

22 CIRCUITOS INTEGRADOS PROGRAMABLES: MATRICES DE CELDAS CONFIGURABLES

- 22.1. Resumen de la configuración y características de los CPLDs
 22.2. Los biestables como componentes programables: SRAM
 22.3. Matrices de celdas programables: FPGAs

Los circuitos integrados programables constituyen una de las opciones de «construir» la electrónica digital en tamaño «micro», es decir, de insertar nuestro diseño digital en un único circuito integrado; la otra es fabricar específicamente dicho diseño.

Dos son los grandes tipos de dispositivos programables, que se diferencian según su «granularidad» (según el tamaño de su celda básica) y por la forma de configurar las funciones booleanas: los de granularidad gruesa y configuración PAL (funciones en forma de suma de términos producto) y los de granularidad fina y configuración LUT (tabla funcional a través de multiplexores): CPLDs y FPGAs. Este capítulo incluye un comprensivo repaso de los CPLDs que ya fueron objeto de estudio en el capítulo 13 (segundo volumen) y la presentación en detalle de las FPGAs.

Frente a la forma física de programación, habitual en los CPLDs (fusibles y transistores NMOS de doble puerta), las FPGAs utilizan una programación más «suave», por almacenamiento de valores sobre biestables: programación SRAM, cuyo fichero debe ser nominado «mapa de valores» (en lugar del clásico «mapa de fusibles»).

Una FPGA contiene bloques lógicos (formados por varias celdas lógicas elementales), bloques de entrada/salida y conexiones entre ellos; dichos tres componentes (cada uno de ellos) son de una gran potencia, en cuanto a posibilidades de programación.

Cada celda lógica permite su utilización como función booleana, como registro de desplazamiento, como pequeña memoria RAM o ROM, como multiplexor rápido,...; además, incluye recursos específicos para operaciones aritméticas: sumadores, restadores, multiplicadores y contadores de muy alta velocidad.

Los adaptadores de entrada/salida son celdas periféricas que ofrecen su programación como entradas, como salidas o como terminales bidireccionales, con sincronización o no de la entrada y la salida, con capacidad de alta impedancia y con la posibilidad de seleccionar los niveles de las tensiones lógicas, las pendientes de las señales (slew rate), ...

Una tupida malla de conexiones programables, de diversos tipos de longitud (cortas, dobles y largas), a través de los canales horizontales y verticales que hay entre los bloques, ofrece una amplísima conectividad entre las diversas celdas, optimizando los retrasos que introducen las líneas de conexión.

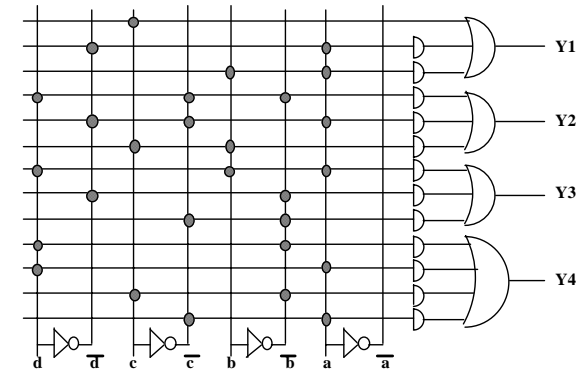
A todo ello se une la disponibilidad de bloques específicos de memoria RAM, de multiplicadores e, incluso, de microprocesadores, todo dentro de una misma FPGA. En suma, cualquier sistema digital, por grande y complejo que sea, puede ser «insertado» en una FPGA, en forma eficiente.

22.1. Resumen de la configuración y características de los CPLDs

Estructura PAL + biestables

Recordemos, una vez más, que la finalidad de la microelectrónica es configurar un circuito digital específico en el interior de un circuito integrado (es decir, «insertar nuestro diseño en un integrado») y que una manera de conseguir este objetivo consiste en programar el circuito específico en un circuito integrado programable.

La configuración reticular de varias funciones booleanas, expresadas en forma de suma de productos, recibe el nombre de PAL (product adding layers –estratos o módulos sumadores de productos–).

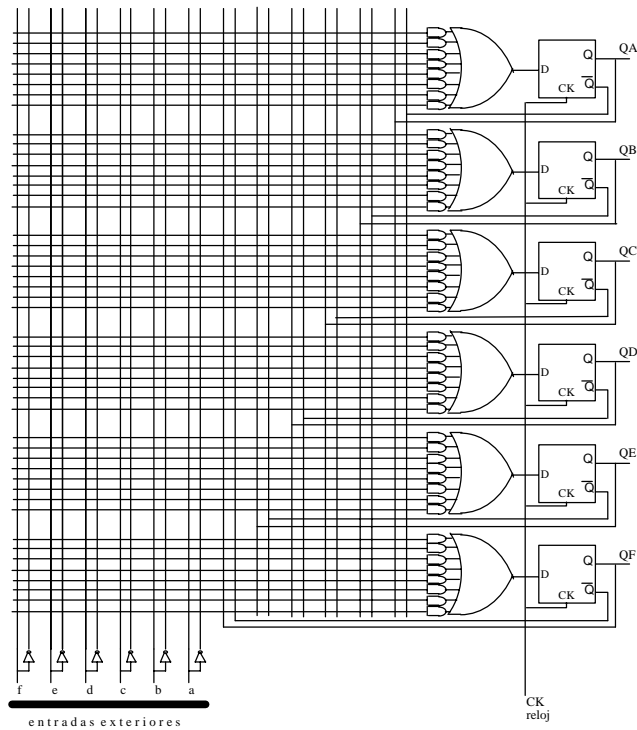


$$Y1 = c + \bar{d}.a + b.a \quad Y2 = d.\bar{c}.\bar{b} + \bar{d}.\bar{c}.a + c.b$$

$$Y3 = d.b.a + \bar{d}.\bar{b} + \bar{c}.\bar{b} \quad Y4 = d.\bar{b} + d.a + c.\bar{b} + \bar{c}.a$$

Cada función (suma de productos $\sum p$) ocupa un módulo separado y dispone de sus propios términos producto; las conexiones de las variables se ajustan a la retícula formada por las entradas y sus negadas sobre las «líneas de entrada» de las puertas "y".

Para poder programar sistemas secuenciales se requiere, además de funciones booleanas, biestables como «memoria» para las variables de estado. Bata, para ello, añadir a la configuración PAL la capacidad de memoria que proporcionan los biestables: incluir un biestable en cada una de las salidas de un bloque PAL. De esta forma tendremos bloques con n módulos, cada uno de ellos con un biestable cuya entrada D es activada por una función en forma de suma de productos, cuyas variables serán las propias variables de entrada y, también, las salidas de los biestables (realimentación). Tales bloques reciben la denominación de PLS (secuenciadores lógicos programables) o, también, la de PLD (dispositivos lógicos programables).



Componentes programables

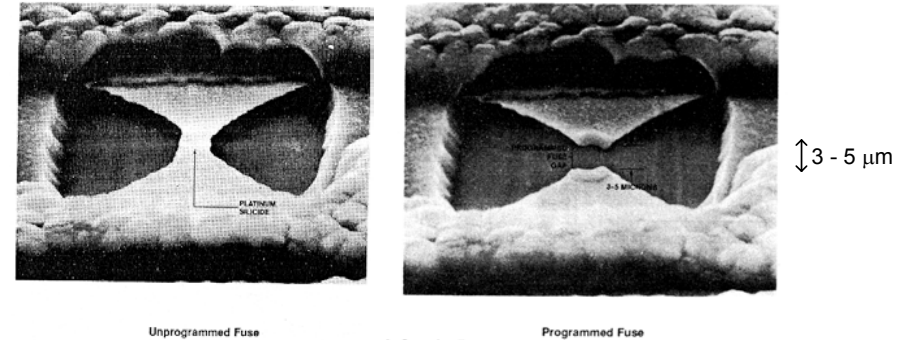
Recordemos las formas de conseguir que una puerta lógica sea programable: ¿cuáles son los mecanismos físicos que permiten tal programación? Básicamente hay dos formas de dotar de «capacidad de programación» a una puerta lógica, dos tipos de componentes circuitales que permiten la conexión-desconexión física entre líneas del circuito:

- > fusibles
- > y transistores MOS de doble puerta

y, dentro de los transistores de doble puerta, se dispone de tres tipos diferentes:

- ◊ EPROM (borrables por luz ultravioleta)
- ◊ EEPROM (borrables eléctricamente)
- ◊ FLASH (con borrado eléctrico global y mayor densidad de integración).

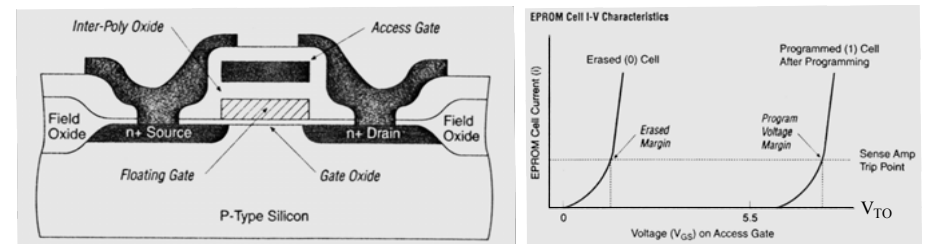
Un fusible se construye mediante un estrechamiento en un conductor de un material apropiado, de forma que pueda «fundirse» por efecto térmico mediante el paso de una intensidad relativamente elevada; suele tener un tamaño aproximado de 5 x 5 µm y se utilizan aleaciones de platino-silicio, titanio-tungsteno o níquel-cromo (nicrom).



