito y (b) explícito.

3.1. Soluciones mediante Sistemas con Almacenamiento Implícito.

En este caso no existe ningún tipo de control desde el exterior sobre los cambios de estados. Estos cambios siempre tendrán lugar una vez haya transcurrido el retraso de la lógica del sistema después de producirse una transición en las señales de entrada y/o de estado. Este comportamiento es el que se ha seguido en el tratamiento de los problemas anteriores.

El primer problema que consideramos era el cambio simultáneo de las señales de entrada. Esta situación es problemática, por lo que debemos prohibir la existencia de dichos cambios, de la misma forma que evitábamos los azares de función en un circuito combinacional. Como, las señales de entrada sí las podemos controlar (ya que se introducen desde el exterior), **no se deben permitir cambios múltiples en las señales de entrada.**

El segundo problema considerado consistía en los cambios consecutivos de señales de entrada, que tendrá la misma problemática que los cambios consecutivos de las señales de

estado. Para que el comportamiento sea correcto, estos cambios deben estar lo suficientemente espaciados temporalmente como para permitir que el sistema alcance el estado estable. Desde el exterior se puede imponer esta restricción sobre las señales de entrada, pero no sobre las señales de estado. Si nos centramos en las señales de estados, el origen de los cambios consecutivos de estas señales serán los azares lógicos (ya que los de función son eliminados por la condición anterior) de la lógica combinacional. También sabemos que un circuito libre de azares y sin cambios múltiples solamente tendrán una transición en sus salidas en el caso de que ésta deba cambiar, evitando cambios consecutivos en las señales de estado. Por lo tanto, otra restricción que debemos imponer es **el diseño de la lógica combinacional libre de azares**. Este efecto, el cambio único de las señales de salida y/o de próximo estado, será el que determine el protocolo de comunicaciones, de tal forma que **las señales de entrada no deben cambiar hasta que no hayan cambiado las señales de salida o no haya transcurrido el tiempo necesario como para garantizar que la operación haya finalizado**.

Los dos siguientes problemas, los cuales están relacionados directamente con las señales de estado, no se les puede dar una solución durante la operación ya que estos cambios no están controlados desde el exterior. Por lo tanto, la eliminación de estos últimos problemas se abordarán en el proceso de diseño. Por ejemplo, un asignamiento adecuado producirá que en el circuito no existan carreras críticas.

Los circuitos cuya operación es regida por dichas restricciones, se dice que opera según el modo fundamental o modo de Huffman. Resumiendo, el modo de operación fundamental o de Huffman es aquel que sigue las siguientes restricciones, además de todas las de cualquier sistema secuencial:

- No se permiten los cambios simultáneos en más de una señal de entrada.
- No se permiten cambios de las señales de entrada hasta que no se haya producido transiciones en las señales de salida o haya transcurrido el tiempo suficiente como para garantizar que la operación haya finalizado.

Como podemos ver, este sistema no tiene una sincronización global y externa por lo que estamos ante un sistema asíncrono. Su correcta operación está restringida a un diseño de la lógica combinacional libre de azares.

3.2. Soluciones mediante Sistemas con Almacenamiento Explícito.

En este caso tenemos un relativo control sobre las variables de estado a través del control del almacenamiento. En este caso, los cambios de estado solamente se producirán cuando lo permita la señal de control de los biestables.

En estas soluciones vamos a hacer una diferencia según sus fases de operación. Estas fases son: fase de almacenamiento, en la que el biestable es transparente y las salidas siguen a las entradas; y fase de mantenimiento, en la que el biestable es opaco y las salidas se mantienen estables, como se pueden ver en la figura 4.5. En función de estas fases podemos diferenciar entre biestables transparentes, latches y flip-flops.

