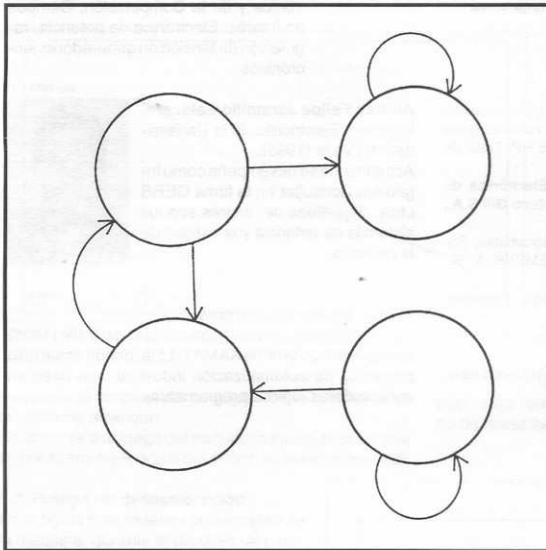


Consideraciones de diseño de las máquinas secuenciales sincrónicas

(Primera parte de dos)



□ Luis Alberto Gómez
Arizabaleta, M.Sc
Ingeniero Electricista
Profesor auxiliar
Universidad del Valle

□ Diego Fernando Duque
Betancourth
Ingeniero Electricista
Universidad del Valle

RESUMEN

Las máquinas secuenciales sincrónicas constituyen la columna vertebral en el diseño de sistemas digitales complejos, tales como microprocesadores, controladores de video, controladores de sistemas de comunicación sincrónica y asincrónica, controladores de floppy, procesadores digitales de se-

ñales, controladores industriales etcétera. La teoría del diseño de dichas máquinas establece el procedimiento para la síntesis de los circuitos que constituirán una «Máquina Secuencial Sincrónica» (en adelante MSS). Sin embargo, existen consideraciones de tipo práctico que limitan el buen funcionamiento de estos sistemas, haciéndolos poco fiables, a menos que se tengan en cuenta ciertas reglas de diseño.

En este artículo se expondrá la teoría básica para el diseño de las MSS y los problemas que pueden surgir durante su implementación, además de las posibles soluciones.

En esta primera parte se explica la teoría básica de síntesis de la MSS, posteriormente se hará énfasis en los problemas producidos por retardos de tiempo, tales como: glitches, generación de estados fly, oscilaciones. En una segunda parte se explicará el clock skew y algunas consideraciones de ruido y se exponen los resultados de una investigación al respecto.

ABSTRACT

The finite state machine (FSM) theory constitutes the backbone when you are dealing with the design of a complex digital system as microprocessors, video controllers, communication controllers, devices controllers, DSP, industrial controllers etc.

The method of synthesis for such circuits sets a procedure that must be followed in order to get the actual FSM working. Though there are several practical issues that have to be taken into account if you want to get a FSM working properly. We will go through a basic FSM theory review, after that, we will consider such problems as glitches, spikes, flight states, oscillations, clock skew and noise considerations.

We will work with typical solution methods and show results from a researching project that is going on.

1. INTRODUCCIÓN

Un sistema digital complejo está constituido por el llamado «data path» («camino de datos») y la unidad controladora. El primero corresponde a toda la circuitería combinatorial necesaria para el procesamiento de información digital y está constituido por funciones combinatorias tales como: multiplexores, decodificadores, codificadores, demultiplexores, contadores, flip-flops, gateways, osciladores de onda cuadrada, ALU's, etcétera. La unidad controladora corresponde a una MSS, la cual tiene como función coordinar todas las acciones que se deben verificar en el data path. Esta unidad «lee» la información suministrada por algunas de las unidades combinatorias y de acuerdo con la misma «toma decisiones» sobre lo que debe ocurrir en el resto del data path. Para hacer efectivo el control sobre otras unidades la unidad de control activa señales de salida. El diagrama de bloques de una MSS genérica se muestra en la figura 1. La MSS está constituida por un decodificador de entrada, elementos de memoria y un decodificador de salida. El decodificador de entrada corresponde a una función combinatoria que «genera» el llamado «estado siguiente», el cual depende del valor binario presente en los elementos de memoria (que

