

Tema III.

Biestables

3.1. Introducción

En el tema anterior hemos estudiado las diferentes formas de construir una puerta lógica, es decir, los componentes de un circuito combinacional. No obstante, hay una parte más importante si cabe que la lógica combinacional: la lógica secuencial. La diferencia entre ambas partes de los sistemas digitales es el tratamiento del tiempo que tiene cada una de ellas, como podemos apreciar en la figura 3.1. En dicha figura podemos apreciar dos circuitos basados en multiplexores con dos arquitecturas similares, junto con una posible descripción en VHDL. La operación del primer circuito es tal que cuando las dos entradas A y B son iguales, la señal de salida es igual a la señal A; mientras que si las entradas son diferentes, la señal de salida es igual a la señal B. En cambio, la operación del segundo circuito se diferencia con respecto a la del primero en que cuando las dos entradas son diferentes, la señal de salida no cambia, es decir, mantiene el valor que tenía antes. Según este comportamiento (visto en las formas de onda), podemos apreciar dos situaciones en las que existen las mismas combinaciones de entrada ($AB = "10"$) y la señal de salida del segundo circuito tiene dos valores diferentes (en el primer caso '0' y en el segundo caso '1').

Esta situación muestra la gran diferencia entre los circuitos combinacionales (el primer caso) y los circuitos secuenciales (el segundo caso): el tratamiento del tiempo. En los circuitos combinacionales, la dependencia con el tiempo es implícita y sólo afecta al retraso del circuito, de tal forma que el comportamiento estacionario siempre será el mismo para las mismas combinaciones de entrada.

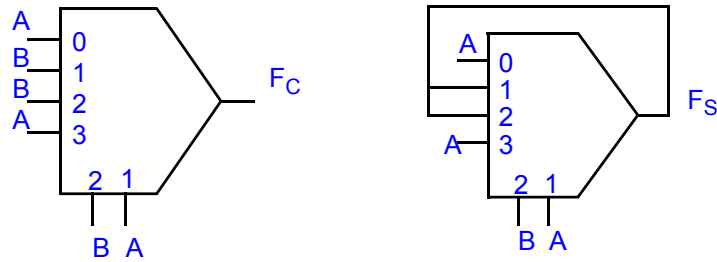
El comportamiento estacionario de un circuito es aquel en el que las señales no cambiarán más hasta que no se produzca un cambio en alguna de las señales de entrada. Dicho valor sería el mismo tratando las puertas lógicas como ideales (con retraso nulo).

En cambio, los circuitos secuenciales muestran una dependencia explícita con el tiempo, de tal forma que el comportamiento estacionario para una misma combinación de entrada puede ser diferente si se evalúan en instantes de tiempo diferentes. Por lo tanto, puede tratarse el tiempo como una entrada más (aunque no es así realmente) y se dice que estos circuitos tienen memoria (ya que como en el caso anterior recuerdan los valores previos).

La arquitectura básica de los circuitos secuenciales se muestra en la figura 3.2. En dicha figura podemos apreciar dos bloques bien diferenciados:

- un bloque combinacional, que realizará la función lógica del circuito
- un elemento de memoria, que dotará al circuito de su capacidad de memoria

En estos circuitos podemos diferenciar tres tipos de señales: las señales de entrada, las señales de salida y las señales de estado. Mientras que las dos primeras tienen el mismo significado que en la lógica combinacional, las señales de estado almacenan la información temporal del circuito (que actuará como la señal de tiempo). Estas señales nos indicarán el estado en el que se encuentra y se encontrará el sistema, cuyo número dependerá de la cantidad de información que sea relevante para el correcto comportamiento del sistema. Realmente vamos a tener dos tipos de señales de estado:



```
signal S1: std_logic_vector(1 downto 0);
```

```
...
```

```
S1(0) <= A;
```

```
S1(1) <= B;
```

```
with S1 select
```

```
  Fc <= A when "00",
        B when "01",
        B when "10"
        A when "11",
        Z when others;
```

```
signal S1: std_logic_vector(1 downto 0);
```

```
...
```

```
S1(0) <= A;
```

```
S1(1) <= B;
```

```
with S1 select
```

```
  Fs <= A when "00",
        Fs when "01",
        Fs when "10"
        A when "11",
        Z when others;
```

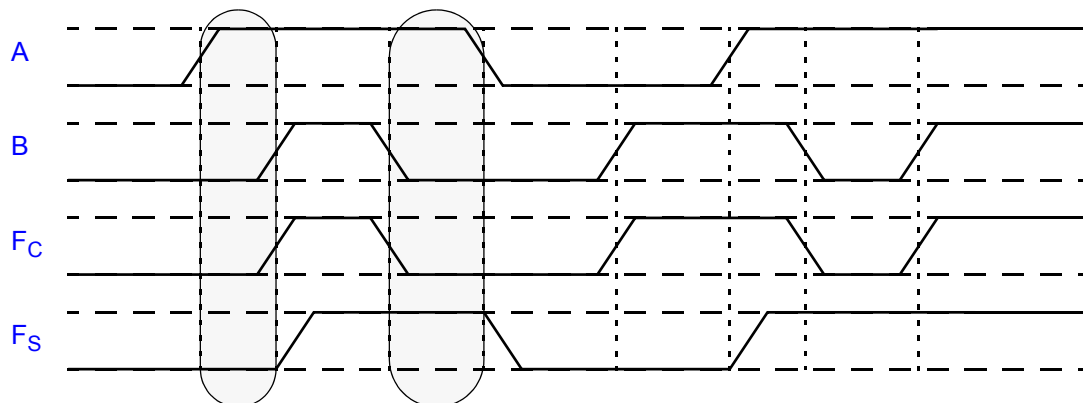


Figura 3.1. Diferencia entre el comportamiento combinacional y secuencial.

- Señales de estado presente, son las señales que indican el estado actual del sistema.
- Señales de próximo estado, son las señales que indicarán el nuevo estado en el que se encontrará el sistema.

El funcionamiento de dicho sistema es el siguiente. Las señales de salida serán una función combinacional de las señales de estado y/o de las señales de entrada (en función del tipo de máquina que estemos tratando: máquina de Mealy o de Moore)

La máquina de Mealy es aquella en la que las señales de salida dependen tanto del estado en el que se encuentre (las señales de estado presente) como de las señales de entrada

La máquina de Moore es aquella en la que las señales de salida únicamente dependen del estado en el que se encuentren (las señales de estado).

Las señales de próximo estado dependerá de las señales de estado presente y de las señales de entrada. Estas señales de próximo estado serán introducidas a las entradas de los elementos de memoria, las cuales pasarán a las salidas (y se convertirán en señales de estado presente) en el siguiente ciclo de operación.

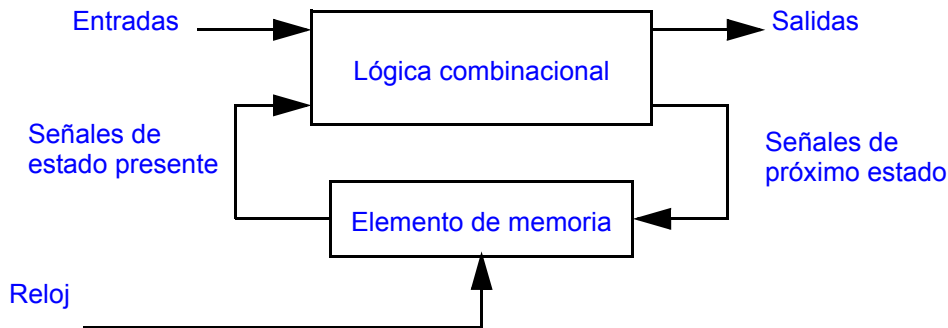


Figura 3.2. Estructura básica de un sistema secuencial.

En el caso de la figura 3.1, el elemento de memoria está formado por la línea de realimentación luego no existe ningún elemento explícito, y la estrategia de temporización se conoce como asíncrona ya que los ciclos de operación vienen determinados por las señales de entrada. En cambio, la figura 3.2 muestra un sistema con un elemento de memoria explícito en el que los ciclos de operación son controlados por una señal global externa, conocida generalmente como reloj; dicha estrategia de temporización se conoce como síncrona.

En el presente tema nos vamos a centrar en los elementos de memoria. En el régimen digital, los elementos de memoria deben ser capaz de almacenar dos valores diferentes, correspondientes a los dos niveles lógicos, por lo que a los elementos de memoria se les suele conocer como biestables.

Un biestable es un elemento capaz de almacenar dos estados (estables) diferentes, coincidiendo con los niveles lógicos.

