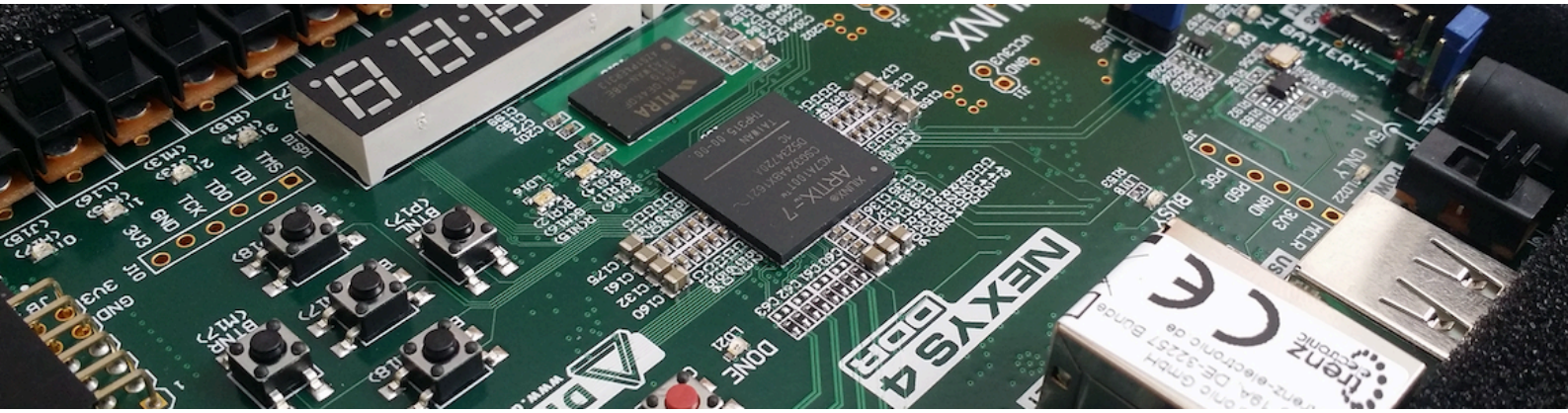

Sistemas Electrónicos Digitales

APUNTES

Alberto Brunete | SED | 2018

Sección I. Introducción

Sección II. FPGAs

APARTADO 2.01 ¿POR QUÉ LAS FPGAS?

Alrededor del comienzo de la década de 1980, se hizo evidente que había una brecha en el continuo digital de los IC. En un extremo, había dispositivos programables como SPLD y CPLD, que eran altamente configurables y tenían tiempos de diseño y modificación rápidos, pero que no podían soportar funciones grandes o complejas.

En el otro extremo del espectro estaban los ASIC. Estos podrían admitir funciones extremadamente grandes y complejas, pero su diseño era muy costoso y laborioso. Además, una vez implementado un diseño ASIC, se congela en silicio para siempre (no puede ser modificado).

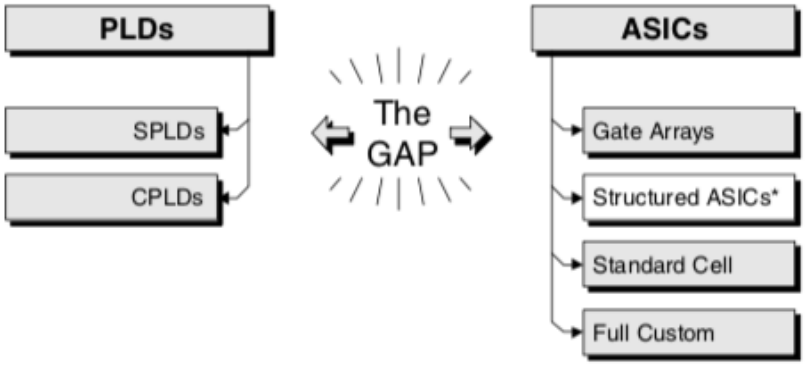

Figura 1. El "gap" de las ICs

Con el fin de resolver este problema, Xilinx desarrolló una nueva clase de IC llamada field-programmable gate array (matriz de compuertas programable en campo), o FPGA, que pusieron a disposición del mercado en 1984.

APARTADO 2.02 PRIMEROS PASOS E INTRODUCCIÓN A LAS FPGAS

Las primeros FPGA se basaban en CMOS y usaban celdas SRAM para fines de configuración. Aunque estos primeros dispositivos eran comparativamente simples y contenían relativamente pocas puertas (o su equivalente) para los estándares actuales, muchos aspectos de su arquitectura subyacente todavía se emplean hoy en día.

