Lógica asincrónica: grandes dispositivos CMOS sin arquitectura de reloj

CMOS (Complementary Metal Oxide Semiconductor) es en la actualidad el proceso más maduro de la industria electrónica. Sin embargo, el nivel de complejidad de los dispositivos y sistemas en pastillas de silicio (SoCs), elaborados en éste, aumenta de tal manera que los ingenieros se ven obligados a afrontar nuevas dificultades.

Por Juergen Pinstake, ExMark

Algunos sistemas que utilizan múltiples elementos de propiedad intelectual (IP), encuentran limitaciones por retrasos en sus señales de reloj, consumo de potencia y limitaciones propias de las herramientas de diseño y verificación. Una de las causas viene del hecho de que el dispositivo en su conjunto está concebido como un único circuito sincrónico. Muchos de estos problemas se pueden resolver con el uso de la lógica asincrónica.

¿Qué es la lógica asincrónica?

La lógica asincrónica no es algo nuevo: fue creada hace unos 20 años, pero salvo algunas excepciones, sigue estando a nivel de investigación. Hasta ahora, los ingenieros no han osado introducirla en sus diseños, como consecuencia de las escasez de herramientas a su disposición. Las herramientas EDA (automatización de diseño electrónico, del inglés Electronic Design Automation) están concebidas para diseñar circuitos sincrónicos, incorporando numerosas funciones en una misma pastilla. Si todas las señales generadas en una pastilla cambian al mismo tiempo, la verificación del sistema se simplifica: "sólamente" hay que comprobar que los retrasos en la salida de los registros, vía la lógica combinatoria, sean menores que el intervalo de tiempo entre dos señales de reloj. Esto se puede considerar como un diseño puro, en comparación con un sistema basado en lógica asincrónica, que requiere un entendimiento más detallado. No obstante, la lógica sincrónica no excluye la asincrónica: una implementación sincrónica es un caso especial dentro del diseño de circuitos asincrónicos.

Tomemos un ejemplo bien conocido: un grupo de soldados marcando una marcha militar en un puente. Si marchan en coordinación, existe peligro de vibración de la estructura, pero si lo hacen sin coordinación alguna, este riesgo desaparece. Esta analogía puede explicar los diferentes comportamientos de ruidos e interferencias electromagnéticas presentes en circuitos electrónicos. En el caso de un sistema sincrónico, aparece un pico de corriente cada vez que el reloj activa los registros, mientras que en el caso de una implementación asincrónica, el valor de este pico puede reducirse en 10dB. (Tabla 3).

Problemas de los sistemas sincrónicos

Veamos cómo funciona un sistema sincrónico y la manera de transformarlo en un diseño asincrónico. En una configuración sincrónica todas las entradas y las salidas están sincronizadas con el reloj, y toda la pastilla se comporta como una máquina de estados finitos (figura 1a).

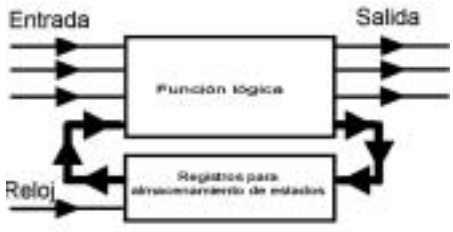

Figura 1a: Máquina de estados finitos (finite state machine).

Una máquina de estados finitos consta de dos bloques: uno para la lógica Bolean sin retroalimentación, y otro para los registros que definen el estado actual cuando se produce la señal de reloj. Cada estado actual está retro-alimentado como parte integrante de las entradas lógicas. Cuando la señal de reloj se establece, el estado actual queda bloqueado. Las señales de las salidas lógicas tienen que estabilizarse antes de la llegada de una nueva señal de reloj. Esto implica que el período del reloj tiene que ser mayor que el tiempo necesario para recorrer el camino más largo (camino crítico) dentro del bloque lógico. Por otro lado, la frecuencia del reloj tiene que ser lo suficientemente elevada para captar todos los cambios de entrada (figura 1b).