Los fusibles se utilizan principalmente en tecnologías bipolares de alta velocidad; se programan mediante «fusión térmica», a través de pulsos de intensidad alta, con una fuerte pendiente di/dt para generar tensiones puntuales apreciables (por efecto inductivo): tensiones típicas de 10,5 V aplicadas en pulsos repetitivos de anchura entre 10 y 50 µs, dando lugar a intensidades de 0,1 - 0,5 A con tiempos de subida inferiores a 100 ns.

Los fusibles son de tipo destructivo en cuanto a su programación; por ello, no son «reprogramables», es decir, una vez programados no pueden «borrarse» y volver a la situación inicial ya que la conexión ha sido «cortada» y no puede repararse. Este tipo de dispositivos se denomina **OTP** (*one time programmable*), a diferencia de los dispositivos **EPROM** que son reprogramables.

Los transistores MOS de doble puerta (MOS EPROM) se programan mediante almacenamiento de carga en la puerta interior; dicha carga modifica la tensión umbral del transistor MOS, situándola por encima de la tensión de alimentación: la tensión umbral V_{TO} pasa del entorno de 1 V a unos 8 V y el transistor no entra nunca en conducción.



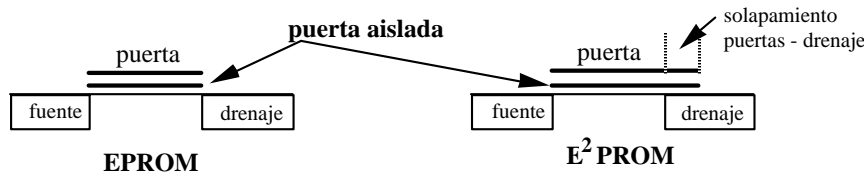
Hay tres tipos de transistores MOS programables:

- i) EPROM, borrables mediante luz ultravioleta (para descargar la puerta aislada).

Su programación se hace a través de una fuerte corriente de canal y de una tensión relativamente alta en la puerta externa (pulsos de tensión de unos 12 V sobre drenaje y sobre puerta). El alto coste que supone el encapsulado con ventana hace que estos dispositivos EPROM se ofrezcan también en versión **OTP** (sin ventana).

- ii) E²PROM, borrables eléctricamente, cuya programación y borrado se hace mediante tensión entre la puerta externa y el drenaje.

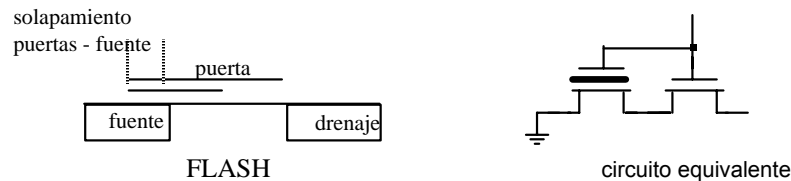
Un pequeño solapamiento de las dos puertas con el drenaje permite que el proceso de carga de la puerta aislada sea reversible: se carga y se descarga desde el drenaje por atracción (tensión de puerta positiva) o repulsión de electrones (tensión negativa).



El borrado es individual (se efectúa sobre un solo transistor), pero presenta un problema de «sobreborrado»: cuando la tensión de la puerta exterior es negativa la puerta aislada suele cargarse positivamente y se genera un canal permanente que pone en conducción al transistor. Por ello, es necesario utilizar dos transistores en serie: el de doble puerta para conectar y desconectar al segundo, que es un transistor NMOS normal.

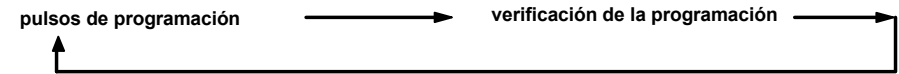
- iii) FLASH, borrables eléctricamente, cuya programación se hace por corriente de canal y el borrado por tensión entre puerta y fuente.

Este tercer tipo de transistores proviene de agrupar los dos transistores de la celda E²PROM en uno solo: las dos puertas se solapan con la fuente y la puerta aislada sólo ocupa la mitad de la longitud del canal.



El proceso de carga de la puerta aislada es del tipo EPROM, con una intensidad de canal alta que deriva electrones hacia dicha puerta y la descarga se produce según el tipo E²PROM, por pulsos de tensión positiva sobre la fuente, que atrae a los electrones de la puerta aislada. El borrado no puede realizarse individualmente sino que afecta a todos los transistores del bloque (la fuente es común a todos los transistores NMOS) y no hay riesgo de «sobreborrado» (aunque la puerta aislada se cargue positivamente, no puede poner en conducción al transistor, ya que sólo actúa sobre la mitad del mismo).

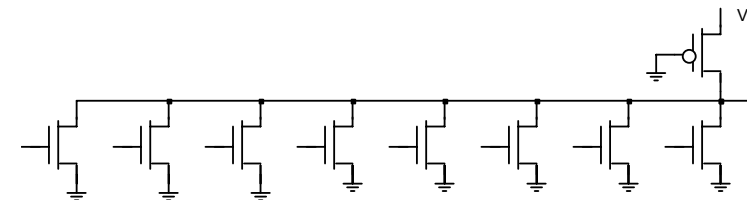
Los dispositivos programables (fusibles y EPROM) se diferencian en el proceso físico que determina su programación: ruptura por efecto térmico y almacenamiento de carga, respectivamente; sin embargo, el proceso de programación de ellos es análogo: se aplica una secuencia de pulsos de tensión relativamente elevada que se repite cíclicamente hasta estar seguros de que la correspondiente conexión ha quedado programada.



Configuración pseudoNMOS

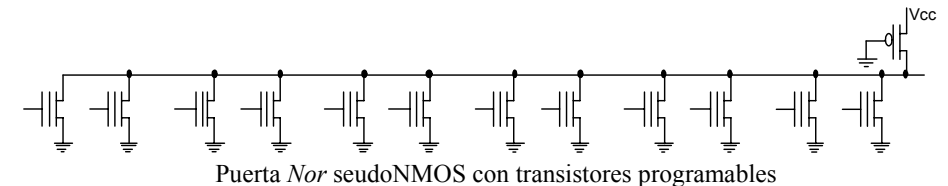
En general, los bloques programables **PAL** requieren puertas con un alto número de entradas: cada término producto recibe un alto número de variables (sumando las de entradas, las salidas de los biestables realimentadas y las negadas de ambas) y en cada módulo el número de términos producto puede ser relativamente alto.

La conexión en serie de un número alto de transistores se evita utilizando puertas "o-negada" (*Nor*) pseudoNMOS: el plano N tiene todos sus transistores NMOS en paralelo y el plano P se reduce a un solo transistor como resistencia de polarización (tales puertas tienen el inconveniente de que su consumo estático, para salida **0**, no es nulo).

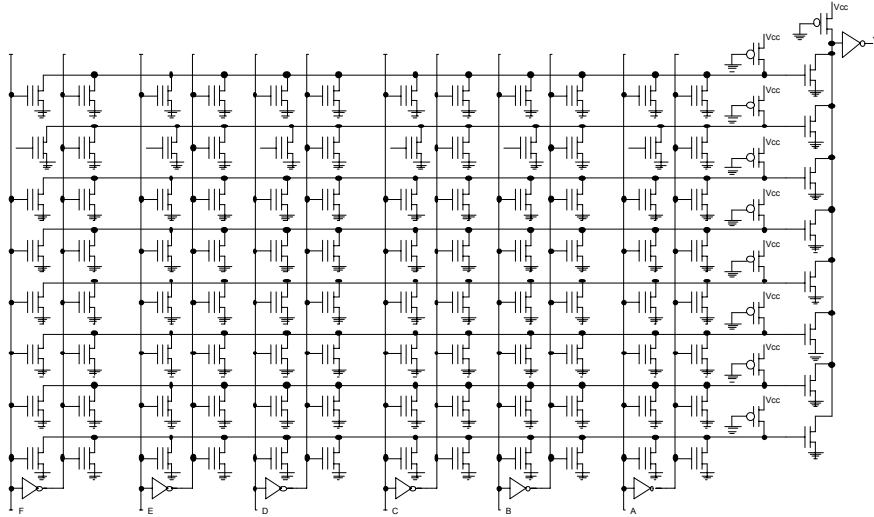


Además, en cuanto a dispositivos programables, esta configuración pseudoNMOS hace que solamente sea necesario programar el plano N, pues el plano P es fijo, constituido por un solo transistor: solamente se utilizan transistores EPROM de tipo NMOS y su programación consiste en eliminar («desconectar») los correspondientes transistores.

Ciertamente la programación de ambos planos sería sumamente compleja, pues, en uno de ellos, los transistores se encontrarían en serie y su programación consistiría en «puentearlos» (lo cual es mucho más difícil que «desconectarlos»).



La figura siguiente representa un módulo PAL de 6 entradas y 8 términos producto.