Los primeros dispositivos se basaban en el concepto de un bloque lógico programable, que comprendía una lookup table (tabla de búsqueda) de 3 entradas (LUT), un registro que podía actuar como un flip-flop o un latch, y un multiplexor, junto con algunos otros elementos que son de poco interés ahora mismo. La Figura 2 muestra un bloque lógico programable muy simple.

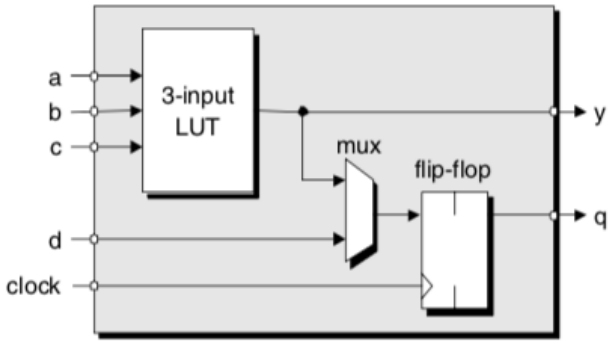

Figura 2. Bloque lógico programable simple

Cada FPGA contenía una gran cantidad de estos bloques lógicos programables. Por medio de células de programación SRAM apropiadas, cada bloque lógico en el dispositivo podría configurarse para realizar una función diferente. Cada registro podría configurarse para inicializarse a un 0 lógico o un 1 lógico y para actuar como un flip-flop (como se muestra en la Figura 2) o un latch. Si se seleccionó la opción del flip-flop, el registro podría configurarse para que se active mediante un flanco de reloj positivo o negativo (la señal del reloj era común a todos los bloques lógicos). El multiplexor que alimenta el flip-flop podría configurarse para aceptar la salida del LUT o una entrada separada al bloque lógico, y el LUT podría configurarse para representar cualquier función lógica de 3 entradas.

Por ejemplo, supongamos que se requirió una LUT para realizar la función:

y = (a & b) | !c

Esto se puede lograr cargando la LUT con los valores de salida apropiados (Figura 3).

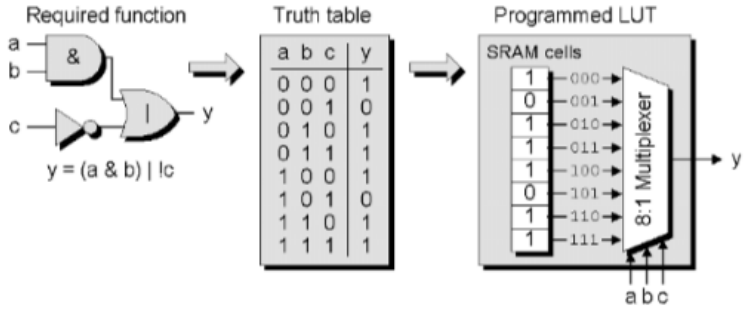

Figura 3. Configuración de un LUT

Hay que tener en cuenta que el LUT basado en un multiplexor 8: 1 de la Figura 3 se utiliza con fines de simplicidad. Normalmente son bastante más grandes.

El FPGA completo comprendía una gran cantidad de "islas" de bloques de lógica programable rodeadas por un "mar" de interconexiones programables (Figura 4).

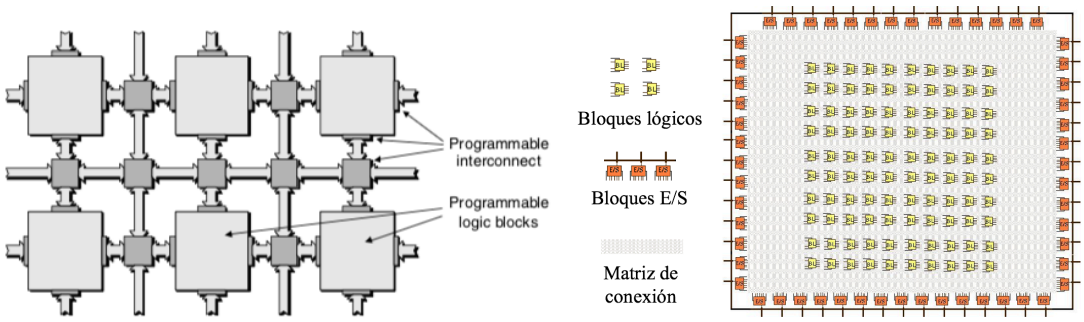

Figura 4. Arquitectura de una FPGA sencilla

Además de la interconexión local reflejada en la Figura 4, también habría rutas de interconexión globales (de alta velocidad) que podrían transportar señales a través del chip sin tener que atravesar múltiples elementos de conmutación locales.