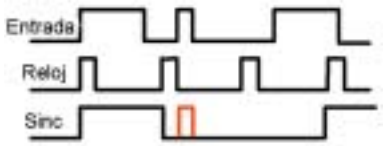

Figura 1b: Sincronización de una entrada con el reloj y posibles problemas.

Todas las señales del circuito tienen que producirse en sincronización con el reloj, estableciendo una clara relación para diseño, verificación y pruebas del circuito. Si todas las puertas lógica se encuentren agrupadas, no habría problema, pero en sistemas reales (figura 1c), los registros están distribuidos en toda la pastilla, a veces constando de millones de puertas.

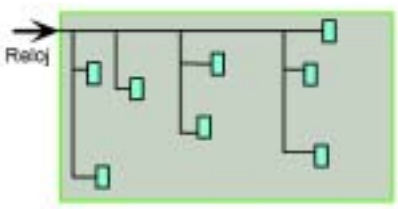

Figura 1c: Distribución del reloj en una pastilla de silicio.

Puesto que el camino crítico más largo es la red de reloj, es necesario tener numerosas interfaces adicionales para mantener la precisión de éste, ocasionando al mismo tiempo retrasos y consumo de potencia. A las altas frecuencias de reloj, las líneas de transmisión de éste se comportan como antenas, generando interferencias electro-magnéticas. Desafortunadamente, ambos efectos se presentan también cuando no hay actividad en las entradas o las salidas, ya que el reloj sigue emitiendo una señal en cada uno de sus ciclos. Estos fenómenos no eran tan importantes en previas generaciones de circuitos integrados, puesto que los conductores metálicos eran más anchos, y ocasionaban menos retrasos en las señales que los producidos a nivel de puertas durante el procedimiento lógico. Pero las nuevas tecnologías de silicio utilizan geometrías muy pequeñas y la velocidad de las puertas ha aumentado. Esto se combina con un aumento drástico de la densidad de los circuitos. Además las ramas de reloj han alcanzado un nivel de complejidad y de longitud tal, que la sincronización de todos los registros en una pastilla de silicio se convierte en una tarea cada vez más difícil.

Transformación de un sistema sincrónico en un sistema asincrónico

En un diseño asincrónico no hay ramas de reloj conectando todos los registros, aunque las funciones tienen que estar coordinadas. Una manera de describirlo es subdividiendo los sistemas asincrónicos en tres partes con necesidades diferentes: pequeñas, medianas y grandes. A continuación, presentamos algunas características de estos diseños.

-Transmisión de datos vía el nivel de la señal: 0 y 1 como lógica convencional. La transición de un nivel al otro puede ocasionar problemas dependientes de la velocidad del cambio. Un estado intermedio define un tercer nivel: "indeterminado".

-Transmisión de datos usando una transmisión de la señal: La señal sólo es aceptada tras la transición. Niveles y velocidad de cambio no tienen importancia y sólo ocasionan retrasos.

-Transmisión de datos en paquete: Los buses de datos o los grupos de señales se acompañan de una línea de reloj adicional.

-Transmisión de datos con líneas dual: Las señales "alta" y "baja" fluyen por dos líneas separadas, como señales activas. Sólo la señal "nivel alto" es tenida en consideración.

-Sin dependencia por la velocidad: El sistema calcula de manera independiente a las variaciones de velocidad de las puertas. No tienen efecto los cambios de alimentación o temperatura. Sensible a los retrasos.

-Sin dependencia por los retrasos: Sólo el rendimiento cambia. El funcionamiento del sistema es totalmente independiente de los retrasos.