En la fase de almacenamiento, el biestable es completamente transparente, y por lo tanto, se producirán tantos cambios de estado como cambios de entrada en esta fase. Excepto en los flip-flops, la fase de almacenamiento es lo suficientemente grande como para producir varios

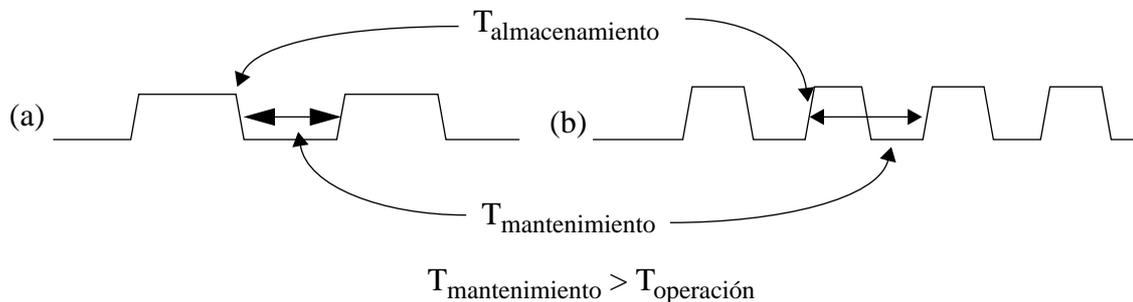


Figura 4.5.- Esquema de la fase de mantenimiento en función de la señal de control para (a) latches y (b) flip-flops.

cambios de estado; luego se deben tener en cuenta todas las precauciones observadas en los sistemas con almacenamiento implícito. No obstante en el caso de los flip-flops, la fase de almacenamiento se reduce a un instante (en el que se produce la transición del reloj), por lo que sólo se producirá un solo cambio de estado si es necesario, evitándose los problemas debido a los cambios “incontrolados” de estados. Esta razón ha motivado a que la mayoría de sistemas secuenciales se realicen utilizando flip-flops en lugar de biestables transparentes o latches.

Al no permitir los cambios de estados de forma incontrolada, los tres primeros problemas no son tales siempre y cuando se cumplan una serie de restricciones temporales. Estas restricciones se resumen básicamente en no permitir el cambio de estado (y por tanto el almacenamiento) hasta que el sistema no haya llegado a un estado estable. El último problema, el comportamiento cíclico, no sería tal puesto que el cambio de estado viene determinado por la señal de control y su comportamiento está bien determinado.

Una vez que hemos visto la solución a los problemas con este esquema secuencial, el cual consiste en controlar los cambios de estado a través de su almacenamiento, se presenta un nuevo problema: la generación de la señal de control de los latches o flip-flops. Según sea la naturaleza de esta señal, encontraremos una solución asíncrona y otra síncrona. Para que se produzca una operación correcta, el biestable tiene que esperar a que la lógica combinacional haya finalizado la operación para poder cambiar de estado, luego se tiene que verificar que el tiempo de mantenimiento debe ser mayor o igual que el tiempo de operación de la lógica combinacional. Además durante la fase de almacenamiento del biestables, la lógica combinacional debe permanecer estable para no almacenar un dato incorrecto. Luego la señal de control de un latch debe tener un periodo mayor (típicamente el doble) que la de un flip-flop, ya que al ser simétrica los tiempos de mantenimiento y almacenamiento serán iguales, luego el periodo debe ser mayor que el doble del tiempo de operación de la lógica. En cambio, en los flip-flops, no se produce esta simetría, reduciendo el tiempo de almacenamiento prácticamente a 0, luego el periodo de la señal de reloj (que coincidirá con la fase de mantenimiento) deberá ser mayor que el tiempo de operación de la lógica.

El siguiente paso será profundizar en las soluciones. La **solución** asíncrona se denomina **autotemporizada**, y consiste en generar localmente la señal de control dentro de la lógica combinacional. Esta señal se activa cuando ha finalizado la operación del sistema y se puede proceder al almacenamiento de los datos. La dificultad de esta solución asíncrona no consiste en la generación de la lógica combinacional, sino en la estrategia que se debe seguir para generar dicha señal de control. La ventaja de esta solución radica en que no existe ninguna restric-

ción temporal en las señales de entrada, propias de la operación del sistema, como sucedía en el modo fundamental y en la solución síncrona que veremos a continuación. El propio circuito se encarga de indicarnos el final de la operación, y por lo tanto del momento en el que podemos cambiar las señales de entrada.

Por lo tanto cualquier solución autotemporizada debe cumplir, además de las cuatro propiedades comunes a todos los sistemas secuenciales, la siguiente propiedad:

- Todos los cambios de estado interno se producen en respuesta a la zona activa (nivel o transición) de una variable, por lo general interna.