Un elemento que cumple con la definición anterior se muestra en la figura 3.3a. Dicho elemento está compuesto por dos inversores conectados en serie realimentando las entradas y salidas. El comportamiento es tal como sigue:

- Si $X = '0' \Rightarrow Y = '0'$
- Si $X = '1' \Rightarrow Y = '1'$

es decir, que la salida mantendrá el valor de la señal de entrada. En la figura 3.3b mostramos las características de los dos inversores conectados en serie y la rampa debida a la realimentación. Podemos apreciar que existen tres puntos de equilibrio (puntos en los que se cortan las dos gráficas y a los que tenderán las señales X e Y. No obstante, según podemos apreciar en la figura 3.3c que uno de estos tres puntos es inestable, es decir, que un pequeño cambio debido a una perturbación (como puede ser el ruido), sacará al sistema de este punto y lo enviará a uno de los dos restantes, en función del valor de dicha perturbación. Este hecho es común a todos los biestables: la existencia de un punto inestable, conocido como estado metaestable.

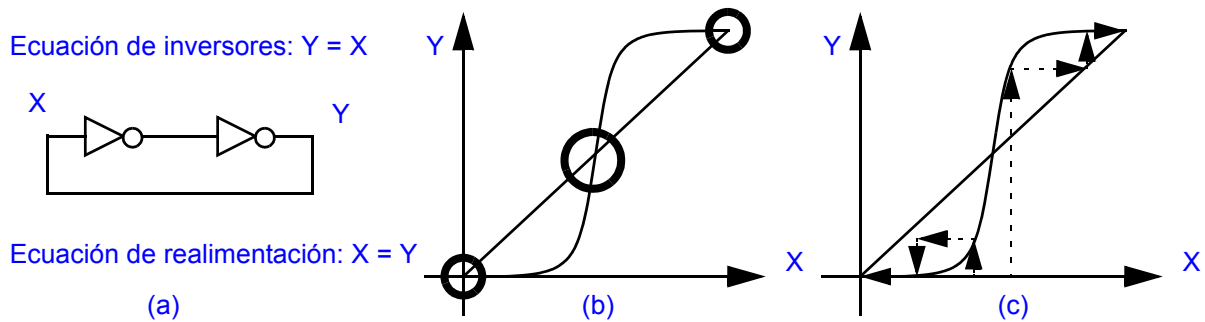


Figura 3.3. Principio de biestabilidad. (a) Elemento de memoria, (b) comportamiento del elemento y (c) comportamiento cuando no se encuentra en ninguna situación de equilibrio.

Según la lógica anterior, cualquier circuito cuya ecuación consista en que la salida sea igual a la entrada tiene el mismo comportamiento que un biestable; por lo que un simple cable tiene dicha característica. Pero en este último caso, ¿dónde se encuentra la capacidad de almacenamiento? Bien dicha capacidad se encuentra en el condensador parásito que tiene asociado dicho cable.

En el caso anterior, la señal de salida siempre sigue a la entrada. Dicha propiedad se denomina transparencia.

La propiedad de transparencia es aquella en la que la señal de salida atiende a un cambio en la señal de entrada.

En función de esta propiedad podemos clasificar los biestables en tres grupos bien diferenciados: biestables transparentes, son aquellos en los que se cumple la propiedad de transparencia en todo momento; latches, son aquellos en los que se cumple la propiedad de transparencia en un intervalo de tiempo (el cual se puede repetir periódicamente y suele ser caracterizado por el intervalo en el que una señal toma un valor determinado); y flip-flops, son aquellos en los que la propiedad de transparencia no se cumple limitándose a un instante de tiempo (caracterizado por una transición).

Tanto en los latches como en los flip-flops, existe una señal que controla el instante o intervalo de tiempo en el que se cumple la propiedad de transparencia. Dicha señal se suele denominar reloj. En la figura 3.4 mostramos dos ejemplos de latches: una pareja de inversores realimentados junto con una llave controlada por la señal de reloj; y un cable (junto con su resistencia y condensador parásito asociados) y una llave controlada por la señal de reloj. En la misma figura se muestran las formas de onda del comportamiento de ambos biestables para una misma señal de entrada y reloj. En ella podemos apreciar que la señal de salida coincide con la señal de entrada cuando la señal de reloj toma el valor '1', es decir, el biestable está en su fase transparente (durante el intervalo en el que el reloj vale '1'). En cambio, cuando la señal de reloj toma el valor '0', la señal de salida permanece sin cambiar, es decir, el biestable está en la fase no transparente.

El comportamiento anterior nos da idea de un problema que muestran todos los biestables controlados por una señal de reloj. Este problema consiste en la incertidumbre que se tiene cuando los cambios del reloj y del dato están demasiado cercanos, como en el caso de los dos últimos pulsos de las señales de salida.

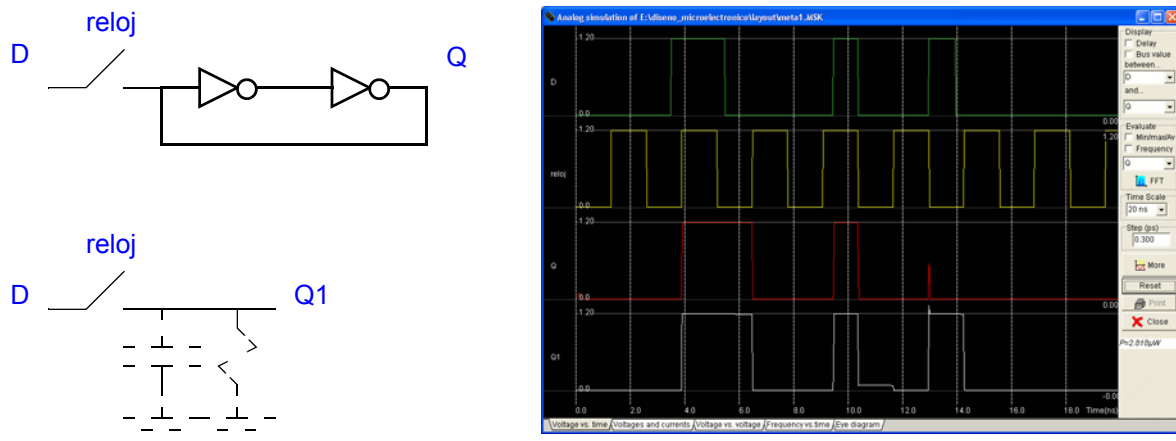


Figura 3.4. Elementos de memoria con señal de reloj.

- En el primero de ellos coinciden la bajada del reloj y del dato. El biestable compuesto por los inversores realimentados alcanza el valor '0' de forma correcta; mientras que el compuesto por el cable no llega al '0', pero la tensión baja lo suficiente como para que se considere un '0', que se recompondrá en el siguiente pulso de reloj.
- En el segundo coinciden la bajada del reloj con la subida del dato. El biestable compuesto por el cable alcanza el valor '1' de forma correcta; mientras que el compuesto por los inversores realimentados intenta completar la subida pero tiene el tiempo suficiente bajando de nuevo a '0'.

Esta situación nos indica que diferentes biestables se comportan de forma diferente ante la cercanía de las transiciones de dichas señales, y por lo tanto, su comportamiento no es predecible. Para eludir esta situación, el comportamiento no predecible, hay que imponer una serie de restricciones temporales. Estas restricciones son:

- Tiempo de setup (t_{setup}),

que se puede definir como el tiempo que la señal de entrada debe permanecer estable antes de que se desactive la señal de reloj

- Tiempo de hold (t_{hold})

que se puede definir como el tiempo que la señal de entrada debe permanecer estable después de que se desactive la señal de reloj

- Anchura de pulso (t_{width})

que se puede definir como el tiempo que la señal de reloj debe permanecer activa para que el almacenamiento se realice de forma correcta

Dichas restricciones se pueden apreciar en la figura 3.5. En dicha figura nos hemos centrado en los latches. No obstante, dichas restricciones son similares en el caso de los flip-flops pero el origen de tiempos estaría centrado en la transición activa (ya que la señal de reloj se activa y desactiva en el mismo instante de tiempo, una transición). Dicha figura valdría para un flip-flop que fuese sensible a la transición de bajada de la señal de reloj. Como ya hemos visto en la figura 3.4, el hecho de no cumplir las restricciones temporales va a llevar al biestable a una

situación parecida a la metaestabilidad, por lo que se dice que el biestable entra en metaestabilidad, aunque no se encuentre la situación de forma lógica.