Si bien varias señales recorren la pastilla, el álgebra Bolean sólo describe la operación lógica, sin tener en cuenta los retrasos existentes a nivel de las conexiones o de las puertas. La lógica tampoco se ve afectada por las variaciones de alimentación. En un circuito digital, las señales sólo pueden tener valores binarios, alto o bajo. Una transición lenta ocasionará un retraso en el cambio de estado de la puerta. Por esta razón, resultaría más sencillo referirse al cambio de nivel, más que a un nivel concreto. Una alternativa es transportar las señales altas y bajas por líneas diferentes. (figura 2).

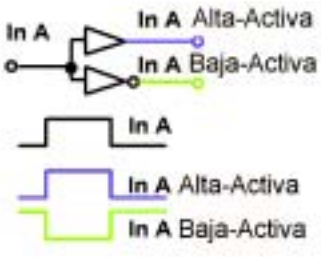

Figura 2: Separación de una señal en dos líneas activas

Esta configuración aumentará el número de líneas en la pastilla, pero sólo se transportarán señales activas. Además, en un mismo instante de tiempo, únicamente estará activa una de las dos señales provenientes de una misma pareja de líneas. En el caso en que haya que organizar una gran cantidad de líneas paralelas a través del circuito, se pueden aprovechar de estas líneas para transmitir la señal del reloj correspondiente, guardando así el vínculo entre reloj y señal. El objetivo es desarrollar un sistema independiente de la velocidad de reloj y no sensible a los retrasos.

La mayor parte de los diseños asincrónicos utiliza señales por línea dual, con codificación sensible al nivel de la señal. En estos diseños la funcionalidad no se ve afectada por variaciones en la alimentación o la temperatura, sólo el rendimiento se ve afectado.

Pequeños sistemas o funciones asincrónicos

Las funciones de control son un buen ejemplo a la hora de implementar la lógica asincrónica, ya que su complejidad es limitada y su comportamiento se puede describir en su totalidad. En la figura 1 se ilustra una máquina de estados. Si el diseño sólo permite un cambio de valor de entrada a la vez (por ejemplo, un teclado con codificación de prioridad o codificación Gray), y si todas las señales se han estabilizado antes del próximo cambio de reloj, se puede utilizar el método de Huffman FSM (Frequency Shift Mode) para la transmisión de señales sin reloj, tomando la señal activa en cada momento el papel de este último (figura 3).

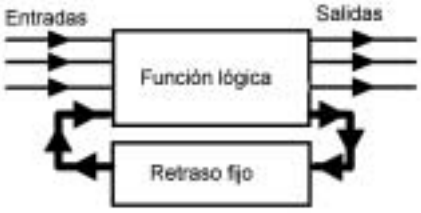

Figura 3: Máquina de estados finitos asincrónica Huffman (sin reloj)

Codificación por línea dual

Como ya se explicó en el párrafo anterior, si se separan las señales alta y baja en dos líneas independientes, la señal activa hace las veces de reloj en ese instante (figura 2). De este modo sólo el estado "alto" tiene importancia, y un estado "bajo" significa "inactivo". Así cada señal transporta su propio reloj. En consecuencia si no hay cambios de entrada, la puertas no trabajan y sólo circula una corriente quiescente. Las puertas funcionan de manera diferente ya que sólo una entrada alta puede cambiar la salida. Si una conexión está rota, la puerta correspondiente no puede cambiar la salida. Un ejemplo podría ser una votación con dos opciones, si o no. No se permite dar ambas respuestas a la vez. No obstante, este tipo de lógica no sólamente se puede utilizar a nivel de una puerta, y la compañía Theseus ha patentado ya una implementación de lógica asincrónica disponible para su inclusión en el diseño de microcontroladores o bloques de propiedad intelectual.

Llamada NCL (del inglés Null Convention Logic), este bloque lógico utiliza una implementación especial donde todas las señales activas son estimuladas para generar todas las salidas. De esta forma, un ciclo "Null" se ejecuta para reiniciar todas las retroalimentaciones, seguido por un nuevo ciclo activo.