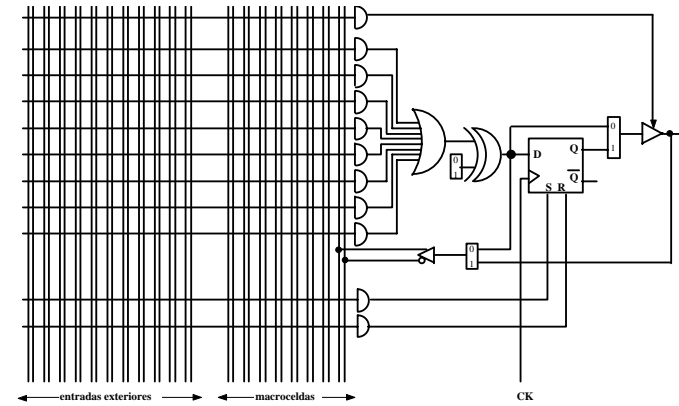


Macroceldas

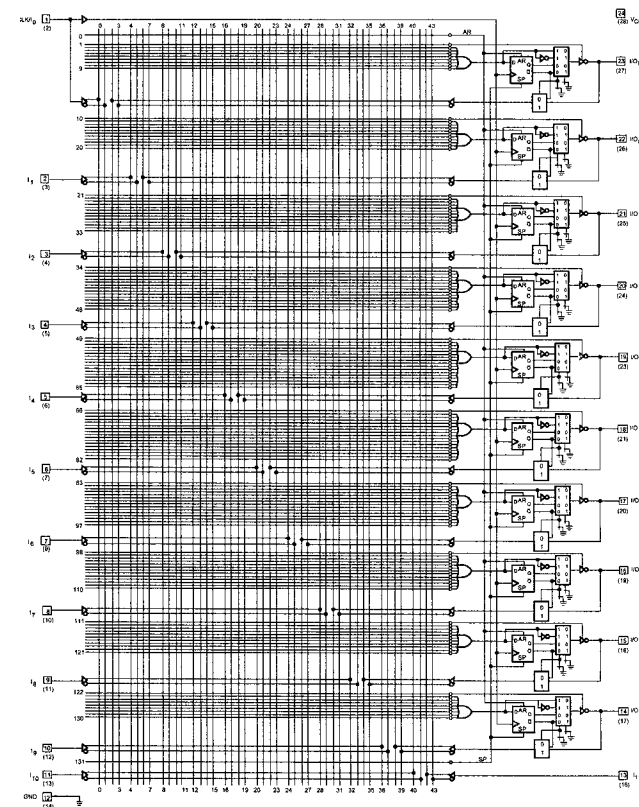
Una macrocelda está compuesta por:

- una suma de productos programables de múltiples entradas entre las que se encuentran su salida y las salidas de otras macroceldas (realimentación), estando disponible cada entrada en su forma afirmada y negada,
- una puerta "o-exclusiva" programable que permite configurar la función de la macrocelda (suma de términos producto) en forma afirmada (y) o negada (y̅, que equivale a producto de términos suma),
- un biestable que recibe dicha función,
- un selector que permite «puentear» el biestable y que configura, por tanto, la macrocelda como combinatorial o como secuencial,
- y un adaptador triestado con capacidad para «desconectar» la macrocelda respecto del terminal de salida, lo cual da lugar a las siguientes posibilidades:
 - la simple desconexión (alta impedancia) del terminal,
 - su utilización bidireccional (I/O),
 - su disponibilidad como entrada, caso de que la macrocelda no sea utilizada.

Mientras el número de macroceldas de un circuito integrado programables no es alto, cada una de ellas recibe todas las entradas del dispositivo y la realimentación de todas las salidas del mismo, que coinciden directamente con las salidas de todas sus macroceldas: en caso de que alguna macrocelda no sea utilizada, la salida correspondiente puede ser usada como entrada.



Macrocella con 8 términos producto



Configuración del circuito integrado programable PLD 22V10

Si bien el nombre de **PLD** es genérico (y, en principio, engloba a cualquier circuito integrado programable) suele utilizarse para referirse a los dispositivos programables de tipo PAL, es decir, módulos PLS: PAL + biestable.

Cuando se trata de un gran número de macroceldas dentro de un mismo integrado no resulta adecuado que cada una de ellas reciba todas las entradas y se conecte con todas las macroceldas ya que la cantidad de conexiones programables sería sumamente elevada; tampoco es viable que las salidas de todas las macroceldas se conecten a terminales de salida del circuito integrado.

Para limitar el tamaño del circuito y el número de sus terminales, las macroceldas se agrupan en bloques disjuntos y reciben solamente las entradas propias del bloque y la realimentación de las macroceldas del mismo; además, no todas las salidas de las macroceldas están conectadas a terminales de salida sino solamente un número reducido de cada bloque.

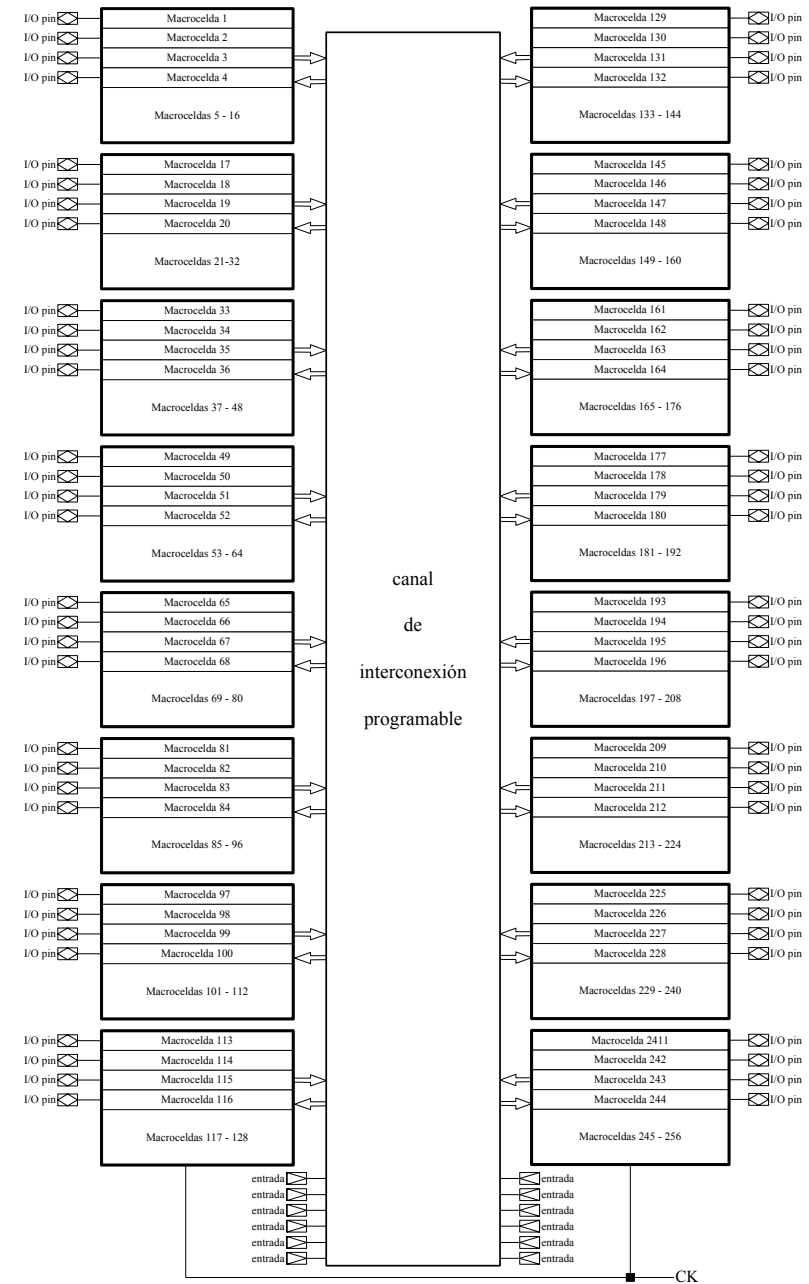
Los bloques se conectan entre sí a través de un bus central de conexiones programables conformando una estructura ramificada: un canal o eje de conexiones (*a manera de tronco*) sobre el cual se insertan los bloques de macroceldas (*como ramas del mismo*). Este tipo de circuitos integrados programables de amplio número de macroceldas con arquitectura ramificada es nombrado con las siglas **CPLD** (dispositivos lógicos programables complejos).

Para reducir la superficie de integración y evitar un alto desaprovechamiento de las estructuras PAL, el número de términos producto de cada macrocelda suele ser pequeño (4 ó 6) y para ampliarlo, cuando se necesite, se incluyen en cada bloque «expansores», en forma de términos producto adicionales que, sin pertenecer directamente a ninguna de las macroceldas del bloque, pueden ser utilizados por cualquiera de ellas.

La figura de la página siguiente representa un CPLD de 256 macroceldas, de las cuales solamente 64 pueden utilizarse como salidas exteriores (las otras 192 macroceldas son internas); el circuito dispone de una entrada específica de reloj y de 12 entradas exteriores, pudiéndose emplear también como entradas los terminales correspondientes a macroceldas que no se usen como tales.

Para proteger los diseños almacenados en estos dispositivos, frente a duplicación o copia de los mismos, suelen disponer de un «fusible de seguridad» que, una vez programado, impide la «lectura» del «mapa de fusibles» y, con ello, la duplicación del circuito.

Además, suelen ofrecer la posibilidad de seleccionar la velocidad de trabajo (y con ello, el consumo) a través de la programación de un «fusible de velocidad»: el dispositivo puede trabajar con dos conjuntos de tiempos funcionales diferenciados; cuando se selecciona máxima velocidad el consumo es mayor, pudiendo actuar con menores intensidades si la velocidad no es tan alta.



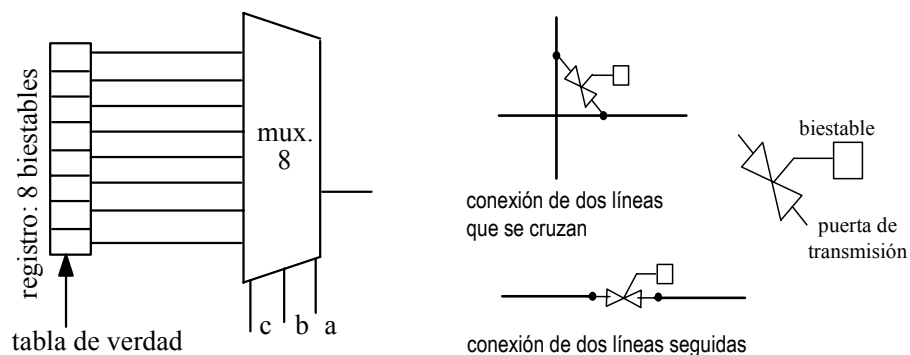
Configuración CPLD con 256 macroceldas

En paralelo con el desarrollo de las FPGAs, las series recientes de CPLDs incluyen también celdas periféricas adaptadoras de entradas/salidas análogas a las que se describen más adelante. Asimismo existen CPLDs de muy alto número de macroceldas con la estructura reticular propia de las FPGAs: los bloques de macroceldas separados por canales de conexión horizontales y verticales. Estas características de los CPLDs «avanzados» serán estudiadas en el apartado 3.