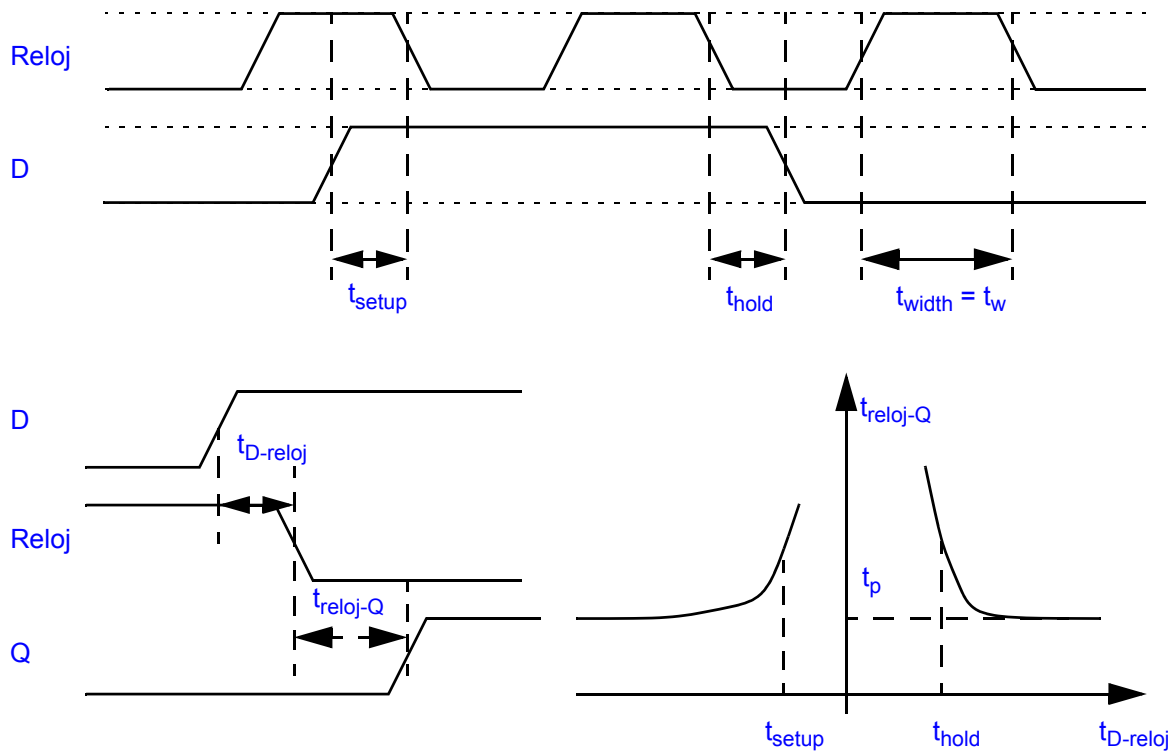


Figura 3.5. Restricciones temporales de los biestables no transparentes.

La forma de obtener estos parámetros consiste en realizar una serie de simulaciones e ir midiendo el retraso de propagación (t_p) del biestable.

El retraso de propagación de un biestable es el tiempo transcurrido para que se almacene un nuevo dato, es decir, el tiempo transcurrido entre la activación de la señal de reloj y el cambio de la señal de salida

Si nos centramos en los tiempos de hola y de setup, en las diferentes simulaciones irán cambiando la diferencia temporal entre la señal de dato y el reloj hasta que el dato deja de ser almacenado (tiempo de setup) o hasta que el nuevo dato empieza a ser almacenado (tiempo de hold). Obviamente no se espera a que los datos dejen de ser o vuelvan a ser almacenados para dar dichos tiempos, sino que se toma un determinado aumento de los retrasos de propagación.

La ventana marginal se denomina al espacio temporal situado entre el tiempo de setup y de hold, en el cual no se tiene certidumbre del comportamiento del biestable.

En el caso de la anchura de pulso, la técnica es similar pero variando la anchura de pulso de la señal de reloj.

Si retomamos el latch compuesto por la llave controlada por la señal de reloj y el cable, en la figura 3.6 mostramos un nuevo comportamiento en el que se ha considerado un mayor diferencia temporal entre el almacenamiento de dos datos. En dicha figura podemos apreciar que el dato no es perfectamente almacenado sino que el valor de tensión va decayendo poco a

poco. En el segundo pulso no se garantiza las restricciones temporales, llevando a una situación de metaestabilidad.

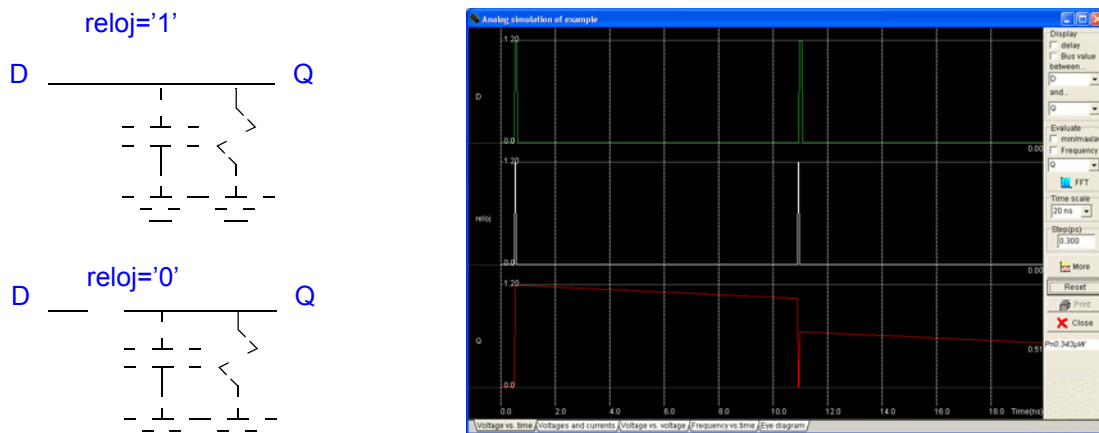


Figura 3.6. Comportamiento del latch formado por la llave y el cable.

La situación de decaimiento de la tensión es debido a la estructura del biestable. Cuando la señal de reloj vale '1', la señal de salida copia a la señal de entrada y en este caso para a nivel alto, el cual es almacenado en el condensador parásito del cable. Cuando la señal de reloj vale '0', la señal de salida está aislada completamente, es decir, está en un estado de alta impedancia. En este nuevo estado el valor de tensión no cambia (ya que estaba almacenado en el condensador parásito); no obstante, la existencia de una resistencia parásita crea un camino de descarga que motivará el cambio lento de dicho valor de tensión, cuya característica principal será el producto de la capacidad y la resistencia.

En función de esta característica, podemos clasificar el almacenamiento en tres grandes grupos:

- Almacenamiento dinámico, es el almacenamiento anterior en el cual el valor almacenado puede llegar a cambiar con el paso de un determinado tiempo. Para evitar la pérdida del dato hay que garantizar que la señal de reloj tenga un periodo menor a la característica temporal de descarga (RC) o volver a almacenar el valor almacenado, lo que se conoce como ciclo de refresco.
- Almacenamiento estático, es el almacenamiento en el que el valor almacenado no cambia a pesar del tiempo que se lleve inactiva la señal de reloj. La condición de estos biestables es que no entran nunca en alta impedancia, es decir, siempre hay un camino habilitado hacia polarización o tierra.
- Almacenamiento semiestático, es el almacenamiento intermedio entre los dos anteriores. En este caso se fuerza a que la señal de salida mantenga un almacenamiento estático aunque algunas señales internas puedan tener almacenamientos dinámicos.

Retomemos el esquema genérico de un sistema secuencial, mostrado en la figura 3.2. En la figura 3.7 mostramos la cuantificación del periodo de la señal de reloj en el caso de tratarse de un elemento de memoria flip-flop sensible a la transición de subida. En dicho caso, y de forma general, la señal de reloj debe permanecer inactivo (el periodo de dicha señal tratándose de flip-flops o el semiperiodo de dicha señal tratándose de latches) tiempo necesario para gene-

rar las señales de próximo estado de forma previa para poder ser almacenadas. Este tiempo será la suma del tiempo de propagación del biestable (para disponer de las señales de estado presente adecuadas a la nueva computación), el tiempo de propagación de la lógica combinatorial (para que las señales de próximo estado sean generadas de forma correcta) y el tiempo de setup del biestable (para que las señales de próximo estado sean almacenadas de forma correcta).