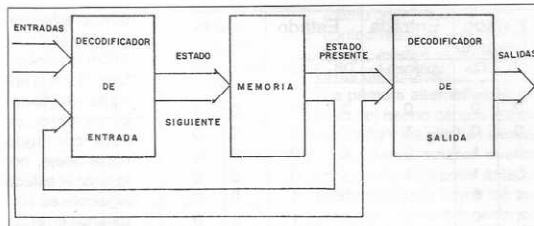


Figura 1. Diagrama de Bloques de una MSS.

corresponde al «estado presente») y de las entradas. Finalmente, las salidas pueden depender del estado presente, en cuyo caso se dice que la máquina es del tipo Moore, o del estado presente y de las entradas, recibiendo el nombre de máquina tipo mealy.

La principal característica de este sistema es que presenta «realimentación», sin la cual sólo sería una función combinatoria. Supóngase que se cuenta con dos elementos de memoria los cuales son flip-flops tipo «D», por lo tanto podrán representarse hasta cuatro estados diferentes codificados como 00, 01, 10 y 11 respectivamente. El decodificador de entrada es diseñado de forma que genere el estado siguiente de acuerdo con la combinación del estado presente y de las entradas.

Podrá verse el procedimiento de diseño de una MSS mediante un ejemplo, debe notarse que la notación para una señal «activa bajo» se hará mediante un guión seguido de la letra «L» (ejemplo: la señal «chip selection» es normalmente activa bajo por lo que se notará como CS-L). Supóngase que se necesita diseñar un típico contador binario con posibilidad de selección de conteo

arriba/abajo y generación de carry en el caso de un conteo hacia arriba y de borrow en un conteo hacia abajo. La secuencia de conteo será: 00, 01, 10, 11 o 11, 10, 01, 00 según el valor de la señal de entrada, la cual llamaremos «up» si su valor es «1» o «down» si su valor es «0». Ahora se establecerán algunas definiciones que nos ayudarán a entender la solución al problema, con la ayuda del diagrama de estados mostrado en la figura 2. Llamaremos estado presente al valor que pueda tener el contador en cualquiera de sus cuentas, en un momento determinado. El estado siguiente se define como aquel que deberá ocurrir después del estado presente, de acuerdo con el valor de la entrada (up o down-L). Finalmente las salidas «carry» y «borrow» están condicionadas a la entrada, por lo tanto ésta es una MSS tipo mealy.

La tabla de verdad que sintetiza el

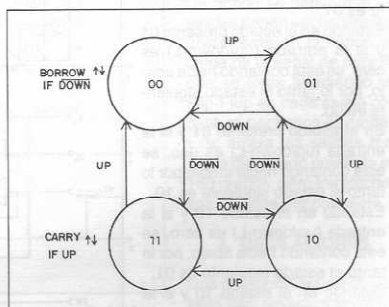


Figura 2. Diagrama de estados del contador síncrono up/down.

Estado presente		Entrada		Estado		salida Siguiente	
QB	QA	up/down-L	DB	DA	Carry	Borrow	
0	0	0	1	1	0	1	
0	0	1	0	1	0	0	
0	1	0	0	0	0	0	
0	1	1	1	0	0	0	
1	0	0	0	1	0	0	
1	0	1	1	1	0	0	
1	1	0	1	0	0	0	
1	1	1	0	0	1	0	

Figura 3. Tabla de verdad del contador síncrono up/down.

comportamiento de la máquina se observa en la figura 3.