La **solución síncrona** consiste en la utilización de una o varias señales externas a nuestro sistema que controlarán el almacenamiento en los biestables. Estas señales reciben el nombre genérico de **relojes**. Un reloj se puede definir de la siguiente forma:

Un **reloj** es una señal cuadrada periódica en la cual la zona inactiva debe ser lo suficientemente espaciada temporalmente como para garantizar que la lógica combinacional haya finalizado la operación.

Por lo tanto, cualquier solución síncrona debe cumplir las siguientes propiedades, además de las necesarias para cualquier sistema secuencial:

- Todos los cambios de estado interno se producen en respuesta a la zona activa (nivel o transición) de una variable externa.
- Una transición en cualquier señal de estado nunca provocará un nuevo cambio de estado. Esto es debido a que usualmente se usan flip-flops, por lo que la fase de transparencia se limita a una transición de la señal de reloj.
- Hay una única entrada privilegiada, el reloj. El privilegio es debido a que debe llegar a todos los flip-flops de forma simultánea (en la mayor medida de lo posible) para sincronizar todos los cambios de estado presentes en el sistema. Cuando utilizamos más de una fase (señal) de reloj, todas estas fases (señales) son generadas internamente a partir de un único reloj externo.

3.3. Resumen.

A modo de resumen, mostramos en la tabla 4.6 las propiedades que deben cumplir todas las soluciones expuestas: modo de Huffman, solución autotemporizada y solución síncrona.

4. Análisis de Sistemas Secuenciales.

Si recordamos la definición de análisis:

El **problema de análisis** de un sistema consiste en obtener el comportamiento de dicho sistema a partir de su implementación a nivel de circuitos.

En el caso de los sistemas secuenciales, el análisis acabará con la tabla y/o diagrama de estados. Por lo tanto, los pasos que se deben seguir serán los siguientes:

- Obtención de las funciones combinacionales de las señales de próximo estado y de salida.

Soluciones asíncronas		Solución síncrona
Modo fundamental	Solución Autotemporizada	
Ninguna dependencia con el futuro		
Requerimientos de memoria finitos		
Ninguna dependencia analógica		
Operación determinista		
No se permiten cambios simultáneos de entrada	Almacenamiento controlado internamente	Almacenamiento controlado externamente
Cambios espaciados hasta finalizar la operación		Ningún control interno sobre los cambios de estados
		Reloj único

Tabla 4.6. Propiedades que deben cumplir las soluciones secuenciales estudiadas.

- Derivación de la tabla de transiciones.
- Asignación de estados.
- Obtención de la tabla y/o diagrama de estados.

Para llevar a cabo el primer paso, básicamente hay que romper los lazos de realimentación para poder distinguir entre las señales de estado presente y de próximo estado. En este punto vamos a distinguir los esquemas con elementos de memoria y con realimentaciones directas.

- En el primer caso, y suponiendo que el sistema está bien diseñado, los lazos de realimentación serán los que involucren a los elementos de memoria. Por lo tanto, las variables de estado presente serán las salidas de dichos elementos, mientras que las señales de próximo estado serán sus entradas.
- En cambio, cuando tengamos realimentaciones directas, hay que identificar cada lazo de realimentación y asignarle una señal de estado a cada uno. La identificación de estos lazos se produce si se consigue cerrar el lazo a través de las conexiones y puertas del circuito.

La generación de la tabla de transiciones es directa, a partir de la funciones combinacionales obtenidas en el paso anterior.

Para la asignación de estados, se le asigna un identificador a cada combinación de señales de estados. Después se sustituye cada combinación por su identificador en la tabla de transiciones, y obtendremos la tabla de estados. Por último, esta tabla se pasa a diagrama de estados.

4.1. Ejemplo de Análisis de Sistemas con Realimentación Directa.

Como primer ejemplo vamos a abordar el problema de análisis del circuito mostrado en la figura 4.6. En primer lugar, hay que identificar qué tipo de sistema estamos tratando: síncrono o asíncrono. Como no está presente ningún tipo de elemento de memoria, en el caso de que se trate de un sistema secuencial, estaremos ante un sistema secuencial con realimentación directa.