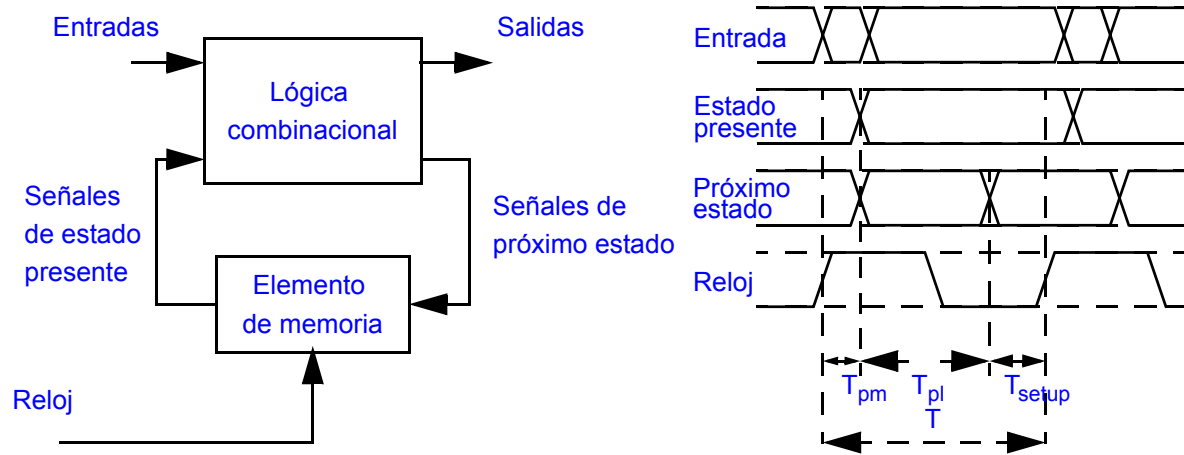


Figura 3.7. Definición del periodo de la señal de reloj.

En el caso de que tengamos varios sistemas secuenciales con la misma arquitectura que la anterior, una técnica muy usada consiste en utilizar la misma señal de reloj para todos los elementos de memoria, la cual se denomina técnica o circuito síncrono.

Un circuito síncrono es aquel en el que todos los elementos de memoria son controlados por la misma señal de reloj.

Luego para que el almacenamiento sea correcto en todos los casos, el retraso de la lógica combinatorial que hay que incluir en la fórmula anterior será el mayor retraso. Por lo tanto, si la lógica combinatorial es muy complicada (con muchos niveles), el retraso combinatorial será muy grande y así la velocidad del sistema (que será la inversa del periodo de la señal de reloj, puesto que es el tiempo necesario para poder realizar una operación) será muy baja.

Para tratar de paliar dicho problema en sistemas muy complicados, se ha optado por una técnica de aceleración denominada pipeline.

La técnica de pipeline consiste en dividir un sistema complejo en varios subsistemas unidos a través de biestables, de tal forma que el periodo de la señal de reloj viene gobernado por el retraso de una de los subsistemas más pequeños (el más lento) en lugar del sistema completo.

Dicha técnica es esquematizada en la figura 3.8 en dos formas diferentes. En el primer esquema todos los elementos de memoria son controlados por la misma fase del reloj, por lo que en cada transición de subida el dato va atravesando todas las etapas de pipeline (ya que los elementos de memoria que se han considerado son flip-flops). En el segundo esquema los elementos de memoria son controlados por fases alternas de la señal de reloj, por lo que las etapas impares son controladas por la subida del reloj y las etapas pares son controladas por su bajada; luego la velocidad del sistema obtenido es el doble que en el caso anterior.

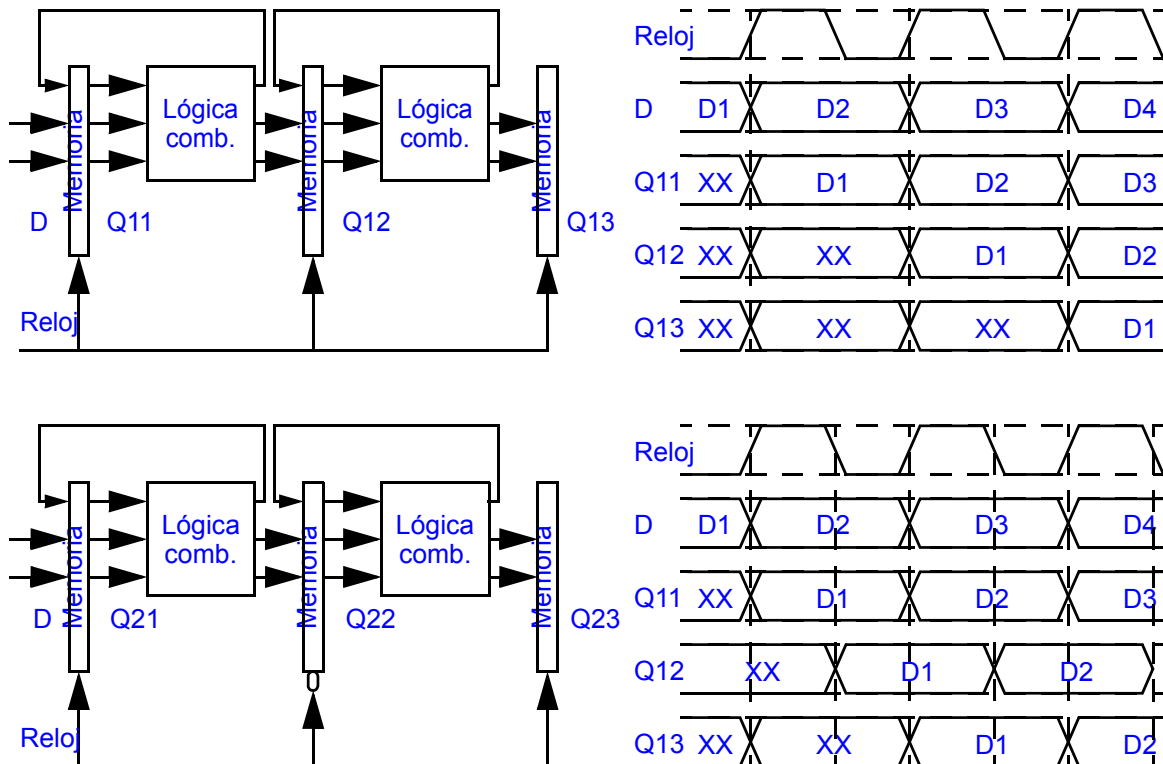


Figura 3.8. Técnicas de pipeline.

Los esquemas anteriores son los ejemplos de dos de los esquemas de temporización más utilizados en la actualidad: esquemas de una fase y de dos fases.

Un esquema de temporización de una fase es un esquema síncrono en el que todos los elementos de memoria son controlados por la misma fase de la señal de reloj.

Un esquema de temporización de dos fases es un esquema síncrono en el que los elementos de memoria de las etapas adyacentes son controlados por las fases complementarias de la señal de reloj.

Las principales características de dichos esquemas (realizando una comparativa de ambos) son las siguientes:

- El esquema de temporización de una fase suele utilizar flip-flops como elementos de memoria, ya que el periodo inactivo de la señal de reloj abarca el periodo completo. En cambio, en el esquema de dos fases pueden emplearse latches ya que el periodo inactivo puede reducirse a un semiperiodo.
- El esquema de una fase muestra una sincronización completa de todas las etapas. En cambio, el esquema de dos fases muestra una doble sincronización: una de las etapas impares y otra diferente de las etapas pares. Esta situación es de importancia ya que toda la actividad se realiza en los mismos instantes (una fase) o se divide en dos instantes (dos fases) reduciéndose de esta manera el ruido de conmutación al existir una mejor distribución en el tiempo.

Veamos un nuevo ejemplo de un biestable transparente denominado biestable RS. La figura 3.9 muestra el esquema del biestable compuesto por dos puertas NOR realimentadas, con dos entradas R y S, y dos salidas Q y nQ (las cuales se suponen que tienen que ser complementadas entre sí). El funcionamiento de dicho biestable es como sigue:

- $R = '1'$ y $S = '0'$, la señal de salida $Q = '0'$, y por lo tanto $nQ = '1'$. Debido a que esta condición asegura que la señal de salida tome el valor '0', dicha condición se denomina condición de reset y dicha señal de entrada reset.
- $R = '0'$ y $S = '1'$, la señal de salida $Q = '1'$, y por lo tanto $nQ = '0'$. Debido a que esta condición asegura que la señal de salida tome el valor '1', dicha condición se denomina condición de set y dicha señal de entrada set.
- $R = '0'$ y $S = '0'$, la señal de salida no sufre ningún cambio, y por lo tanto nQ tampoco. Debido a que esta condición mantiene el valor de salida anterior, dicha condición se denomina condición de almacenamiento.
- $R = '1'$ y $S = '1'$, ambas señales de salida toman el mismo valor igual a '0', es decir, $Q = nQ = '0'$. Dicha condición viola la condición de que ambas salidas sean complementarias, además de ser la fuente en potencia de varios problemas de temporización como carreras. Por lo tanto, dicha combinación debe ser prohibida y no se debe permitir que se produzca en ningún caso.