El dispositivo también incluiría pines y pads de E/S primarios (no se muestran). Por medio de sus propias células SRAM, la interconexión podría programarse de modo que las entradas primarias al dispositivo estuvieran conectadas a las entradas de uno o más bloques lógicos programables, y las salidas de cualquier bloque lógico podrían usarse para conducir las entradas a cualquier otro bloque lógico, las salidas primarias del dispositivo, o ambos.

APARTADO 2.03 LA FAMILIA XILINX XC4000 FPGA

Los bloques lógicos programables en la familia Xilinx XC4000 de FPGA se denominan bloques lógicos configurables (configurable logic blocks, CLB).

II.03.1 CONFIGURABLE LOGIC BLOCKS, CLB

Los elementos programables más importantes del CLB son los generadores de función lógica F, G y H. Tanto F como G pueden realizar cualquier función lógica combinacional de sus cuatro entradas, y H puede realizar cualquier función lógica combinacional de sus tres entradas.

Como en los CPLD, las figuras trapezoidales en la Figura 5 representan multiplexores programables. Las salidas de F y G, así como otras entradas del CLB se pueden dirigir a las entradas de H mediante los multiplexores M1-M3, por lo que es posible realizar algunas funciones de más de cuatro entradas.

Con la programación adecuada de los multiplexores M7-M8 y M12-M13, las salidas de los generadores de función pueden dirigirse a las salidas del CLB X e Y, o pueden capturarse en flip-flops de tipo D disparados por flanco FF1 y FF2. Los flip-flops pueden usar el flanco ascendente o descendente de una entrada de reloj común, K, según lo seleccionado por los multiplexores M9 y M14. También pueden hacer uso de una señal de habilitación de reloj (EC) seleccionada por M10 y M15. Las fuentes de EC y otras tres señales internas se seleccionan de un conjunto de cuatro entradas misceláneas C1-C4 por multiplexores M3-M6 en la parte superior de la CLB.

Las salidas XQ e YQ del CLB llevan las salidas flip-flop fuera del CLB. Si no se usa un flip-flop en el CLB, el multiplexor M11 o M16 puede seleccionar XQ o YQ para que sea una "salida de derivación" que es simplemente una copia de una entrada CLB seleccionada por M4 oM6.

El bloque etiquetado como "control S/R" de cada flip-flop determina si el flip-flop se pone a SET o RESET en la configuración. También determina si el flip-flop responde a una señal global de SET / RESET (no mostrada) o a la señal SR del CLB seleccionada por el multiplexador MS.