Puede observarse el comportamiento de la máquina, tanto en el diagrama de estados como en la tabla de verdad. Su comportamiento puede describirse de la siguiente manera; supondremos que la máquina siempre comienza con el estado «00», observando el diagrama de estados de la figura 2 podemos decir que:

- Estando en el estado presente 00 y si la entrada (up/down-L) es cero, se está contando hacia abajo, por lo tanto el estado siguiente es 11, además se debe generar un «borrow».
- Estando en el estado presente 00 y si la entrada (up/down-L) es uno, se está contando hacia arriba, por lo tanto el estado siguiente es 01.
- Estando en el estado presente 01 y si la entrada (up/down-L) es cero, se está contando hacia abajo, por lo tanto el estado siguiente es 00.
- En el estado presente 01 y si la entrada (up/down-L) es uno, se está contando hacia arriba, por lo tanto el estado siguiente es 10.
- Estando en el estado 10 y si la entrada (up/down-L) es cero, se está contando hacia abajo, por lo tanto el estado siguiente es 01.
- Estando en el estado 10 y si la entrada (up/down-L) es uno, se está contando hacia arriba, por lo

guiente es 00, además se debe generar un «carry».

Todas estas relaciones mencionadas están sintetizadas en la tabla de verdad de la figura 3, donde se observa que los códigos que representan el estado presente son llamados Q_B , Q_A respectivamente, la función que genera el estado siguiente se representa como D_B , D_A y existen dos funciones de salidas para el «carry» y el «borrow».

Sintetizando el circuito se encuentran los mapas de Karnaugh y el circuito de la figura 4.

El circuito real podrá ser construido con circuitos integrados TTL, o me-

tanto el estado siguiente es 11.

- Estando en el estado 11 y si la entrada (up/down-L) es cero, se está contando hacia abajo, por lo tanto el estado siguiente es 10.
- Estando en el estado 11 y si la entrada (up/down-L) es uno, se está contando hacia arriba, por lo tanto el estado si-

guiente es 00. Cualquiera que sea el método de síntesis, siempre habrán problemas de índole práctico que no permiten el funcionamiento correcto del mismo. En lo sucesivo analizaremos dichos problemas, planteando soluciones prácticas que se sugieren en muchos textos de lógica digital.

2. LOS ESTADOS FLY Y LA GENERACION DE GLITCHES

Como estado fly, se conoce la condición en la cual la MSS transita por un estado intermedio, en su paso de un estado presente hacia un estado siguiente. Esta condición se presenta por la diferencia en los retardos de tiempo asociados a los elementos de memoria. Esta diferencia de tiempo es común aún para los Flip-Flops que han sido construidos en un mismo circuito integrado, encontrándose diferencias de tiempo de hasta 15 nsg. Esta condición se debe a la imposibilidad del fabricante de circuitos integrados para controlar los procesos de fabricación, de forma que los retardos de tiempo sean despreciables (la palabra «despreciable» se usa aquí en forma relativista, ya que se trabaja en el rango de los nano-

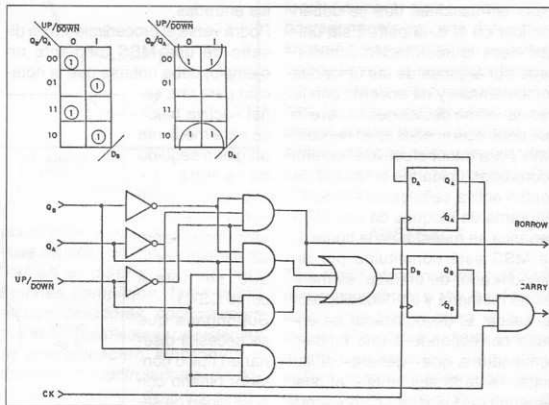


Figura 4. Síntesis de la MSS de la figura 3.

segundos, sin embargo, una diferencia de una decena de nanosegundos puede provocar un funcionamiento erróneo de un circuito).