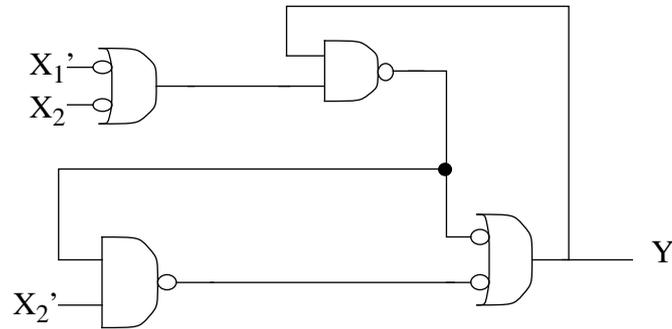


Figura 4.6.- Ejemplo de análisis de sistemas con realimentación directa.

El siguiente paso en este tipo de circuitos consiste en identificar los lazos de realimentación directa. La presencia de estos lazos será el identificativo de que nos encontramos ante un sistema secuencial. Una posible forma de obtener estos lazos consiste en inspeccionar las señales de salidas de las puertas que se dirigen a entradas de puertas anteriores. Si podemos cerrar un lazo, estaremos ante uno de estos lazos. En el caso del circuito que nos ocupa, encontramos dos lazos de realimentación, como podemos ver en la figura 4.7.

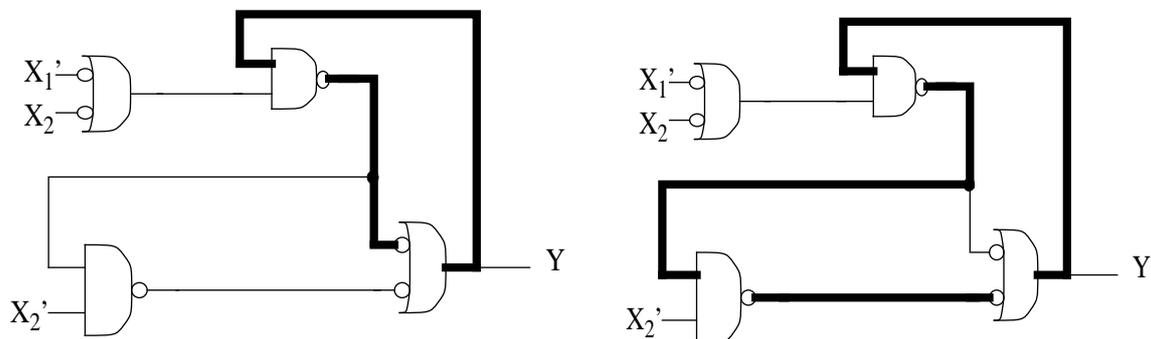


Figura 4.7.- Obtención de los lazos de realimentación del ejemplo de la figura 4.6.

Por lo tanto, asignaremos una variable de estado a cada lazo de realimentación. En concreto consideraremos la señal q en la salida de la segunda puerta NAND y la señal y como la señal de salida, según la figura 4.8.

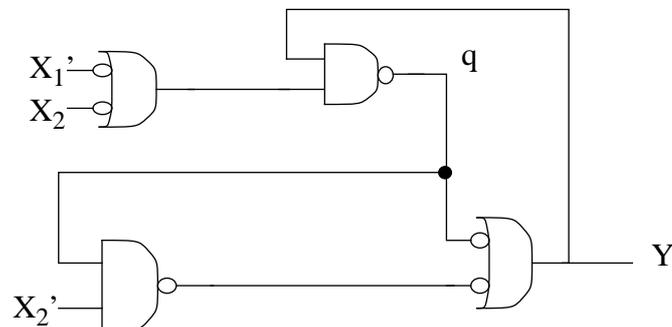


Figura 4.8.- Asignación de las señales de estado al ejemplo de la figura 4.6.

El siguiente paso consiste en obtener las funciones combinacionales de las señales de estado y de salida, siguiendo las técnicas de análisis de circuitos combinacionales estudiadas en el cuatrimestre anterior. Por lo tanto, obtenemos las siguientes funciones:

$$Q = (y*(x_1+x_2'))'$$

$$Y = q' + x_2'*q$$

Una vez que hemos obtenido las funciones combinacionales, generamos la tabla de transiciones en la que tendremos tantas columnas como combinaciones de señales de entrada y tantas filas como combinaciones de señales de estado. Dicha tabla se muestra en la tabla 4.7.

QY		x_1x_2			
		00	01	11	10
qy	00	11	11	11	11
	01	01*	11	01*	01*
	11	01	10	00	01
	10	11	10*	10*	11

Tabla 4.7. Tabla de transición correspondiente al ejemplo de la figura 4.6.