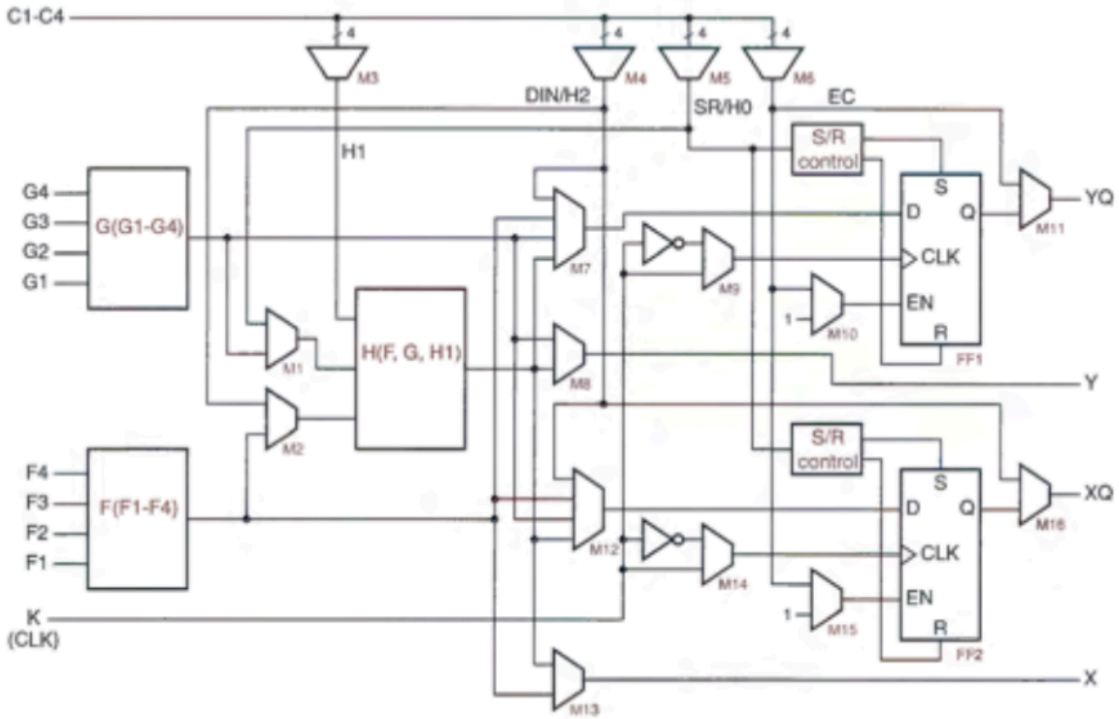

Figura 5. CLB

Los generadores de funciones F y G son realmente muy compactos y rápidos, 16x1SRAMs, y H es un 8x1SRAM. Cuando se usa un CLB para realizar la lógica, las tablas de verdad de las funciones lógicas F, G y H se cargan en SRAM en el tiempo de configuración desde una memoria externa de solo lectura. Los multiplexores programables en la Figura 5 también están controlados por "SRAM", en realidad latches D individuales que también se cargan en el momento de la configuración. Esta programación se realiza para todos los CLB en el FPGA.

II.03.2 BLOQUES DE ENTRADA/SALIDA

El XC4000 IOB (Input Output Block) tiene más controles "lógicos" que su primo en el CPLD XC9500. En particular, sus rutas de entrada y salida contienen flip-flops D activados por flanco seleccionables por los multiplexores M5-M7. Colocar los flip-flops de entrada y salida "cerca" de los pines E/S del dispositivo es especialmente útil en los FPGA. En la salida, las demoras relativamente largas de las salidas de flip-flop de CLB internas a los IOB pueden dificultar la conexión a sistemas síncronos externos a velocidades de reloj muy altas. En la entrada, largos retrasos desde los pines E/S a las entradas del flip-flop del CLB pueden dificultar cumplir con los requerimientos de los sistemas externos si las entradas externas

se conectan directamente a los flip-flops del CLB sin pasar por el IOB. Por supuesto, el uso de flip-flops IOB es posible solo si las especificaciones de la interfaz externa de la FPGA permiten el "pipeline" de las entradas y salidas.

Para las entradas canalizadas, el XC4000 IOB realmente va un paso más allá al proporcionar un elemento de retardo, seleccionable por M8, en serie con la entrada D del flip-flop de entrada FF2. El efecto de este elemento es retrasar la entrada D en relación con las copias internas del reloj del sistema del FPGA, lo que garantiza que la entrada tendrá un requisito de tiempo de espera cero con respecto al reloj del sistema externo. Este beneficio se produce a expensas de un mayor tiempo de configuración, por supuesto.

Los otros controles lógicos de la IOB son polaridad seleccionable, utilizando multiplexores M1-M4, para sus cuatro entradas que provienen de la matriz CLB a través de la interconexión programable. Estas entradas, OUT, T, OCLK e ICLKEN, son el bit de salida, su activación de tres estados, el reloj de salida y la habilitación del reloj de entrada, respectivamente.

Al igual que el XC9500 IOB, el XC4000 IOB también tiene controles analógicos. La velocidad de respuesta del controlador de salida es programable y se puede conectar una resistencia pull-up o pull-down al pin de E/S.