Analizando el tránsito de la MSS del estado 01 al 10 (recordar que el estado presente se simboliza como $Q_B Q_A$) se pueden obtener dos casos:

1. Suponiendo que el Flip-Flop «A» es más veloz que el Flip-Flop «B», exactamente por 10 nsg y tomando como referencia el flanco de subida de la señal del reloj (ck) podrá verse que el tránsito del estado 01 al 10 provoca un estado intermedio 00, justamente por un tiempo igual a la diferencia en los retardos de los Flip-Flops (10 nsg). Este hecho no tendría importancia si no es por que justamente en el estado 00 se produce la «activación» de la salida «borrow». Es claro que esta salida **NO** debe activarse a menos que se esté en el estado presente 00. Esta activación no premeditada de una salida, bajo las condiciones anteriores, es conocida como «glitch».

Un diagrama de tiempos donde se ve la manera como se genera un glitch se muestra en la figura 5.

2. Se supondrá ahora que el Flip-Flop más veloz es el «B». Bajo estas nuevas condiciones y manteniendo la diferencia de tiempos entre elementos de memoria de 10 nsg, el tránsito desde el estado 01 al 10 provoca un estado fly 11. De nuevo el hecho relevante es que durante los 10 nsg de vida del estado fly 11, se activa la señal de salida «carry», lo que no es correcto. El diagrama de tiempos de la figura 6 ilustra este hecho.

Sin duda, un pulso provocado

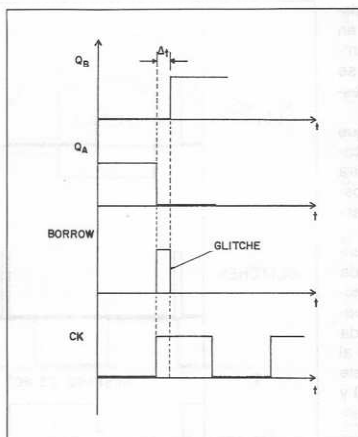


Figura 5. Diagrama de tiempos correspondiente a la generación de glitches por el estado fly 00.

por este fenómeno activará a otro contador que se encuentre en cascada con el primero, haciendo que las cuentas no sean secuenciales. El lector podrá comprobar lo que ocurre si el cambio va de 10 a 01

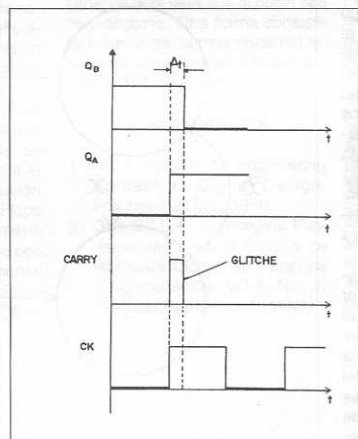


Figura 6. Diagrama de tiempos correspondiente a la generación de glitches por el estado fly 11.

(como ocurriría en el caso de conteo hacia abajo). Sin embargo, existen varias alternativas de solución.

La primera alternativa se deriva del hecho de que los estados fly y los glitches asociados a estos, ocurren cuando hay un tránsito entre estados codificados de forma tal que cambian varios bits, esto es, en el ejemplo, se va del estado 01 al 10, nótese que el primer bit debe conmutar de 0 a un 1 y que el segundo debe hacerlo de 1 a 0. Si la codificación de los estados es tal que «uno y solamente uno de los bits cambia», desaparecerían los estados fly y con ellos los glitches. Por ejemplo los estados podrán codificarse como: 00 01 11 y 10. Sin embargo, esta posibilidad no puede aplicarse en este caso

debido al estricto orden de las cuentas en un contador.

La segunda alternativa para la prevención de los glitches es hacer un filtrado desplazando en el tiempo la ocurrencia de las salidas.

Obsérvese la figura 7 de donde podremos expresar que:

- Ocurrirán glitches si las salidas se habilitan en $f(t_i)$.
- No ocurrirán glitches si las salidas se habilitan en $f(t_i + p_i)$.