22.2. Los biestables como componentes programables: SRAM

En la configuración LUT (*look-up-table*) las funciones se construyen a través de su tabla funcional, mediante un multiplexor que «toma valores» sobre dicha tabla; dichos valores pueden ser almacenados en un registro.

Las conexiones entre funciones (o entre módulos) son seleccionadas, también, a través de biestables que actúan sobre puertas de transmisión: el valor booleano (1/0) almacenado en un biestable determina que las dos líneas conductoras enlazadas por la puerta de transmisión queden conectadas entre sí o permanezcan independientes.



De esta forma, las funciones se configuran mediante su tabla de verdad que se memoriza en un registro y las conexiones se establecen a través de los valores que se almacenan también en registros (en conjuntos de biestables).

En ambos casos, se establece una programación por memoria (*software*), a través de un amplio número de biestables que almacenan las tablas de las funciones y la selección de conexiones entre líneas. De forma que tanto las funciones booleanas de los módulos (*look-up-table*) como las conexiones (selección de caminos posibles) se programan almacenando valores booleanos sobre registros; las siglas **SRAM** identifican este tipo de «programación por memoria estática».

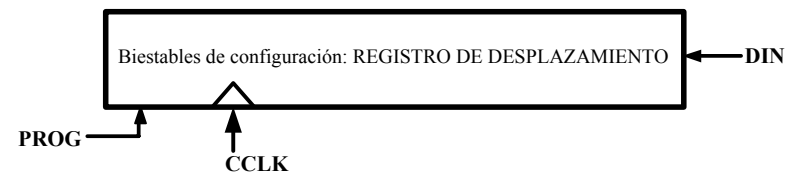
Este tipo de dispositivos, con programación SRAM, se denomina FPGA y será estudiado en detalle en el próximo apartado; están formados por:

- celdas con funciones lógicas configurables y biestables
- celdas adaptadoras de entradas y salidas configurables
- conexiones configurables
- y la memoria (el conjunto de biestables) donde se almacena la configuración de todo lo anterior junto con el circuito de control de la misma.

La programación se refiere tanto a las «tablas de verdad» de las funciones como a la configuración interna de las celdas en que éstas se encuentran y la de las celdas adaptadoras de entrada/salida y a las conexiones entre celdas. En conjunto será un amplio número de biestables cuyas salidas actúan sobre líneas de entrada de multiplexores (en el caso de «tablas de verdad») o sobre las entradas de control de otros multiplexores (en el caso de configuración de celdas o de conexiones entre ellas).

El número de biestables es muy alto: una FPGA mediana, de unos 2000 módulos lógicos suele tener del orden de 10^6 biestables de programación, es decir, unos 500 biestables por cada módulo, de los cuales unos 50 son de configuración del propio módulo y el resto de programación de las conexiones entre módulos.

En relación con su programación, es decir, con el almacenamiento de los valores booleanos correspondientes a un determinado diseño, todos estos biestables se encuentran conectados formando un largo registro de desplazamiento, en el cual se introduce, bit a bit, el «mapa de valores» que corresponde a dicho diseño. Este registro de desplazamiento es análogo al que se forma con los biestables de un circuito secuencial para efectuar el test del mismo: *scan path* (que será estudiado en 25.4).



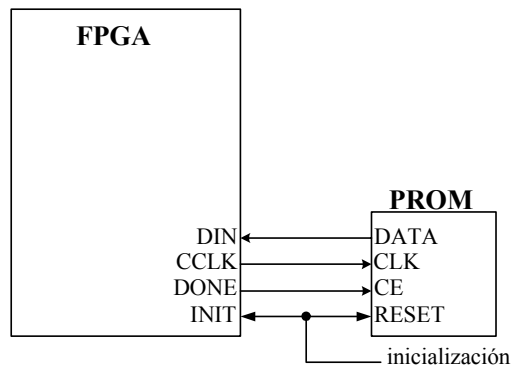
Una entrada de control PROG (selector de modo de programación), determina el paso del circuito a la situación de programación, en la cual todos los biestables de configuración (de programación) SRAM se unen, en cadena, formando un registro de desplazamiento, con una entrada DIN (entrada de datos para la programación) que recibirá los dígitos del «mapa de valores», bit a bit conforme a una señal de reloj CCLK que desplaza los dígitos a lo largo del registro. Un contador auxiliar sirve para contar el número de dígitos recibidos y señalar (salida DONE) el momento en que se completa la programación: se ha llenado todo el registro.

Sin tensión de alimentación el dispositivo carece de programación; en la puesta en marcha (*power-on*), la primera tarea que realiza el dispositivo consiste en leer su programación: ésta puede ser proporcionada por una PROM auxiliar (generalmente una PROM serie que se comunica con la FPGA a través de un reducido número de líneas) o bien suministrada directamente desde un computador o microprocesador.

La figura siguiente muestra la conexión entre una FPGA y una memoria PROM serie que contiene su programación; las líneas de conexión para efectuar la programación son las siguientes:

	PROM	FPGA	
- línea de control	RESET	INIT	señal externa de inicio
- línea de datos	DATA →	DIN	lo envía la PROM
- línea de reloj	CLK ←	CCLK	lo envía la FPGA
- línea de finalización	CE ←	DONE	lo envía la FPGA.

Al activar una señal de inicio (RESET / INIT) la FPGA lee el contenido de la memoria PROM a través de la línea de datos (DATA → DIN), lectura que es controlada mediante dos líneas que la FPGA envía a la PROM: una línea de reloj (CLK ← CCLK) y otra de habilitación (CE ← DONE); la segunda línea determina la finalización de la lectura cuando se ha completado el fichero necesario para la programación.



También pueden utilizarse las memorias PROM habituales de 8 líneas de datos; su lectura será más rápida pero requieren un mayor número de líneas dedicadas a la programación.

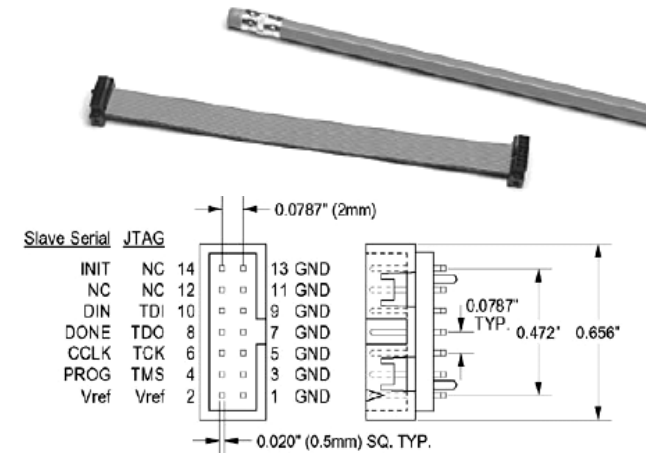
Asimismo, el fichero de programación de una FPGA puede ser enviado desde un computador o desde un microprocesador a través de 4 líneas de interconexión:

	procesador	FPGA	
- línea de control	TMS →	PROG	control de la programación
- línea de datos	TDI →	DIN	lo envía el procesador
- línea de reloj	TCK →	CCLK	lo envía el procesador
- línea de finalización	TDO ←	DONE	lo envía la FPGA.

Basta, pues, con dejar 4 terminales en la placa de la FPGA preparados con este fin; es el computador el que envía el reloj a la FPGA para sincronizar la transmisión de datos, mientras que con una PROM es la FPGA la que envía el reloj a la memoria para leerla.

En ambos casos la FPGA recibe los datos de la programación a través de la línea DIN e informa de que se ha completado tal programación a través de la línea DONE.

La siguiente figura muestra un cable normalizado de interconexión: a las 4 líneas básicas de programación se les añaden otras líneas con la referencia de «tierra» y la alimentación.



La figura que sigue muestra un adaptador para conectar el cable anterior al puerto paralelo de un PC:



De igual forma la programación puede realizarla un microprocesador presente en la misma placa circuital, a través de 4 líneas del circuito impreso.

Como puede apreciarse, la programación de los dispositivos SRAM se efectúa en el mismo sistema digital del que forman parte: es una programación en el propio circuito global aplicado **ISP** (*in system programmable*). Estos dispositivos, programables directamente en el propio sistema digital del que forman parte, permiten, incluso, el cambio de su programación durante el funcionamiento del mismo (de manera que pasen a realizar funciones diferentes de las que estaban realizando).

No todas las FPGAs son de tipo SRAM; la configuración LUT y selección de línea por multiplexores puede también conformarse con programación E²PROM o FLASH, reflejando sobre simples transistores MOS programables los valores 0/1 que en el caso SRAM se almacenan en los biestables.

Las FPGAs de tipo E²PROM o FLASH suelen incluir internamente la generación de las tensiones y la circuitería necesaria para su borrado y programación, de forma que también pueden ser programadas en el propio sistema, de manera análoga a las de tipo SRAM; son pues dispositivos ISP (programables en el sistema a través de 4 líneas).