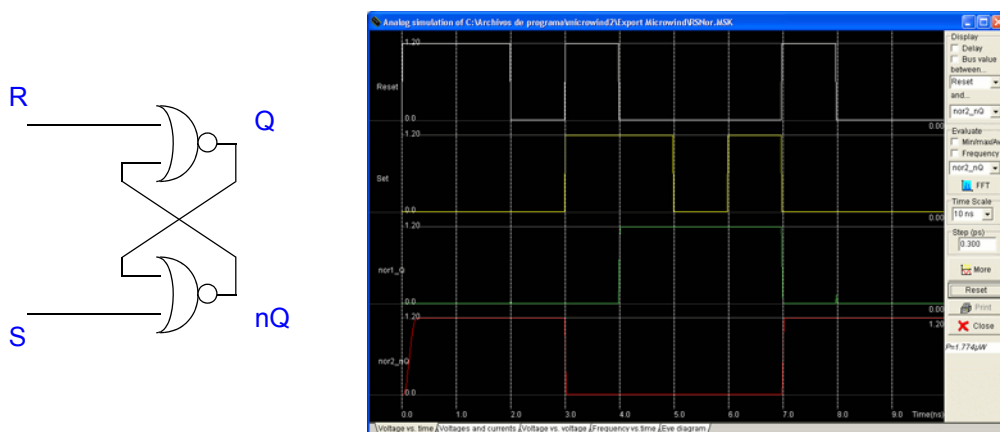


Figura 3.9. Biestable RS junto con su comportamiento.

Podemos encontrar un biestable RS similar simplemente cambiando las puertas NOR por puertas NAND. El comportamiento sería exactamente el mismo intercambiando las condiciones de almacenamiento y prohibida; pero tendríamos los terminales de reset y de set con su mismo comportamiento.

El siguiente paso sería construir un latch RS a partir del biestable transparente. La forma de operación de un latch implica que cuando la señal de reloj tiene un determinado valor (por ejemplo '1'), el latch se debe comportar como un biestable transparente. En cambio, cuando la señal de reloj toma el valor complementario, el latch debe mantener el valor anterior, es decir, adoptar la condición de almacenamiento. Luego, el bloque que hay que incluir antes del biestable latch será una puerta AND en cada entrada, con la señal de reloj como una de sus entradas, como se puede apreciar en la figura 3.10a. También mostramos el comportamiento de dicho latch incluyendo las señales de dato del biestable RS transparente (señales r1 y s1). Podemos

apreciar que las señales $r1$ y $s1$ son iguales a R y S cuando la señal de reloj tiene el valor '1', y son iguales a '0' cuando la señal de reloj tiene el valor '0', es decir, la condición de almacenamiento. En el caso de seleccionar la combinación prohibida (supuestamente por error), vemos que se mantiene la situación de que ambas salidas tienen el mismo valor hasta que se alcanza la condición de almacenamiento, ya sea por las señales R y S o por la señal de reloj.

En el caso de obtener un flip-flop, podemos optar por dos posibles soluciones: utilizar un latch con un bloque que reduce el intervalo activo de la señal de reloj a un instante; o bien utilizar un latch forzando a que las entradas no cambien durante el intervalo activo de la señal de reloj. El primer caso es mostrado en la figura 3.10b. En dicha figura podemos apreciar un latch similar al caso anterior pero en la entrada de la señal de reloj se ha introducido una lógica de reducción del intervalo activo de la señal de reloj. Esta lógica se basa en la introducción de un glitch en dicha señal que coincide con una de las transiciones (en este caso particular la transición de subida). El ejemplo más utilizado es el de implementar el producto

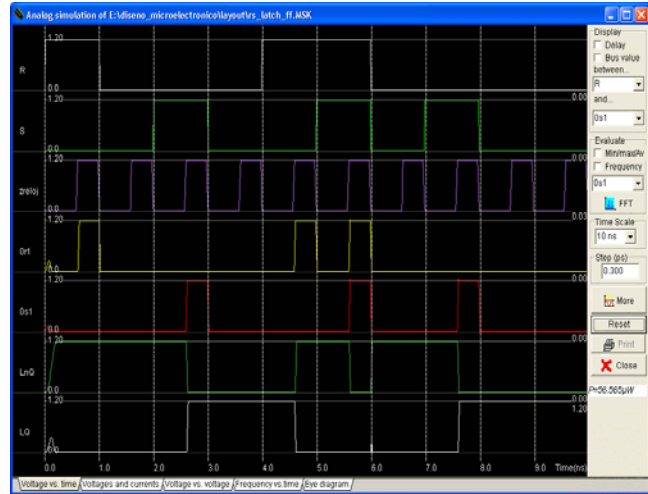
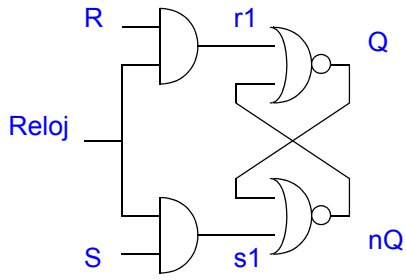
$$\text{reloj1} = \overline{\text{reloj}} \cdot \text{reloj}$$

Para que se produzca el glitch debe estar el suficiente tiempo las dos señales $\overline{\text{reloj}}$ y reloj iguales a '1' (debido al retraso de la inversión). Este hecho es la explicación de la existencia de tres inversores en el camino de la inversión. En las formas de onda podemos apreciar la nueva señal de reloj generada, reloj1 , cuyo nivel activo coincide únicamente con la transición de subida de la señal de reloj. También podemos advertir que cuando se selecciona la combinación prohibida (supuestamente por error), las señales de salida no dejan de ser complementadas debido a la regeneración de los niveles en la condición de almacenamiento. Por lo tanto, salvamos el problema de que las señales de salida no sean complementadas, pero no podemos predecir el valor obtenido.

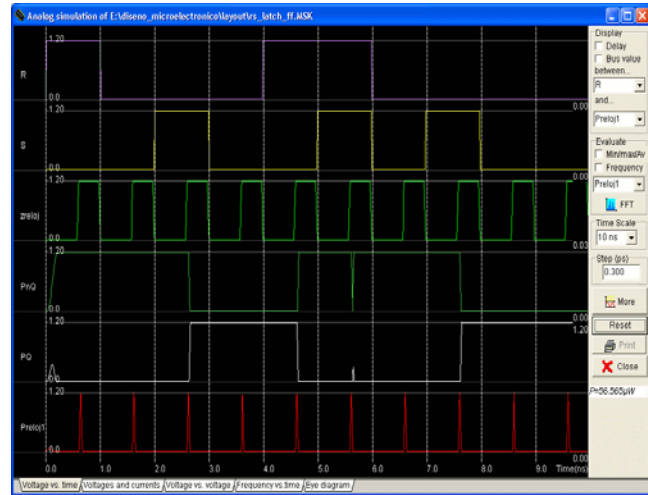
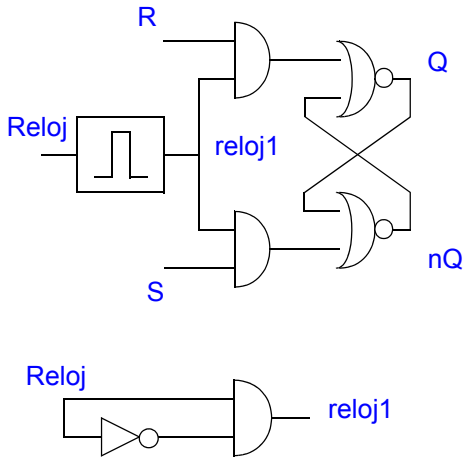
En el segundo caso tenemos que forzar que durante el intervalo activo de la señal de reloj no se produzca ningún cambio en las entradas; de esta forma el biestable sólo podrá tener una transición como máximo coincidiendo con el principio del intervalo, es decir, con la transición. Una forma muy común de conseguir este comportamiento es utilizar un nuevo latch controlador por la fase complementaria del latch en cuestión, como se aprecia en la figura 3.10c. Los flip-flops desarrollados de esta forma se denominan flip-flops maestro-esclavo (master-slave), de tal forma que el primer latch es el maestro (por realizar la función lógica) y el segundo es el esclavo por realizar la función de seguidor. Es importante indicar que el seguidor funcione únicamente como seguidor, para obtener la función lógica del biestable en cuestión. En el comportamiento se puede ver como las señales de entrada del segundo latch no cambian durante su intervalo activo, los cambios siempre se producen en el intervalo inactivo.