Si elegimos el siguiente alfabeto de estados:

- A = “00”
- B = “01”
- C = “11”
- D = “10”

CamBIANDO las codificaciones por el alfabeto elegido, la anterior tabla de transiciones se transforma en la tabla de estados mostrada en la tabla 4.8.

Y	x_1x_2			
	00	01	11	10
A	C,1	C,1	C,1	C,1
B	B,1*	C,1	B,1*	B,1*
C	B,1	D,0	A,0	B,1
D	C,1	D,0*	D,0*	C,1

Tabla 4.8. Tabla de estados correspondiente al ejemplo de la figura 4.6.

Para pasar de la tabla de estados al diagrama de estados, solamente debemos enmarcar los estados y generar los arcos hacia los próximos estados que encontremos en la tabla, utilizando como etiquetas de dichos arcos la combinación de las señales de entrada y el valor de las señales de salida separados por una coma. El diagrama correspondiente se puede ver en la figura 4.9.

Con el diagrama y/o tabla de estados se acaba el análisis formal. No obstante, llegado a este punto podemos estudiar los fenómenos transitorios que nos podríamos encontrar con la implementación analizada.

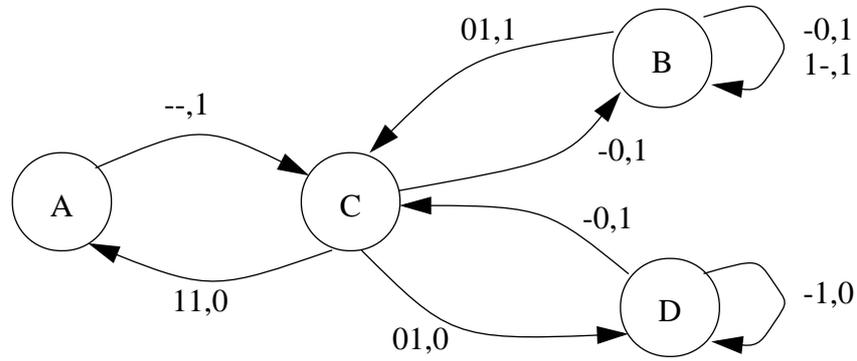


Figura 4.9.- Diagrama de estado correspondiente al ejemplo de la figura 4.6.

Por tratarse de un sistema con realimentación directa, estamos ante un circuito que opera en el modo fundamental. Por lo tanto, la implementación de las funciones combinacionales debe ser libre de azares. En cambio, existe un azar lógico estático en la función Y para $x_2=0$.

$$Y = q' + 0' * q = q' + q$$

En cuanto a los fenómenos relativos a la sincronización, empezaremos por los comportamientos cíclicos. Podemos comprobar que existe un ciclo entre los estados A y C para la combinación $x_1x_2="11"$. En el caso de las carreras, vemos que existen carreras para todas las combinaciones de entrada en el estado 00. Salvo en el caso de la combinación $x_1x_2="11"$, todas las demás carreras no son críticas, como podemos ver en la figura 4.10.