Existen también FPGAs con programación destructiva, mediante antifusibles. Los antifusibles consisten en contactos entre polisilicio y difusión separados inicialmente por un aislante de óxido de silicio SiO₂ muy fino, de forma que la perforación del aislante pone en contacto la difusión con el polisilicio; la resistencia inicial del antifusible es superior a 100 MΩ y se reduce por debajo de los 500 Ω al perforar el óxido de silicio.

Los antifusibles se programan por «perforación eléctrica» a través de la acción de un campo eléctrico fuerte, producido por una tensión relativamente elevada; para ello se utilizan pulsos de tensión de unos 21 V muy repetitivos (en dos pasos: una precarga de todos los antifusibles a 10,5 V, para evitar la perforación simultánea de varios de ellos, seguida de la aplicación de 21 V al fusible particular que se quiere perforar).

22.3. Matrices de celdas lógicas programables: FPGAs

Las matrices de celdas programables FPGA son dispositivos de «granularidad fina», compuestos por una matriz de bloques lógicos, formados por 2 ó 4 parejas de celdas lógicas de reducido tamaño (y limitada capacidad booleana), separados por canales de conexión verticales y horizontales, con celdas de entrada/salida situadas en la periferia, que conectan con los terminales del dispositivo.

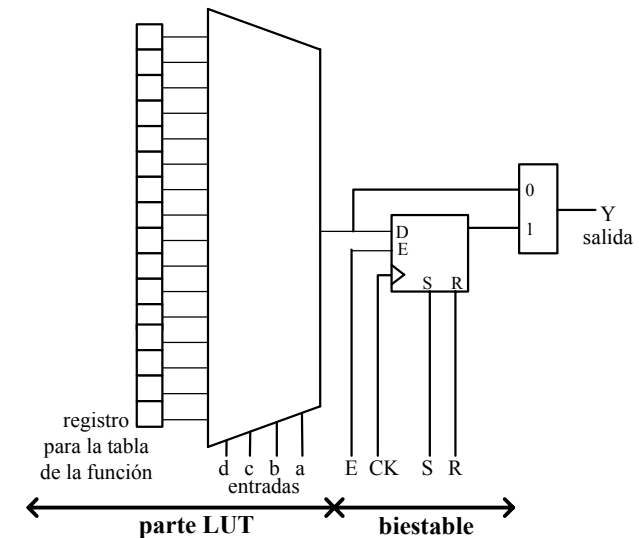
El calificativo programable adjetiva a las celdas y a la matriz de conexiones, es decir, tiene una doble repercusión: son programables tanto las celdas internamente (su función booleana y su configuración) como las conexiones de las celdas entre sí.

La descripción que, a continuación, se detalla pretende ser una aproximación conceptual y genérica a la conformación de las FPGAs, que proporcione una idea clara y justificada de su configuración y de sus posibilidades; la configuración concreta de una FPGA depende del fabricante y de la serie específica y alcanza progresivamente mayores niveles de complejidad.

Celdas, módulos y bloques lógicos

Las celdas lógicas cuentan con una parte combinacional, capaz de realizar una función booleana de pocas entradas (generalmente 4 entradas), un biestable y algunos «multiplexores de configuración»; las funciones se construyen mediante multiplexores por el procedimiento de «búsqueda sobre tabla» LUT (*look-up-table*).

El biestable ofrece una entrada de habilitación **E** para mantener su valor cuando interese (**E = 0**), lo cual simplifica mucho el diseño de determinados registros (como los de retención o los contadores) y sendas entradas asíncronas de marcado **S** y de borrado **R**, para la inicialización del biestable.



De esta forma, una celda lógica ofrece una función de 4 variables, con salida directa (combinacional) o «registrada» (a través de un biestable). Caso de que no se utilice la celda como función booleana, pueden emplearse los biestables de programación de la misma en las tres formas siguientes:

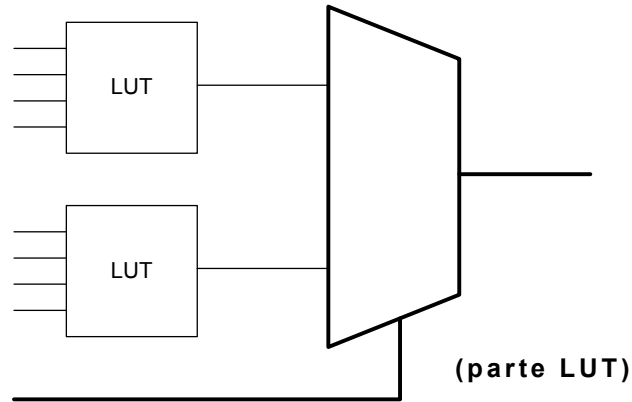
- como registro de desplazamiento (muy rápido) de hasta 16 biestables (el multiplexor LUT permite controlar la longitud de dicho registro) o de 17 biestables utilizando el propio de la celda lógica;
- como memoria ROM de 16 x 1 bits, fijando los 16 valores sobre el registro de programación y tomando la salida del multiplexor LUT;
- como memoria RAM de 16 x 1 bits, utilizando los 16 biestables de programación como biestables numerados; en principio, la línea de entrada y la de salida de dicha memoria son independientes: se incluye en la celda la lógica necesaria para escribir selectivamente en uno de los biestables y la lectura es a través del multiplexor LUT (que selecciona el biestable que aporta su valor).

Asimismo, la función LUT puede ser configurada como multiplexor rápido de 2 líneas.

De manera que una celda lógica equivale a:

- una función de 4 variables + biestable,
- un registro de desplazamiento de hasta 17 bits,
- una pequeña memoria ROM de 16 x 1 bits,
- una pequeña memoria RAM de 16 x 1 bits,
- un multiplexor de 2 líneas de entrada.

Dos celdas lógicas se agrupan en un módulo lógico, con un multiplexor de las dos salidas LUT a fin de poder configurar funciones de 5 variables.



De esta forma, un módulo lógico puede ser utilizado como:

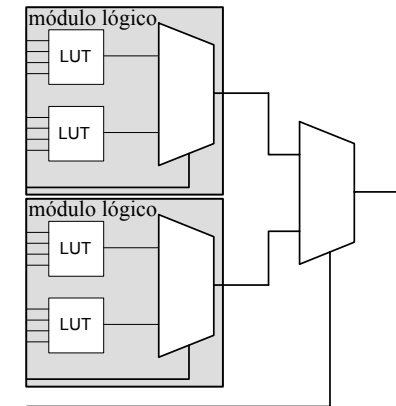
- dos funciones de 4 variables + 2 biestables,
- una función de 5 variables + 2 biestables (el bloque permite la programación de cualquier función de 5 variables y, también, algunas funciones, no todas, de hasta 4 variables),
- dos registros de desplazamiento de hasta 17 bits cada uno o uno de hasta 34 bits,
- una pequeña memoria ROM de 16 x 2 bits o de 32 x 1 bits,
- una pequeña memoria RAM de 16 x 2 bits o de 32 x 1 bits,
- un multiplexor de 4 líneas de entrada.

Además, según veremos en el próximo subapartado, los módulos lógicos incluyen una circuitería adicional específica para operaciones aritméticas rápidas.

Los módulos lógicos se agrupan para formar bloques lógicos que, separados por canales horizontales y verticales de conexión, forman la «matriz» básica de las FPGAs. Los bloques lógicos contienen dos o cuatro módulos lógicos, o sea, 4 u 8 celdas lógicas.

Un bloque lógico de 2 módulos (ver la figura siguiente), además de las prestaciones que ofrecen sus dos módulos por separado, puede ser utilizado, conjuntamente, como:

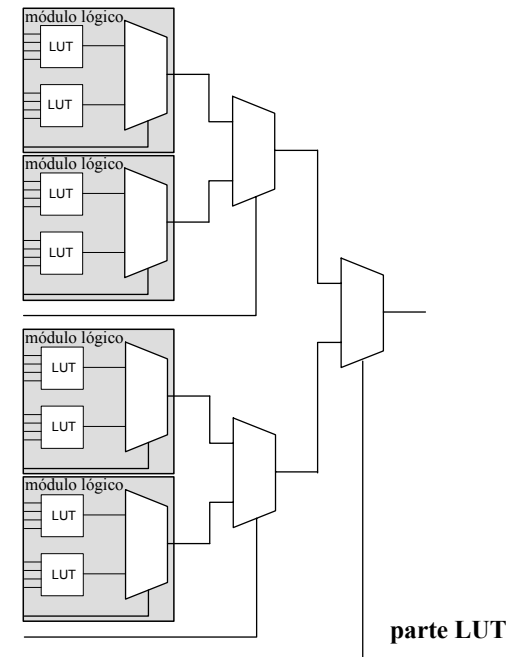
- una función de 6 variables + 4 biestables (el bloque permite la programación de cualquier función de 6 variables y algunas funciones de hasta 19),
- un registro de desplazamiento de hasta 68 bits,
- una memoria ROM de 16 x 4 o de 32 x 2 o de 64 x 1 bits,
- una memoria RAM de 16 x 4 o de 32 x 2 o de 64 x 1 bits,
- un multiplexor de 8 líneas de entrada.



Bloque lógico formado por 2 módulos lógicos

Cualquiera de las prestaciones que se configuran dentro de un bloque lógico son de alta velocidad, ya que las funciones y conexiones, en su interior, están optimizadas en cuanto a tiempos de propagación.