El elemento C de Muller (figura 4) ilustra la función de dicho bloque asincrónico. El bucle de retroalimentación adicional produce un comportamiento secuencial especial.

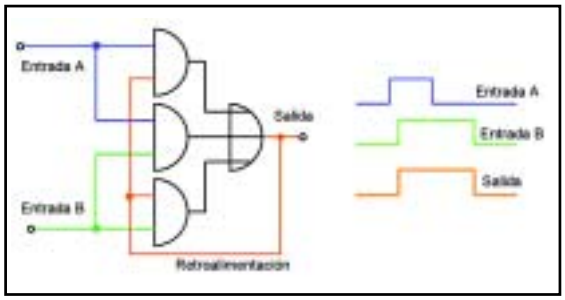

Figura 4: Elemento C de Muller. Función AND asincrónica

Con las dos entradas inactivas, la salida permanece inactiva. La primera entrada (la más rápida) no cambia la salida, sólo se produce el cambio cuando la segunda entrada se activa. Luego si una de las entradas se inactiva, la salida permanecerá activa vía la retroalimentación. Cuando la segunda entrada se inactiva, la salida pasa de nuevo al estado inactivo (bajo). Los sistemas de complejidad media tienen buses además de sus estructuras de control.

Transmisión de datos utilizando una señal "ready" (listo)

Los diversos bloques funcionales de un circuito pueden transmitir datos únicamente cuando ciertas señales adicionales indican que los datos están estabilizados y que se pueden tomar en consideración (figura 5). Esto se puede alcanzar añadiendo una señal "inicio" que provoca un retraso fijo de amplitud mayor que el camino crítico en el bloque de lógica correspondiente.

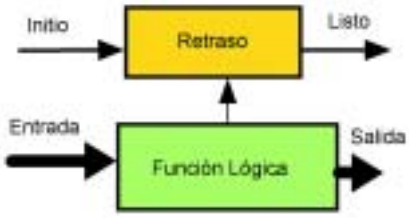

Figura 5: El uso de retrasos para la sincronización.

La señal "listo" indicará que las señales de salida se han estabilizado. En esta configuración, ninguna de las puertas funciona a la máxima velocidad, puesto que el factor de seguridad en la cadena paralela de retrasos definirá el rendimiento máximo.

Transmisión de datos utilizando una señal "handshake" (intercambio)

El rendimiento del circuito sería mayor si los bloques vecinos pudieran cooperar. En una versión pasiva, el remitente indica que los datos están disponibles. El siguiente bloque en la cadena recibe los datos e informa al remitente, vía la señal "listo", de que ya se pueden preparar nuevos datos para iniciar un nuevo ciclo (figura 6). Los datos y el protocolo de intercambio se propagan en la misma dirección.

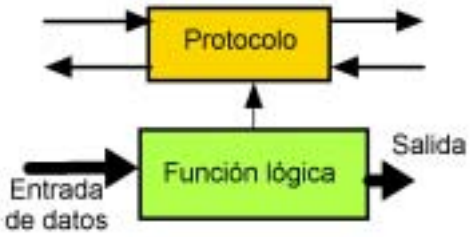

Figura 6: Transmisión de datos con un protocolo de intercambio, activa o pasiva.

La versión activa funciona de manera diferente: el remitente indica que está esperando los datos y da la señal de entrega. En este escenario, los datos y la solicitud se propagan en sentidos opuestos.

Micro-conductos

Si combinamos el cálculo de datos utilizando el retraso paralelo y la secuencia solicitud/listo, alcanzamos una manera de comunicar muy eficaz entre los bloques de lógica asincrónica (figura 7). Un tipo de comunicación muy parecido fue utilizado en el desarrollo de varios procesadores Amulet, basados en la arquitectura ARM. Para coordinar varios retrasos en una pastilla se implementan FIFOs (primero-dentro, primero-fuera del inglés first-in, first-out) y LIFOs (último-dentro, primero-fuera del inglés last-in, first-out), para asegurar que los datos que llegan de manera asincrónica estén disponibles en el momento adecuado para su uso.