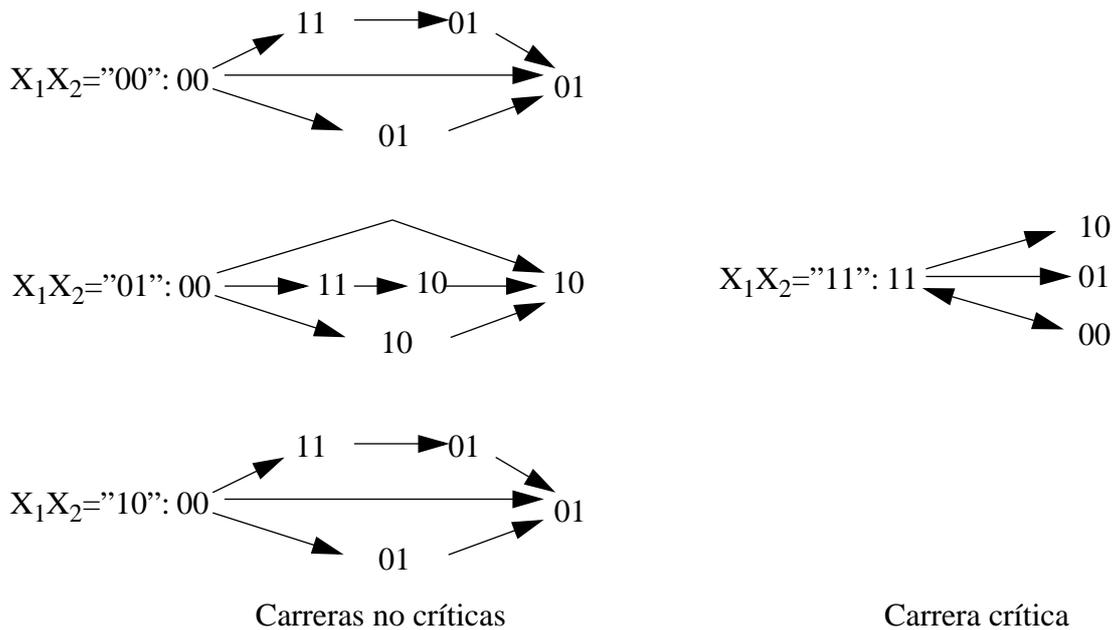


Figura 4.10.- Carreras presentes en el circuito de la figura 4.6.

4.2. Ejemplo de Análisis de Sistemas con Biestables.

Consideremos el circuito mostrado en la figura 4.11, donde consideraremos los tres tipos de biestables estudiados: biestables transparentes, latches y flip-flops.

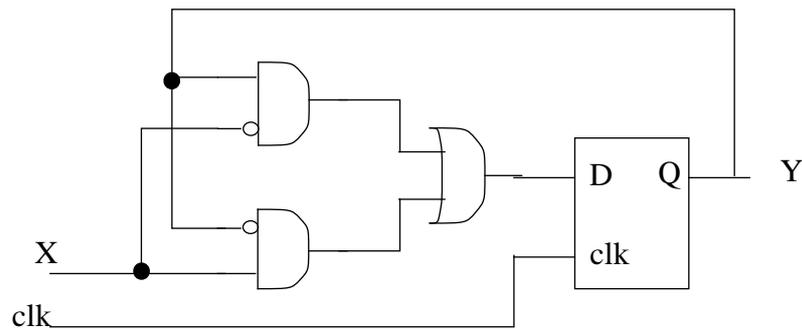


Figura 4.11.- Ejemplo de análisis de sistemas con biestables.

Por tener elementos de memoria, las señales de estado son asignadas a dichos elementos de tal forma que las señales de próximo estado dependerán de las entradas y sus salidas serán las señales de estado presente. Por lo tanto, tendremos una sola señal de estado presente q, que en este caso particular coincide con la señal de salida, y su señal de próximo estado Q. Una vez que hemos identificado todas las señales de estado, generaremos las funciones combinacionales, las cuales son:

$$Q = x*q' + x'*q$$

$$Y = q$$

En estas ecuaciones no aparecen la señal clk porque no intervienen en la función lógica sino que determina el instante temporal en el que se produce el cambio de estado.

Una vez que hemos obtenido las funciones combinacionales, pasamos a generar las tablas de transición:

Q,Y		X	
		0	1
q	0	0,0	1,0
q	1	1,1	0,1

Tabla 4.9. Tabla de transición del circuito de la figura 4.11.

Si elegimos el siguiente alfabeto de estados:

- A = "0"
- B = "1"

La anterior tabla de transiciones se transforma en la tabla de estados mostrada en la tabla 4.10.

Para pasar de la tabla de estados al diagrama de estados, solamente debemos enmarcar los estados y generar los arcos hacia los próximos estados que encontremos en la tabla, utili-

		X	
		0	1
A	A, 0*	B, 0	
B	B, 1*	A, 1	

Tabla 4.10. Tabla de estados correspondiente al ejemplo de la figura 4.11. zando como etiquetas de dichos arcos la combinación de las señales de entrada y el valor de las señales de salida separados por una coma. El diagrama correspondiente se puede ver en la figura 4.12.

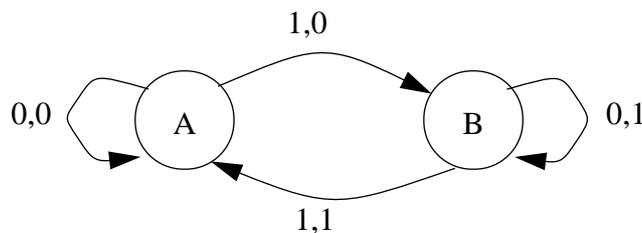


Figura 4.12.- Diagrama de estado correspondiente al ejemplo de la figura 4.11.