También podemos encontrar configuraciones híbridas que utilizan una mezcla de las dos soluciones anteriores, como veremos en el apartado correspondiente.

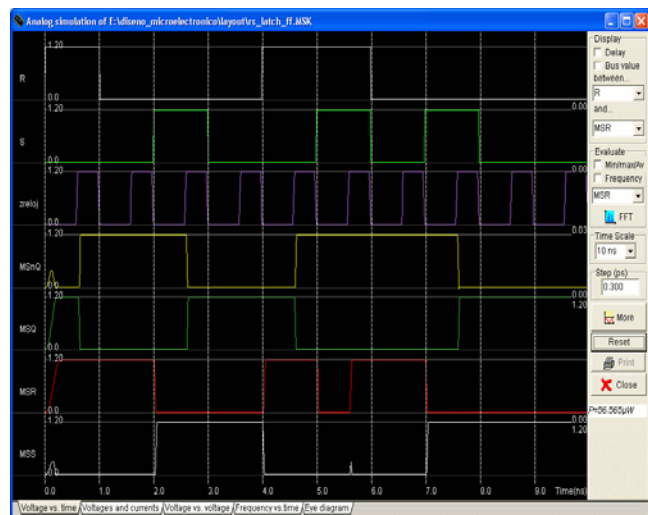
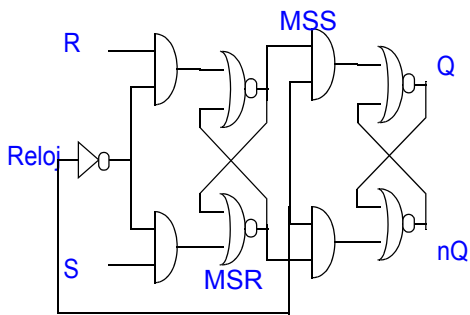
El siguiente paso consiste en decidir el tipo de biestable más interesante. En primer lugar, los latches y flip-flops son más utilizados ya que su señal de control aumenta en gran medida la controlabilidad de su funcionamiento, muy útil para cuestiones de test de circuitos. Entre el latch y el flip-flop, el flip-flop es más observable que el latch puesto que únicamente permite un cambio en el periodo de la señal de control, mientras que el latch permite tantos cambios como sean posibles en el intervalo activo de la señal de control. Luego si entramos en situaciones problemáticas como pueden ser situaciones cíclicas o carreras, tenemos la seguridad de controlar el tiempo en el cual se producen los diferentes cambios (que coincidirán con las tran-



(a)



(b)



(c)

Figura 3.10. (a) Implementación del latch RS y, (b) y (c) implementaciones de flip-flops RS.

siciones activas de la señal de control); mientras que en los latches, los cambios en el intervalo activo de la señal de control únicamente dependerán del retraso del circuito, y no de ningún evento controlable por el usuario.

Por lo tanto, debido a la mayor controlabilidad de los eventos de los flip-flops, estos elementos de memoria son los más utilizados a pesar de requerir mayores recursos de hardware.

A la hora de implementar los flip-flops, es necesario determinar la arquitectura así como las geometrías adecuadas. De hecho hay un número relativamente alto de flip-flops, debido a que son uno de los componentes que más intervienen en las prestaciones del sistema completo, tal que

- los elementos de memoria son la principal fuente de ruido de conmutación
- los elementos de memoria son la principal fuente de consumo de potencia
- los elementos de memoria son los elementos en los cuales se manifiesta la velocidad del sistema

Por lo tanto hay que desarrollar elementos que consuman poca potencia, sean muy rápidos y produzcan poco ruido. No obstante, los dos primeros parámetros, consumo de potencia y velocidad, son complementarios en el sentido de que cuando uno aumenta, el otro disminuye y viceversa, tal como se muestra en la figura 3.11. Por lo tanto, la figura de mérito que se suele utilizar es el producto retraso-potencia, PDP (power-delay product), en el cual se muestra la influencia de ambos parámetros en uno solo. Optimizando el PDP se obtiene una implementación con una buena solución tanto para el consumo de potencia como para el retraso. Además habría que optimizar el ruido de conmutación, así como las restricciones temporales.

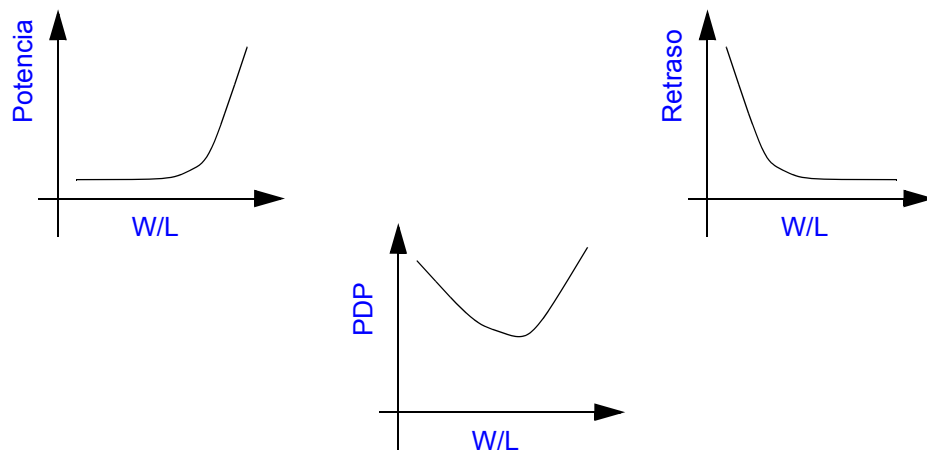


Figura 3.11. Parámetros de optimización en el desarrollo de biestables

3.2. Configuraciones edge-triggered

A continuación vamos a tratar algunos ejemplos de flip-flops basados en pulsos. Dichos ejemplos son: SPPFF (Single-Phase Pulsed Flip-Flop), HLFF (Hybrid Latch Flip-Flop) y uno de los flip-flops utilizados en el procesador AMD K6.

El esquema a nivel de transistores del SPPFF es mostrado en la figura 3.12a. Podemos apreciar que la generación del pulso del nuevo reloj es realizado a través de los tres inversores y la conexión serie de dos transistores NMOS (operación AND). Dicha conexión habilita una estructura dinámica, la cual sirve de entrada a un inversor triestado controlado por el nivel alto de la señal de control (clk). La operación de este flip-flop es como sigue:

- Cuando la señal de reloj (clk) toma el valor '0', el nodo interno del flip-flop está precargado a '1', mientras que el nodo de salida se encuentra en alta impedancia manteniendo el valor anterior.
- Cuando la señal de reloj sube a '1', las señales cd y clk toman ambas el valor '1' momentáneamente (durante el retraso de los tres inversores), por lo que la influencia de la señal de entrada D puede llegar a la salida obteniéndose que $Q = D$.
 - Si $D = '1'$, su transistor NMOS conduce colocando en el nodo interno un '1', y en la salida un '0'.
 - Si $D = '0'$, su transistor NMOS está cortado deshabilitando el camino a tierra, y el nodo interno mantiene de forma dinámica el valor de precarga, es decir, '1', que colocará en la salida un '0'.
- Cuando la señal de reloj se mantiene a '1', el nodo interno se mantiene en alta impedancia almacenando de forma dinámica el valor anterior, el cual es negado por el inversor de salida.

Las principales características de este flip-flop son las siguientes:

- Estamos ante un elemento con almacenamiento dinámico, por lo que necesitará un reloj relativamente rápido o un esquema de refresco.
- El intervalo de transparencia es el retraso de tres inversores.
- Si sustituimos el transistor controlado por la señal de entrada por un árbol NMOS, todo el bloque formará una etapa de pipeline, incluido la lógica combinacional y el elemento de memoria que almacena el valor de salida de la lógica combinacional, como se puede ver en la figura 3.12b.