Las FPGAs «avanzadas», de alta capacidad, contienen 4 módulos lógicos en cada bloque lógico (en cada celdilla de su matriz):



Bloque lógico formado por 4 módulos lógicos

Un bloque lógico de 4 módulos, además de las prestaciones de dichos módulos, puede ser utilizado como:

- una función de 7 variables + 8 biestables (permite la programación de cualquier función de 7 variables y algunas funciones de hasta 39 variables),
- un registro de desplazamiento de hasta 136 bits,
- una memoria ROM de 16 x 8 o de 32 x 4 o de 64 x 2 o de 128 x 1 bits,
- una memoria RAM de 16 x 8 o de 32 x 4 o de 64 x 2 o de 128 x 1 bits,
- un multiplexor de 16 líneas de entrada.

Recursos para operaciones aritméticas

A fin de poder configurar operaciones aritméticas y que éstas puedan efectuarse a alta velocidad, se incluye en cada módulo lógico una circuitería adicional específica para el diseño de sumadores, restadores y multiplicadores. Dicha circuitería permite, también, la configuración de contadores rápidos, que pueden ser bidireccionales, con borrado síncrono y con carga paralelo.

Una celda sumadora requiere las siguientes funciones:

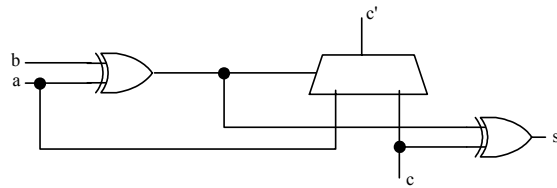
suma: $s = c \oplus (b \oplus a)$

acarreo: $c' = b \cdot a + c \cdot (b + a)$;

en relación con la velocidad, hay que prestar particular atención a la función de acarreo porque es recursiva: el acarreo de cada dígito c' debe esperar al acarreo del dígito anterior c_{-} .

Ahora bien, dentro de la función acarreo hay dos terminos: la generación de acarreo $a \cdot b$ y la propagación del acarreo anterior $(a+b) \cdot c_{-}$, siendo este segundo el que limita la velocidad; resulta eficaz la utilización de un multiplexor rápido que transmita el acarreo «propio» cuando éste es generado ($a = b = 1$) o el anterior cuando a o b son 1.

Las dos funciones de la celda sumadora pueden construirse de la siguiente forma, que aprovecha una puerta "o-exclusiva" para ambas:

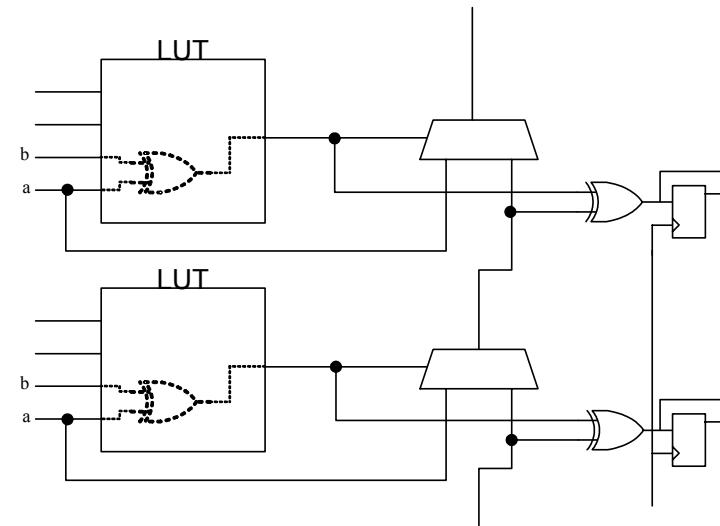


suma: $s = c \oplus (b \oplus a)$

acarreo: $c' = (\overline{b \oplus a}) \cdot a + (b \oplus a) \cdot c_{-} = (\overline{b \cdot a} + b \cdot a) \cdot a + (b \cdot \overline{a} + \overline{b} \cdot a) \cdot c_{-}$
 $= b \cdot a + (b \cdot \overline{a} + \overline{b} \cdot a) \cdot c_{-} = b \cdot a + c_{-} \cdot (b + a)$

Con ello, un módulo lógico queda en la forma de la figura siguiente, donde la primera puerta "o-exclusiva" se configura en la parte funcional (LUT) de cada celda y se

añade el multiplexor y la segunda puerta "o-exclusiva" como elementos específicos para operaciones aritméticas:



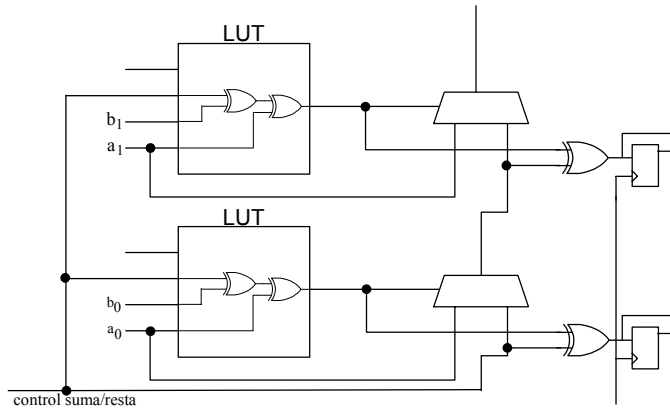
De esta forma, cada celda lógica se convierte en una celda sumadora (el módulo lógico es un sumador de 2 dígitos). Esta configuración presenta muy bajos tiempos de retraso en lo que se refiere a la propagación del arrastre, ya que el paso a través de multiplexores (que se encuentran todos ellos previamente habilitados, conforme al valor de su entrada de control) es muy rápido y las líneas de conexión son directas y cortas. Téngase en cuenta que, en muchas ocasiones, los tiempos de propagación a través de las conexiones son mayores que los que presentan las puertas lógicas.

Además, la salida de «propagación de arrastre» de cada módulo lógico se encuentra unida, a través de una línea directa y corta, con la entrada de «propagación de arrastre» del siguiente módulo lógico de la misma columna (el que se encuentra «encima» del mismo de acuerdo con la disposición matricial de las FPGAs); es decir, existe una línea directa y «mínima» de propagación de arrastre a lo largo de cada columna de módulos lógicos que permite la configuración de sumadores ultrarrápidos en dicha columna.

Aprovechando aún más la parte funcional (LUT) de las celdas lógicas, el anterior sumador puede convertirse en un sumador/restador (en complemento a 2) controlado por una entrada de selección. La resta $A-B$, en complemento a 2, equivale a sumar a A el inverso de B más una unidad, ya que el cambio de signo de B equivale a invertirlo y sumarle una unidad:

$$A - B = A + (-B) = A + (\overline{B} + 1) = A + \overline{B} + 1.$$

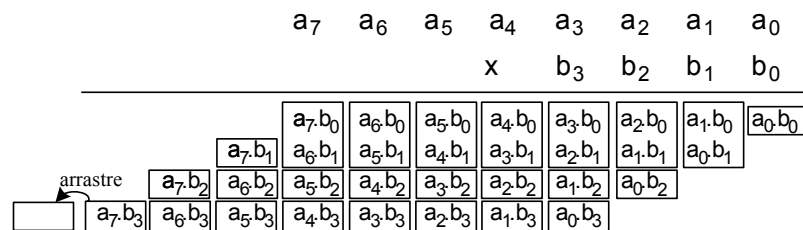
La suma de dicha unidad (+1) se efectúa haciendo el arrastre inicial igual a 1. La siguiente figura representa la configuración del primer módulo lógico para un sumador/restador que efectúa la suma cuando $s = 0$ y la resta cuando $s = 1$; la primera puerta "o-exclusiva" de cada celda invierte el dígito de B cuando $s = 1$ (dejándolo pasar sin invertir cuando $s = 0$) y la entrada de selección s actúa como arrastre inicial para sumar una unidad en el caso de la resta.



Habida cuenta de que contar es ejecutar la suma $n + 1$ (sumar una unidad) sobre un registro y que contar «hacia abajo» equivale a restar una unidad ($n - 1$), esta misma configuración, utilizando sus biestables como registro (número n) permite construir contadores bidireccionales que admiten frecuencias de reloj muy altas: siendo Q el estado de los biestables, basta hacer $A = Q$ y $B = \dots 00001$; la entrada de habilitación de los biestables sirve también para la habilitación de los contadores.

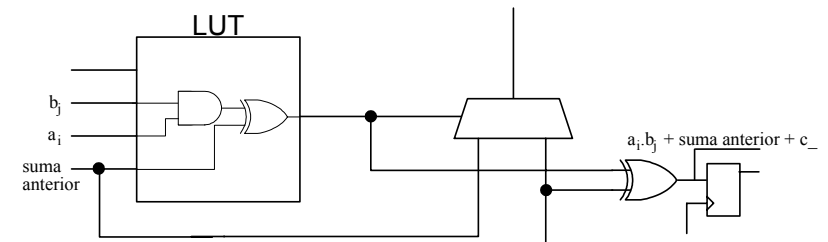
Además del multiplexor de propagación de arrastre y de la puerta "o-exclusiva", se añade en cada celda una puerta "y" (en el control del citado multiplexor) para facilitar la construcción de multiplicadores y de contadores con carga paralelo y borrado síncrono.

La forma habitual de efectuar un producto

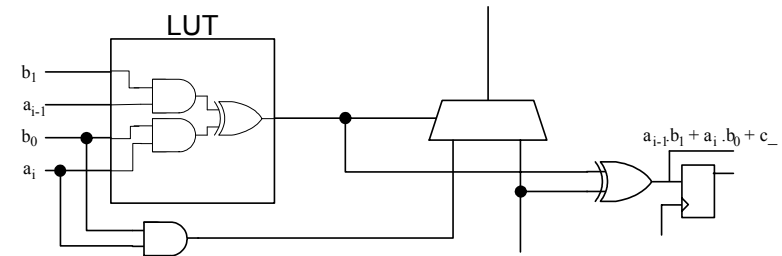


requiere sumadores para los productos lógicos $a_i \cdot b_j$; en la operación anterior, cada cuadrado es una suma de tipo $a_i \cdot b_0 + a_{i-1} \cdot b_1$ y cada rectángulo lo es en la forma $a_i \cdot b_j + \text{suma anterior}$; ambos tipos de sumadores pueden ser realizados utilizando una sola celda (en total, 26 celdas).