En este tipo de circuitos sólo habría que estudiar los fenómenos transitorios en el caso de tener biestables con fases transparentes, es decir, biestables transparentes y latches. Si contamos con este tipo de elementos de memoria, deberíamos estudiar estos fenómenos.

En el caso de los azares lógicos de la implementación combinacional, podemos comprobar que la implementación bajo estudio no presenta ningún tipo de azares, ya que no podemos encontrar ninguna combinación tal que la salida o el próximo estado tenga una apariencia de la siguiente forma:

$$F = a + a'$$

$$F = a * a'$$

En el caso de los comportamientos cíclicos, observamos la existencia de un fenómeno de esta índole para la combinación $x="1"$.

Finalmente no existe ninguna carrera ya que solamente tenemos dos estados, y por lo tanto su codificación binaria no puede diferenciarse en más de un bit.

Por último vamos a ver las posibles formas de onda para los tres casos de elementos de memoria, en la figura 4.13.

En estas formas de onda hemos supuesto que el valor inicial de la señal de próximo estado era "0", aunque también podía ser "1" ya que en ambos casos los estados son estables (para la combinación $x="0"$). En el primer caso, la señal clk no interviene puesto que los biestables transparentes no tienen ninguna señal de control. En cambio en los casos de los latches y los flip-flops esta señal sí interviene.

En el caso de los biestables transparentes, el circuito solamente cambiará de estado cuando la señal de entrada x se encuentre a nivel alto. En este momento se producirá un ciclo

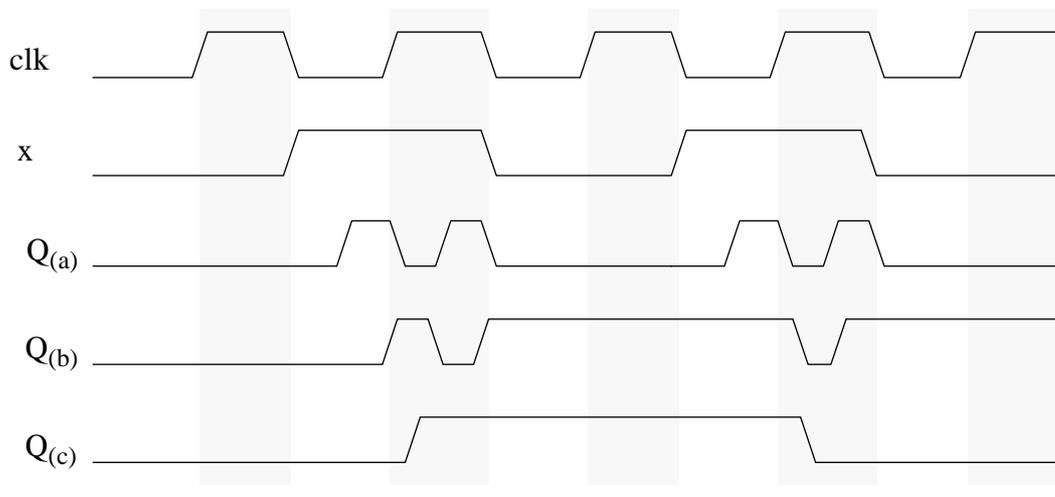


Figura 4.13.- Posibles formas de onda para el circuito de la figura 4.11, suponiendo que los elementos de memoria son (a) biestables transparentes, (b) latches y (c) flip-flops (en los dos últimos casos la señal de control es la señal clk).

entre los dos estados hasta la señal de entrada valga un nivel bajo, momento en el que se almacenará el valor anterior, independientemente del que sea (“0” en nuestro caso).

En el caso de los latches, observamos un comportamiento similar cuando la señal clk valga un nivel alto, es decir, el latch esté en su fase transparente. En cambio, cuando la señal clk valga un valor bajo, el estado no cambiará manteniendo el valor anterior.

En el caso de los flip-flops, solamente se producirán cambios en la transición de subida de la señal clk. El comportamiento es el siguiente: si la señal de entrada es “0”, no se producirá ningún cambio de estado, por ser ambos estables; pero si la señal de entrada vale “1”, habrá un solo cambio de estado (en la transición de la señal clk).