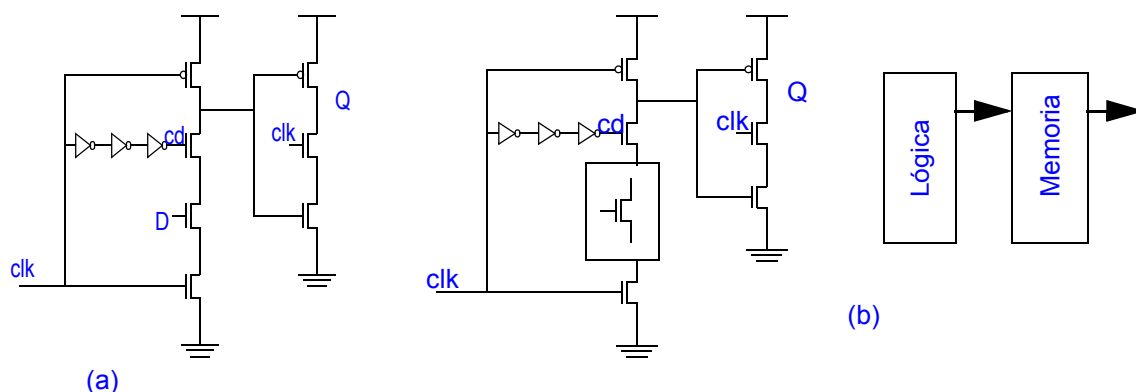


Figura 3.12. (a) Esquema del SPPFF. (b) Esquema del HLFF.

El esquema del HLFF se muestra en la figura 3.13. Dicho esquema se puede ver como una modificación del SPPFF; de hecho, es la versión estática del SPPFF. Los nuevos transistores PMOS (controlados por D y cd) fuerzan el almacenamiento estática de la primera parte del flip-flop, es más, la primera parte realiza la función lógica

$$\overline{\text{clk} \cdot \text{cd} \cdot \text{D}}$$

La segunda parte se puede ver como un nuevo flip-flop con la misión de aislar el valor de pre-carga de la primera parte. Por último, la pareja de inversores realimentados almacenan de forma estática el nodo de salida. Uno de dichos inversores (el etiquetado con un asterisco) es débil en el sentido de que la razón W/L de sus inversores son menores para permitir el posible cambio de valor en una nueva transición de subida.

Las principales características del HLFF son similares a las del SPFF exceptuando el cambio del carácter dinámico por el estático.

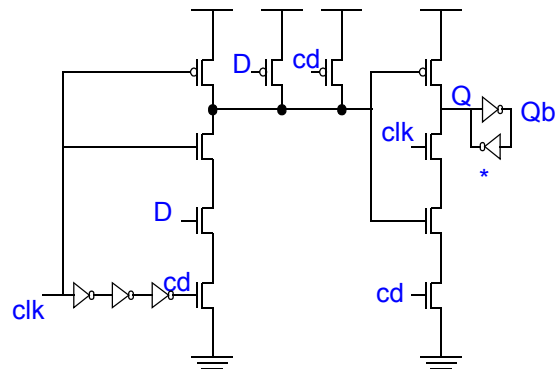


Figura 3.13. Esquema del HLFF.

El esquema a nivel de transistores del flip-flop de doble raíl utilizado en el procesador AMD K6 se muestra en la figura 3.14. Dicho flip-flop tiene una propiedad de autoreset de tal forma que en el mismo momento en que se almacena el nuevo valor se resetea; por lo tanto, vamos a abordar el funcionamiento de dicho flip-flop sin esta propiedad para posteriormente indicar el funcionamiento de la lógica de autoreset. Estamos ante una estructura diferencial cuyo bloque de carga está formado por un amplificador de sensibilidad de nivel alto y un par de transistores NMOS de pre-carga que, junto con tres inversores en cascada, realizan la operación AND de la señal de reloj y su negada. El funcionamiento de dicho flip-flop es el siguiente:

- Cuando la señal de reloj toma el valor '0', el camino al nodo de tierra está cortado por lo que el nodo de salida del árbol diferencial (compuesto por dos transistores controlados por la señal de entrada y su complemento) conectado a la tierra mantiene dicho valor de forma dinámica. Este '0' se convierte en '1' a través del amplificador de sensibilidad del bloque de carga.
- Cuando la señal de reloj sube a '1', el camino a tierra se habilita por lo que el árbol diferencial coloca en el nodo de salida el valor correspondiente que será negado en la salida complementaria por el amplificador de sensibilidad.
- Cuando la señal de reloj se mantiene en '1', estamos ante el mismo comportamiento de que cuando dicha señal toma el valor '0'.

El bloque de autoreset detecta cuando las dos señales de salida toman diferente valor (operación NOR). Dicha detección es retrasada a través de dos inversores, los cuales controlan a dos transistores que conectan las salidas a nivel bajo (a través de los inversores de salida).

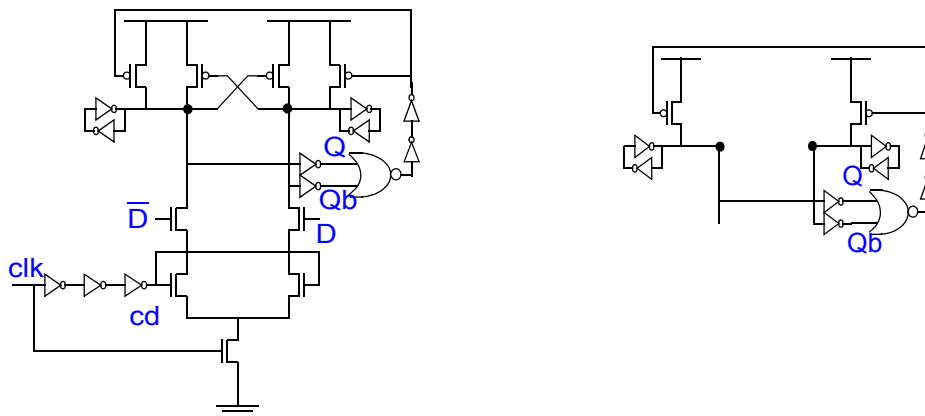


Figura 3.14. Esquema del flip-flop de doble raíl utilizado en el procesador AMD K6.

Las principales características de este flip-flop son las siguientes:

- Tiene la propiedad de autoreset.
- Puede disponer de una lógica combinacional empotrada (igual que los casos SPPFF y HLFF), obteniendo una etapa pipeline completa en un solo bloque. Para ello se debe sustituir los dos inversores controlados por la señal de entrada (que formarían el árbol diferencial del par seguidor-inversor) por el árbol NMOS diferencial de la función lógica que se desea empotrar.
- La anchura de pulso será el correspondiente al retraso de los tres inversores.

3.3. Configuraciones maestro-esclavo

Los ejemplos que vamos a ver de las configuraciones maestro esclavo son el registro convencional (utilizando latches basados en llaves de paso, multiplexores), el registro maestro-esclavo de transistores de paso utilizado en el procesador PowerPC, el registro C²MOS (Clocked CMOS) y la configuración maestro-esclavo basado en celdas estándares.

En la figura 3.15 mostramos el latch y el registro maestro-esclavo convencional, utilizando un esquema de multiplexores. El latch, figura 3.15a, funciona de la siguiente forma:

- Cuando la señal de reloj toma el valor '0', la llave de realimentación está cortada mientras que la llave del camino directo está conduciendo. Por lo tanto, la señal de salida es el complemento de la señal de entrada.
- Cuando la señal de reloj toma el valor '1', la llave de realimentación es la que conduce, por lo que tenemos funcionando un biestable formado por la pareja de inversores realimentados.

Luego, esta latch está controlado por el nivel bajo de la señal de reloj. El registro, figura 3.15b, está formado por dos latches como los anteriores: el primero controlado por el nivel bajo de la señal de reloj; y el segundo controlado por el nivel alto. De esta forma, la señal de salida (la salida del segundo latch) cambiará con la subida de la señal de reloj.



Figura 3.15. Esquema del registro maestro-esclavo convencional. (a) Latch y (b) registro.

Las principales características de este registro son las siguientes:

- Existen versiones en las que las llaves son implementadas como llaves CMOS, para que no exista degradación en ninguno de los niveles lógicos, o llaves NMOS, con la condición de que el nivel alto degradado es regenerado con el inversor de salida.
- Las llaves CMOS muestran una mayor capacidad que las llaves NMOS, lo cual se traduce en un mayor retraso y consumo de potencia. En cambio las llaves NMOS provocarán que el transistor PMOS del inversor no esté completamente cortado cuando pasa un '1', por lo que se consume potencia en estática.