Los rectángulos son sumadores $a_i \cdot b_j + \text{suma anterior}$ que pueden ser configurados en una celda en la siguiente forma:



En cambio, para la suma $a_i \cdot b_0 + a_{i-1} \cdot b_1$ de las dos primeras filas se necesita disponer de una puerta "y" adicional en la forma representada en la figura siguiente:



De esta forma un multiplicador de $8 \times m$ dígitos requiere un sumador de 9 bits (y una puerta "y": $a_0 \cdot b_0$) para las dos primeras filas y otro sumador de 8 bits para cada una de las filas siguientes más una celda lógica que recoja el arrastre más significativo. El producto de $n \times m$ dígitos necesita $m - 1$ sumadores con un total de $n \cdot (m - 1) + 2$ celdas lógicas.

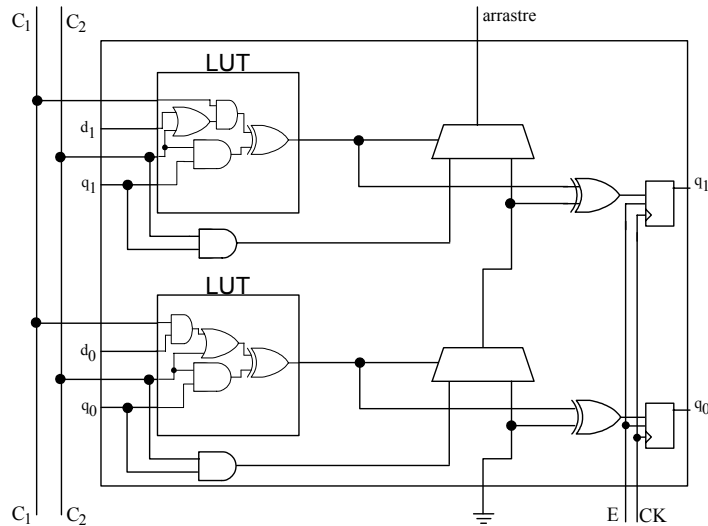
Un multiplicador de números binarios de 16 dígitos requiere un amplio número de celdas lógicas (242 celdas: 121 módulos lógicos) pero realiza el producto muy rápidamente, en un solo ciclo de reloj. En el capítulo 24 se considera otra opción de construcción de multiplicadores, en forma de máquina algorítmica, de menor tamaño (36 celdas: 18 módulos lógicos) pero, también, con menor velocidad pues requiere 32 ciclos de reloj para realizar el producto. Es habitual, en el diseño digital, la existencia de un compromiso entre velocidad y tamaño de los circuitos.

Además, habida cuenta del gran número de celdas lógicas que requiere la configuración de un multiplicador largo, las FPGAs avanzadas incorporan, como módulos específicos, multiplicadores de 16 o de 32 bits

La puerta "y" que controla el multiplexor del arrastre permite configurar la celda en la forma $a \cdot b + F(a, b, c, d)$ (suma aritmética) que da lugar a contadores bidireccionales, con borrado síncrono y carga paralelo, a través de las siguientes expresiones:

$$q_i^+ = \text{Suma aritmética}[\langle q_i, C_2 \rangle \text{ "+" } \langle C_1 \cdot (d_i + C_2) \rangle \text{ "+" } \langle \text{arrastre del bit anterior} \rangle]$$

$$\text{para } i \neq 0 \text{ y para el primer dígito } q_0^+ = \text{Suma aritmética}[\langle q_0, C_2 \rangle \text{ "+" } \langle C_1 \cdot d_0 + C_2 \rangle].$$



Esta figura representa un módulo lógico con los dos primeros dígitos del contador (unidades y «dosenas»); las celdas correspondientes a los siguientes dígitos tendrán la misma configuración que la celda superior (la de arriba, q₁) de este módulo. La entrada de control C₂ puede ser considerada como habilitación de contaje (cuando C₂ = 1) o bien como habilitación de carga negada (C₂ = 0):

- cuando C₂ = 0 se efectúa la carga si C₁ = 1; si C₁ = 0 se produce borrado síncrono;
- cuando C₂ = 1 el contador realiza su operación propia de contaje: si C₁ = 0 el contaje es ascendente y si C₁ = 1 es hacia abajo.

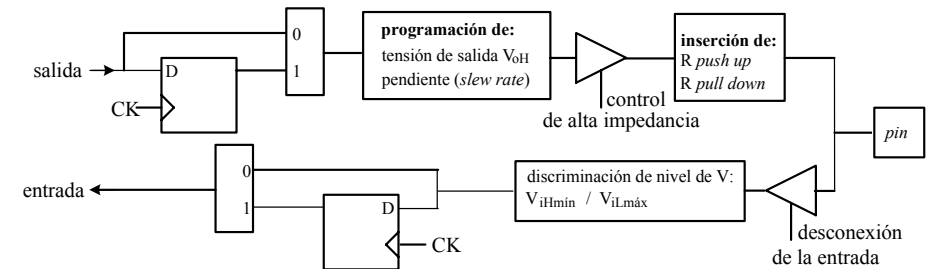
C ₂ = 0	C ₁ = 0	q _i ⁺ = 0	Q = 000000....	borrado síncrono
C ₂ = 0	C ₁ = 1	q _i ⁺ = d _i	Q = D	carga paralelo
C ₂ = 1	C ₁ = 0	q _i ⁺ = Suma aritmética[q _i "+" 0 "+" "arrastre"]	q ₀ ⁺ = Suma aritmética[q ₀ "+" 1]	suma Q + ...00001 contaje hacia arriba
C ₂ = 1	C ₁ = 1	q _i ⁺ = Suma aritmética[q _i "+" 1 "+" "arrastre"]	q ₀ ⁺ = Suma aritmética[q ₀ "+" 1]	suma Q + ...11111 contaje hacia abajo

Este contador bidireccional (cuya entrada de habilitación del contaje es la propia de los biestables), con borrado síncrono y carga paralelo, requiere solamente una celda lógica para cada dígito.

Celdas periféricas adaptadoras de entradas/salidas

Las celdas de entrada/salida contienen sendos biestables para sincronizar la entrada y la salida, adaptadores triestado para «desconectarlas» y una serie de multiplexores de configuración que permiten programar:

- la utilización de la celda como entrada, como salida o como terminal bidireccional
- respecto a la entrada:
 - su desconexión como tal entrada
 - su paso a través de biestable: sincronismo de la entrada
 - los niveles de tensión para discriminar los valores booleanos 0 / 1
- respecto a la salida:
 - su paso o no a través de biestable: sincronismo de la salida
 - el control de la situación de alta impedancia: triestado
 - el valor de la tensión de salida para el 1 lógico
 - la pendiente de los flancos de salida (slew rate): lento o rápido
 - la inclusión de una resistencia de push-up o de pull-down.



En relación con las tensiones de salida (es decir, con V_{oH}, ya que V_{oL} = 0 V) y de entrada (con V_{iHmin} y V_{iLmax}) las FPGAs suelen presentar múltiples posibilidades (en ocasiones hasta una decena de ellas) de niveles de tensión para facilitar su conexión con otros circuitos integrados digitales o analógicos; en general admiten todo tipo de tensiones lógicas normalizadas (tanto TTL como CMOS y las diversas series de baja tensión), de las cuales debe seleccionarse en cada caso la que interesa mediante programación.

Conexión entre celdas

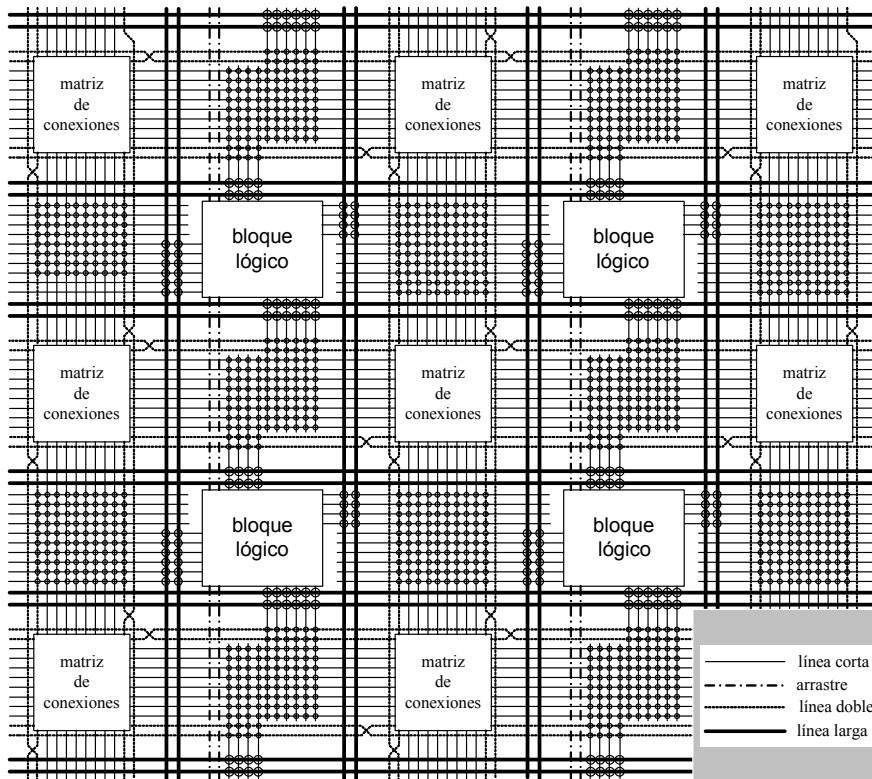
Los canales de conexión, verticales y horizontales, situados entre los bloques lógicos, contienen líneas de diferentes longitudes, con muy alta capacidad de interconexión entre ellas; a través de tales líneas se efectúan las conexiones de las celdas lógicas entre sí y de éstas con las celdas de entrada/salida.