De donde se deriva que las salidas deben ser habilitadas únicamente en el flanco de caída de la señal de reloj (lo que se logra añadiendo un Flip-Flop en cada salida con captura por flanco de caída). Esto atrasa las salidas respecto del estado presente en 180 grados, pero los glitches son filtrados por completo.

3. OSCILACIONES

Una oscilación ocurre en una MSS bajo las siguientes condiciones: los elemen

tos de memoria son latches y existe el tránsito entre dos estados en forma consecutiva donde uno o ambos bits cambian. El cambio se debe a una entrada según se muestra en la figura 8.

Del diagrama de estados se ve que esta máquina no tiene una aplicación real, sin embargo, demuestra la posibilidad de ocurrencia de oscilaciones bajo las condiciones establecidas.

Supóngase que la máquina comienza en el estado 11, y que la entrada x es igual a 0. Bajo estas condiciones la máquina quedará bloqueada en el estado 11. Si la entrada cambia a 1, la máquina avanza al estado 01. Supóngase que en este estado la entrada x es igual a 0 y permanece así por largo tiempo, entonces la máquina reciclará por los dos estados 00 y 01, mientras la entrada continúe en 0. Es aquí donde las oscilaciones ocurren si el ancho del pulso del reloj

es suficientemente grande (de nuevo esta afirmación es relativista y significa que el ancho del pulso de la señal de reloj supera varias veces el período de las oscilaciones). Observando a Q_1 , se ve que este bit debe conmutar de 1 a 0 y luego de 0 a 1, en otras palabras el decodificador de entrada se «transforma en un inversor». Teniendo en cuenta los retardos propios del decodificador de entrada y de los elementos de memoria, se observa que el período de la oscilación es la suma de dichos retardos.

La oscilación ocurre debido al efecto de la realimentación en el inversor equivalente según se muestra en la figura 9.

Supóngase que el inversor equivalente tiene en su entrada un valor de 0, luego de un tiempo igual al retardo en el decodificador de entrada, este cero forzará un 1 en la salida. Este 1 viajará a través del elemento de memoria que ahora es «transparente» debido al pulso de reloj. Luego del retardo en el elemento de memoria el 1 entrará de nuevo en el inversor equivalente

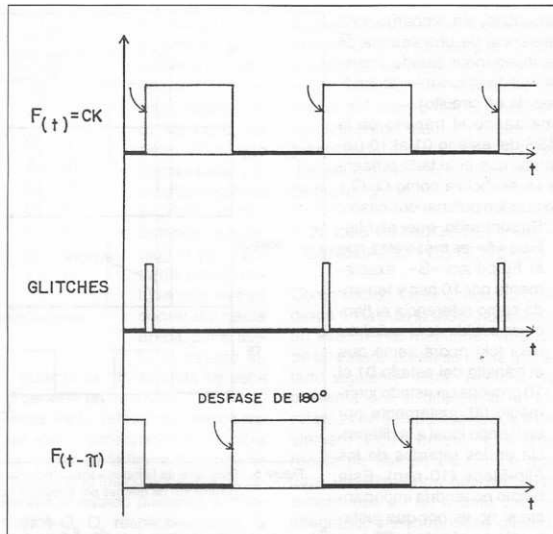


Figura 7. Diagramas de tiempo del método de filtrado de glitches.

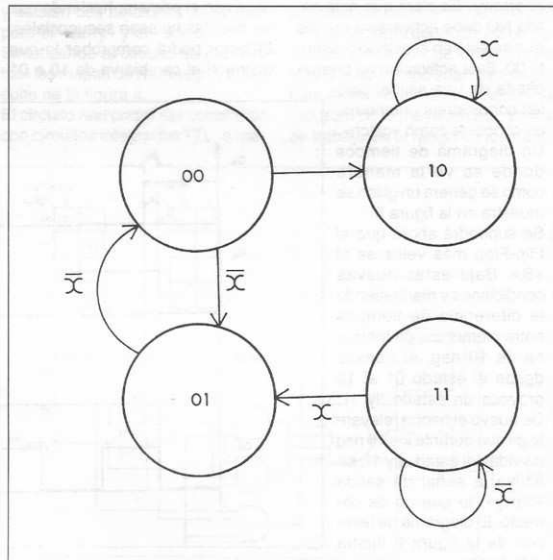


Figura 8. Diagrama de estados para el ejemplo de oscilaciones.