Una nueva implementación de registros maestro-esclavos es desarrollada utilizando el latch C^2 MOS. Dicho latch es mostrado en la figura 3.16a, y está formado por cuatro inversores controlados por la señal de entrada, la señal de reloj y su negada. El funcionamiento de dicho latch es el siguiente:

- Cuando la señal de reloj toma el valor '0', los dos transistores controlados por la señal de reloj conducen reduciendo el circuito completo a un inversor.
- Cuando la señal de reloj toma el valor '1', los dos transistores controlados por la señal de reloj están cortados por lo que el nodo de salida se encuentra en alta impedancia manteniendo el valor anterior.

Luego este latch está controlado por el nivel bajo de la señal de reloj. De hecho, este latch se puede ver como el conjunto de una llave de paso y un inversor. Dependiendo de qué conjunto llave-inversor se sustituya por el latch C^2 MOS, podemos encontrar dos tipos de registros diferentes.

El caso de la figura 3.16b mostramos el latch C^2 MOS utilizado en el procesador PowerPC. En este caso, únicamente se sustituye el primer conjunto llave-inversor por el latch C^2 MOS, tanto del maestro como del esclavo. En cambio, en la figura 3.16c mostramos el registro C^2 MOS convencional. En este caso se sustituyen el primer conjunto llave-inversor y la llave de realimentación por latches C^2 MOS en el maestro y el esclavo, manteniendo el inversor de realimentación (el cual es necesario para dotar de un pequeño retraso y que las llaves del siguiente latch C^2 MOS puedan tener el tiempo suficiente para estabilizarse).

Las principales características de los registros maestro-esclavo basados en latches C^2 MOS son las siguientes:

- El consumo de potencia debido a la realimentación es pequeña.

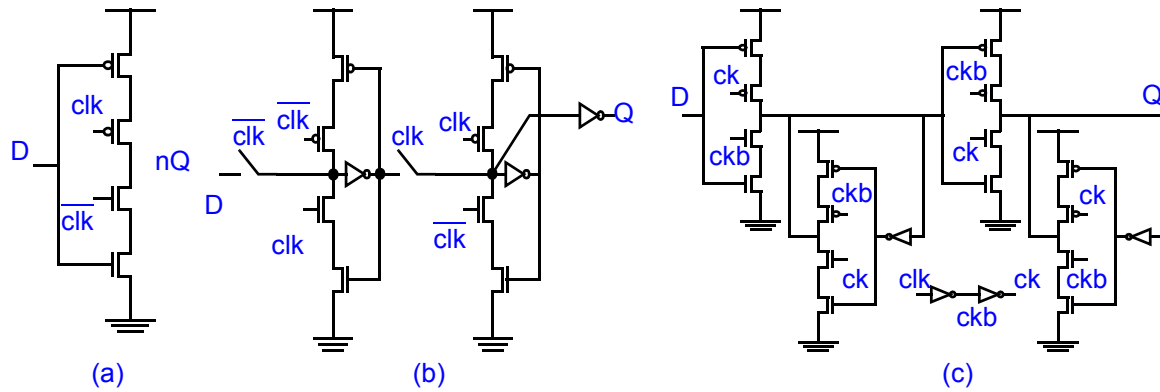


Figura 3.16. (a) Latch C²MOS. (b) Registro C²MOS de PowerPC. (c) Registro estático C²MOS.

- En el caso del registro convencional, las fases de la señal de reloj deben ser generadas localmente, con el fin de obtener unas señales de reloj lo más ideales (con transiciones verticales) posibles.
- Las llaves del registro utilizado en el PowerPC son llaves CMOS.

Un nuevo latch es el que implementa directamente la ecuación del biestable con celdas estándar. Esta ecuación es:

$$Q = D \cdot \text{clk} + Q \cdot \overline{\text{clk}}$$

como se muestra en la figura 3.17a. En dicha figura, las operaciones AND y NOR es incluidas dentro de un mismo recuadro ya que existe una celda estándar que implementa dicha operación, la cual se suele conocer como AOXXX, donde las “X” nos dan idea del número de entradas de las puertas AND, OR y de la complementariedad de la salida.

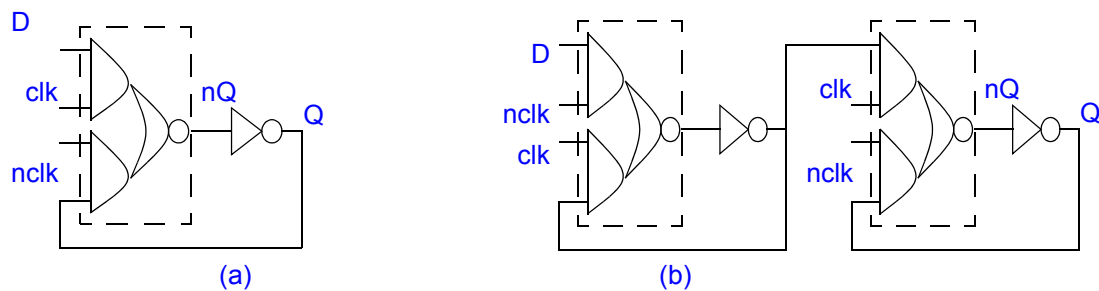


Figura 3.17. (a) Latch implementado con celdas estándares. (b) Registro maestro-esclavo implementado con celdas estándares.

Las principales características de este registro son las siguientes:

- Facilidad de diseño
- Necesidad del inversor de salida para provocar un retraso y estabilizar el camino de realimentación.

3.4. Configuraciones híbridas

Por último tenemos otro tipo de configuraciones que no se pueden englobar en ninguna de las anteriores. No obstante, debido a que tiene características de las dos anteriores la hemos definido como configuración híbrida.

El ejemplo más significativo de esta configuración es el registro SAFF (Sense-Amplifier Flip-Flop) basado en una estructura diferencial, el cual es mostrado en la figura 3.18. En dicho registro, las características de las configuraciones anteriores son las siguientes:

- De la configuración edge-triggered tiene la influencia en el límite de la transición de subida del reloj. Pero a diferencia de los anteriores, en lugar de generar un pequeño pulso basado en la generación de un glitch, deshabilita la influencia del reloj y del dato con el transistor T, excepto en la transición debido al retraso de dicho transistor.
- De la configuración maestro-esclavo tiene un segundo latch (en concreto un latch RS basado en puertas NAND), el cual tiene la misión de filtrar el dato de precarga.

El funcionamiento de dicho registro es el siguiente:

- Cuando la señal de reloj se encuentra en nivel bajo, la estructura diferencial se encuentra en la fase de precarga. Luego el árbol diferencial está deshabilitado por el transistor NMOS de precarga. Los nodos de salida de la estructura diferencial toman ambas el valor '1' debido a los transistores PMOS de precarga. Dicha combinación es la condición de almacenamiento del latch RS por lo que se mantiene el valor anterior.
- Cuando la señal de reloj sube a nivel alto, el árbol diferencial se habilita a través del transistor NMOS de precarga. Entonces una de las salidas del árbol diferencial toma el valor '0', mientras que la complementaria permanece a nivel alto debido a la precarga anterior.
- Cuando la señal de reloj se mantiene en nivel alto, ha transcurrido el suficiente tiempo para que el '0' pase a la salida complementaria del árbol diferencial, por lo que ya ambas salidas toman el valor '0'. En esta situación un nuevo cambio en la señal de entrada no provoca ningún cambio en las salidas del árbol diferencial y por tanto en las salidas del flip-flop.

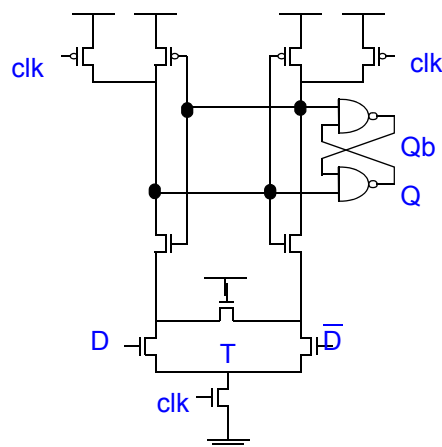


Figura 3.18. Esquema a nivel de transistores del registro SAFF.

Las principales características del registro SAFF son las siguientes:

- Tiene la posibilidad de tener una lógica combinacional empotrada para componer por sí sola una etapa de pipeline.
- El intervalo de transparencia coincidirá con el retraso de propagación del transistor T.
- Tiene un buen comportamiento de retraso y consumo de potencia debido a la estructura diferencial. De hecho, la mayor contribución es la debido al latch RS.