Se trata de poder conectar dos elementos cualesquiera, optimizando, en lo posible, los tiempos de retardo que introducen las líneas de conexión. Ya, de entrada, se minimizan las longitudes de las líneas dentro de cada bloque lógico y en los canales entre bloques se dispone de tres tipos de líneas de conexión: *cortas* para conexión de bloques lógicos vecinos, *dobles* para los bloques siguientes y *largas* que recorren toda la fila (si son horizontales) o toda la columna (si son verticales).

En cada cruce de los canales de conexión horizontales con los verticales (es decir, en las cuatro esquinas de los bloques lógicos), hay una *matriz de conexiones* que permite prolongar las líneas que llegan a ella y conectarlas con las líneas que se cruzan en ella.

De forma que cada canal de conexiones está formado por *líneas cortas*, que van de una matriz de conexiones a la siguiente, junto con unas pocas *líneas dobles* (que enlazan matrices alternas) y algunas *líneas largas* que recorren todo el canal (y no pasan por las matrices de conexión). Las entradas y salidas de cada bloque lógico se prolongan en forma de líneas cortas perpendiculares a los canales de conexión.

La figura siguiente presenta una hipotética distribución de líneas y conexiones:

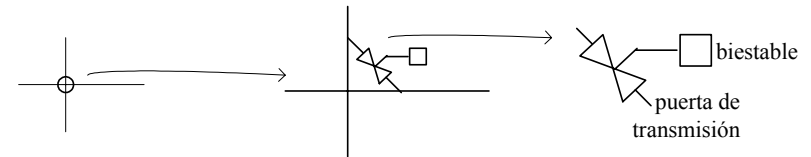


Las líneas cortas (de mínima resistencia y capacidad) son las más numerosas y relacionan a cada bloque y a cada matriz de conexión con los cuatro bloques y con las cuatro matrices inmediatamente contiguas: los bloques vecinos se conectan a través de sus propias líneas cortas (utilizando las líneas de canal que las cruzan); los bloques no contiguos lo hacen a través de las matrices de conexión.

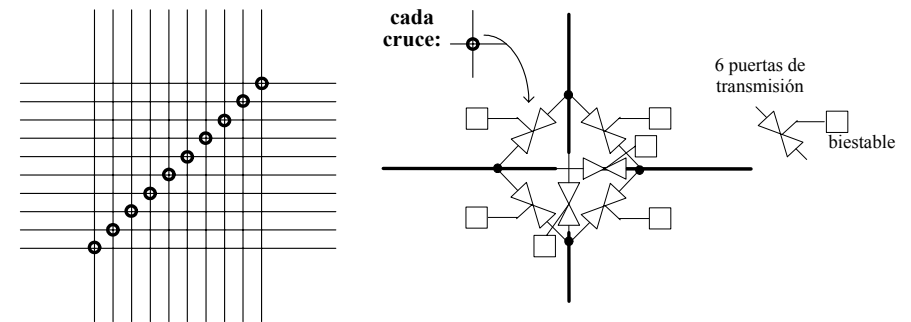
Las líneas dobles permiten la conexión directa entre matrices de conexión alternas y las líneas largas (más lentas) son para señales de tipo global (que van a muchos bloques) y para conexiones rápidas entre bloques muy alejados.

Las conexiones se configuran a través de puertas de transmisión (que equivalen a interruptores estáticos) controladas, cada una de ellas, por un biestable que determina su situación de conexión (biestable a **1**) o desconexión (biestable a **0**); las conexiones son siempre bidireccionales (ya que los transistores MOS y las puertas de transmisión formadas con ellos también lo son).

En los cruces señalados con círculo (en la figura anterior) hay una conexión programable entre las líneas que se cruzan:



En las matrices de conexión cada línea corta (o doble) puede prolongarse hacia la siguiente (situada «en línea», a continuación de ella) y también puede unirse a las otras dos líneas perpendiculares a ella.



En principio, las puertas de transmisión están formadas por sendos transistores NMOS (que transmite bien los «ceros») y PMOS (para transmitir los «unos») en paralelo; pero también es viable utilizar solamente transistores NMOS de paso (en cuyo caso conviene aumentar la tensión umbral de los transistores PMOS de la tecnología para recuperar la situación de consumo nulo de las puertas complementarias –ver T8.2–).

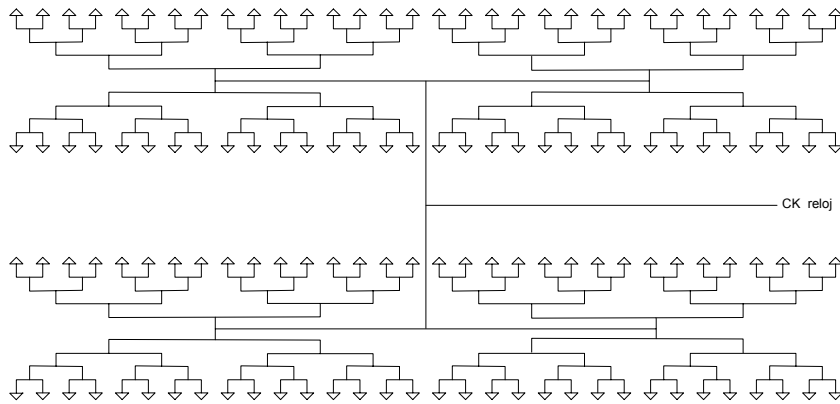
El número de conexiones en una FPGA es muy alto, de manera que los biestables de programación de las conexiones son siempre un número muy superior al de biestables de configuración: un módulo lógico suele tener unos 60 biestables para su programación (32 para las «tablas de verdad» de las 2 funciones de 4 variables que contiene y unos 28 de configuración del propio módulo); una FPGA de unos 2000 módulos tiene del orden de 10^6 conexiones programables (es decir, unas 500 por módulo).

La distribución del diseño entre los diversos bloques lógicos es de vital importancia respecto a la velocidad de trabajo que puede alcanzar dicho diseño (en forma análoga a la importancia que tiene la distribución de circuitos integrados en una placa de circuito impreso): los bloques que se relacionan entre sí deben estar lo más cercanos posible y los sumadores, restadores, multiplicadores y contadores deben situarse en la misma columna de módulos lógicos (para aprovechar la propagación directa del arrastre).

Distribución de la señal de reloj

Dada la importancia de la simultaneidad de la señal de reloj, en relación con el sincronismo, se incorporan esquemas de distribución de dicha señal que aseguran que recorre caminos de longitud prácticamente igual para los diversos bloques lógicos.

Distribuciones de tipo «fractal», como la representada en la figura siguiente, consiguen que la longitud de todas las líneas de distribución de reloj sea la misma: desde la conexión que la recibe (señalada como «CK reloj») hasta su aplicación a cada módulo lógico (indicada con un triángulo) la señal de reloj recorre caminos de igual longitud.



Esta forma de asegurar la simultaneidad de la señal de reloj aumenta, en forma apreciable, la longitud (ya de por sí grande) de las líneas de reloj y el largo recorrido que debe efectuar para llegar a los diversos biestables determina un retraso significativo entre el reloj interno y el reloj externo que recibe la FPGA.

Para evitar los efectos del retardo de propagación de la señal de reloj, las FPGAs avanzadas incorporan módulos de «resincronización de flanco» (bloques denominados **DLL**, *Delay Locked Loop*) que permiten sincronizar el reloj interno con el externo, es decir, retrasan el reloj interno de forma que sus flancos coinciden con los de la señal de reloj externa. De esta forma, en cada flanco activo (en los momentos de transición de una unidad de tiempo a la siguiente) el flanco actual del reloj interno procede de un flanco anterior del reloj de entrada pero coincide con el flanco actual de dicho reloj.

Memoria RAM y bloques incrustados

Al describir la celda lógica, se comentó la posibilidad de configurar los registros SRAM de la programación LUT (biestables previos a los multiplexores que conforman las funciones) como memorias RAM internas y rápidas; este tipo de memoria RAM, utilizando celdas no empleadas para funciones booleanas, se denomina memoria RAM distribuida.

Además, las FPGAs avanzadas suelen incluir bloques específicos de memoria RAM de varios *Kbytes*, a fin de disponer de módulos de memoria más amplios y compactos; esta memoria es, también, muy rápida por ser interna al dispositivo. Tales bloques de memoria suelen ser configurables; por ejemplo, un bloque de 8K x 8 bits puede utilizarse como 16K x 4, 32 K x 2 ó 64K x 1 bit y, también, como 4K x 16 ó 2K x 32 bits.

Asimismo, las FPGAs más grandes incorporan otros módulos operativos de utilidad «incrustados» (*embedded*) dentro del mismo circuito integrado; de esta forma, en el interior de la FPGA se encuentran disponibles algunos bloques grandes de utilidad general, sin necesidad de programarlos sobre módulos lógicos (lo que daría lugar a un mayor tamaño del bloque «distribuido» resultante y menor velocidad de trabajo del mismo). Por ejemplo, multiplicadores de 16 ó de 32 bits (ya citados en páginas anteriores), microprocesadores de 8 ó de 16 bits, etc.