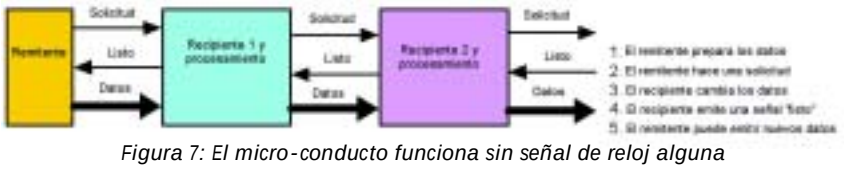

Figura 7: El micro-conducto funciona sin señal de reloj alguna

Hardware y software

El problema del procesamiento de datos asincrónicos no sólo existe en hardware. Problemas similares tendrán lugar en sistemas que soportan procesado en paralelo en procesadores múltiples o en una misma pastilla, si los datos de entrada no están sincronizados. En ambos casos, la señal de datos tiene que permanecer estable hasta una próxima tarea. Una expresión de moda, "hardware-software co-diseño", aparece a menudo cuando se trata de la simulación de sistemas completos. Puesto que la simulación en C es independiente de la implementación, no define cómo las diferentes funciones se comparten entre hardware (puertas) y software. La decisión se toma más tarde, teniendo en consideración requisitos tales como la velocidad de ejecución, los costes de hardware, el vólumen de producción y el procesador elegido.

Implementación

En tal sistema el software será optimizado por el compilador C y el hardware por la etapa de síntesis lógica. Aunque el sistema deba reaccionar con eventos asincrónicos, esto no excluye el uso de herramientas estándares. La tabla 2 nos ofrece una comparativa entre el número de transistores utilizados en varios diseños implementados con lógica sincrónica y con lógica asincrónica en la tecnología NCL. El número de transistores varia en estos ejemplos entre 0.7 y 2.1.

Modulo	sincrónico	NCL	Ratio
AddrConv	41	62	1.5
X2VHD	395	826	2.1
Decoder	1010	1804	1.8
32x16FIFO	1691	1123	0.7

Este ratio no toma en consideración cuánto espacio, complejidad o número de transistores se han ahorrado en la pastilla usando el NCL asincrónico. Es muy importante compensar la adición de puertas con tiempos de desarrollo más cortos. Una comparación muy interesante es la implementación llevada a cabo por la empresa Theseus, con el NCL08GP32, un núcleo de microcontrolador compatible con la serie Motorola HC05/HC08/STAR08.

Detalles del núcleo de procesador Theseus NCL08GP32 realizado con lógica asincrónica NCL:

- Instrucciones compatibles con el Motorola HC05, HC08 y la serie STAR08
- El núcleo funciona sin reloj.
- 40% menos de consumo de potencia en comparación con una implementación sincrónica.
- Sin retraso en los modos de parada o de espera.
- Alrededor de 10 dB menos de interferencias.
- Operativo entre 1,6V y 3.6V.

Futuro papel de la lógica asincrónica

La lógica asincrónica tiene sentido en el diseño de productos portátiles, donde el ahorro de energía es una ventaja clara sobre los diseños sincrónicos, cuyo reloj emite una señal para toda la pastilla. El cambio a un diseño asincrónico implica también que diferentes bloques de propiedad intelectual no sufren retrasos de intercomunicación, ya que su funcionamiento es independiente del reloj. Productos, tales como la implementación de Theseus, llegan al mercado, basados en herramientas de diseño estándares.

Para mas información, véase:

ExMark: www.exemark.com
NCL logic: www.theseus.com
Synthesis: www.synopsys.com
El procesador AMULET: www.cs.man.ac.uk/amulet/
Lógica asincrónica: www.cs.man.ac.uk/async/