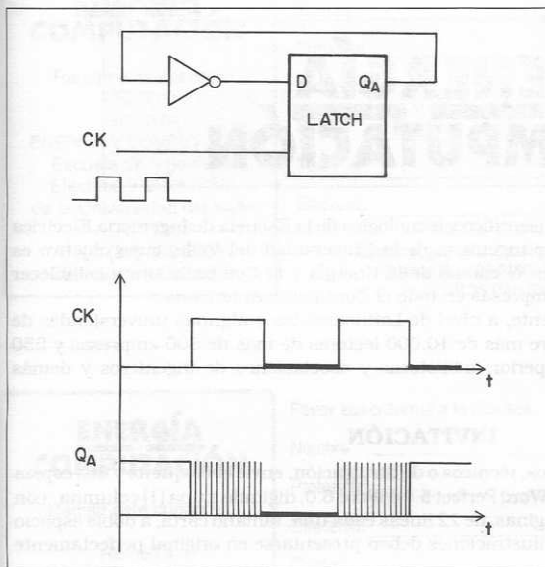


Figura 9. Circuito equivalente y diagrama de tiempo de la MSS de la figura 8.

para forzar un 0. Este ciclo de oscilación se repite hasta que el pulso de reloj caiga a cero. El mismo fenómeno ocurrirá en cada ciclo de trabajo de la señal de reloj.

Las condiciones para que ocurran oscilaciones son muy particulares, sin embargo, debe tenerse en cuenta este fenómeno durante el diseño de una MSS. La solución más sencilla es utilizar Flip-Flops en lugar de latches en la síntesis de los circuitos. Existe un truco con el cual se evitarían las oscilaciones y consiste en «simular un Flip-Flop» limitando el ciclo de trabajo de la

señal de reloj para que el pulso sea muy angosto. Otra forma consiste en incrementar la frecuencia del reloj hasta que el ciclo en alto sea suficientemente angosto.

4. BIBLIOGRAFIA

1. FLETCHER, W. An Engineering Approach to Digital Design. Prentice-Hall Inc., 1980.
2. GOMEZ, L.A. Conceptos Fundamentales en el Diseño de Hardware Digital. En : Energía y Computación. Vol.4, No. 2, (Segundo semestre de 1995).

3. GRAY, P. y SEARLE, C. Electronic principles Physics, Models and Circuits. John Wiley & Sons Inc, 1969.
4. NAGLE, H.T, CARROLL, B.D. y IRWIN, J.D. An Introduction to Computer Logic. Prentice-Hall Inc., 1975.
5. NATIONAL SEMICONDUCTOR CORP. LS/S/TTL Logic Data-Book
6. TEXAS INSTRUMENTS ELECTRONICS SERIES. Designing with TTL Integrated Circuits.
7. TINDER, R. Digital Engineering Design, Prentice-Hall Inc., 1991

AUTORES

Luis Alberto Gómez Arizabaleta

Ingeniero Electricista, Universidad del Valle 1991. Magíster de la Universidad del Valle 1996, Profesor Auxiliar del Departamento de Electricidad. Director del Plan de estudios de Ingeniería Electrónica. Actualmente realiza investigaciones en las áreas de sistemas digitales, inteligencia artificial y arquitectura de computadores.



Diego Fernando Duque Betancourth

Ingeniero Electricista, Universidad del Valle 1996. Estudiante Programa de Magíster en Automática. Actualmente realiza investigaciones en las áreas de sistemas digitales e inteligencia artificial.