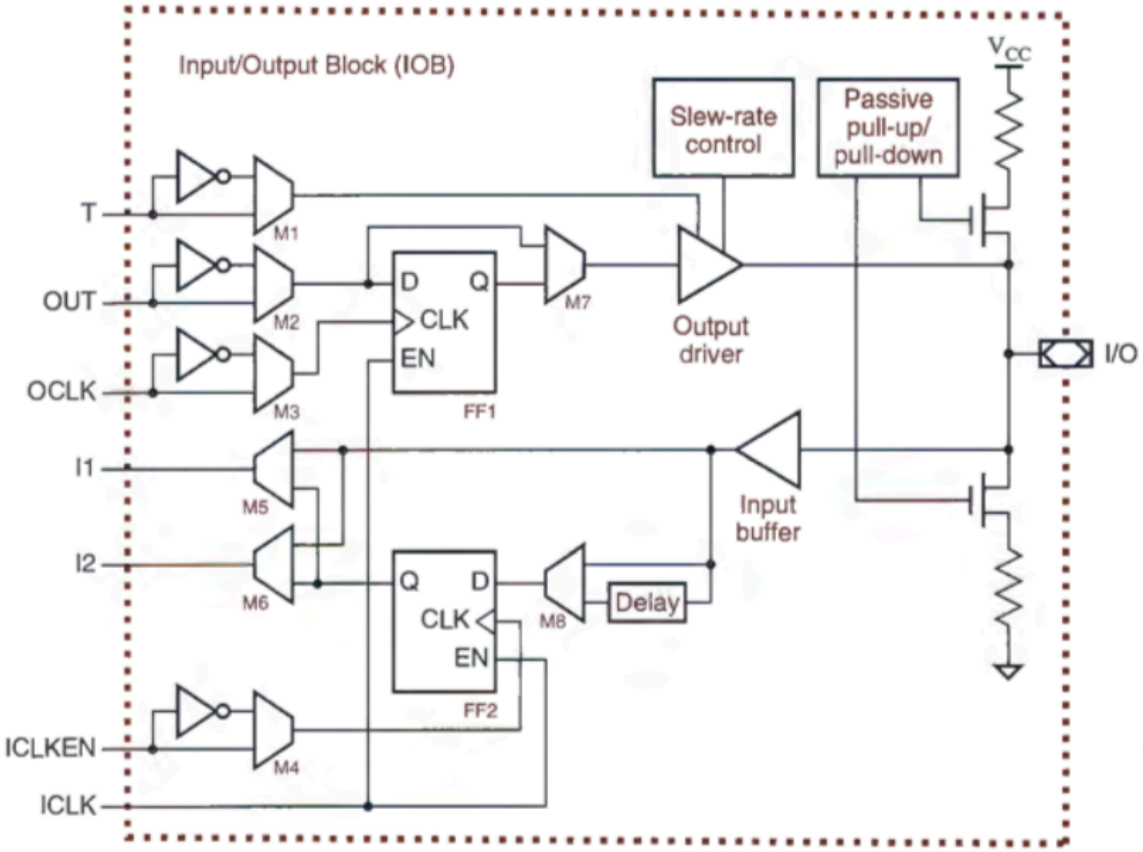

Figura 6. Bloque de E/S del XC4000.

II.03.3 MATRIZ DE CONEXIÓN (PROGRAMMABLE INTERCONNECT)

La arquitectura de interconexión programable del XC4000 es un ejemplo fascinante de una estructura que proporciona conectividad rica y simétrica en un área pequeña de silicio. La Figura 7 da un poco más de detalles del esquema de conexión del XC4000. Los cables en realidad no son "propiedad" de ningún CLB, sino que se crea una estructura como la que se muestra en la figura. Por ejemplo, 100 copias de esta figura forman el conjunto 10x 10 CLB de un XC4003.

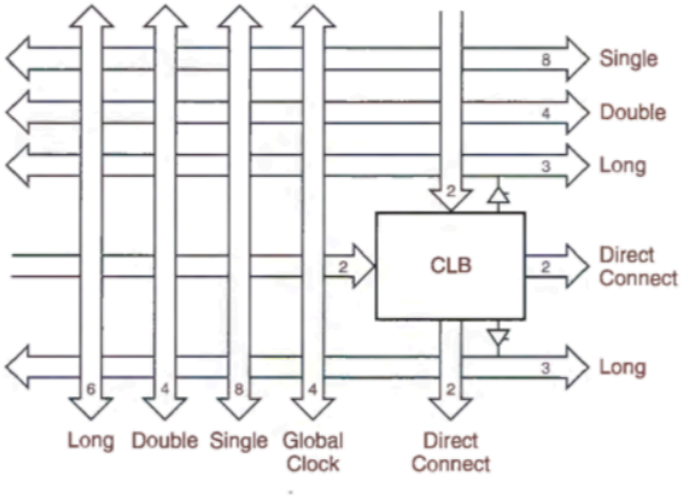

Figura 7. Estructura de interconexión general

El número en cada flecha indica la cantidad de cables en esa ruta de señal. Por lo tanto, puede ver que un CLB tiene dos cables (salidas) cada uno yendo a los CLB inmediatamente debajo y a la derecha del mismo. También se conecta a tres grupos de cables arriba, uno debajo y cuatro a la izquierda. Las señales en estos cables pueden fluir en cualquier dirección.

Es posible conectar un CLB a otro que esté a más de un salto utilizando cables "Single", pero tienen que pasar por un interruptor programable para cada salto, lo que agrega demora. Los cables en los grupos "Double" viajan más allá de dos CLB antes de dar con un interruptor, por lo que proporcionan demoras más cortas para las conexiones más largas. Para conexiones realmente largas, los grupos "Long" no pasan por ningún conmutador programable; en su lugar, viajan a lo largo de una fila o columna y son conducidos por conductores de tres estados cerca del CLB.

La Figura 8 muestra un CLB y cableado con mucho más detalle.

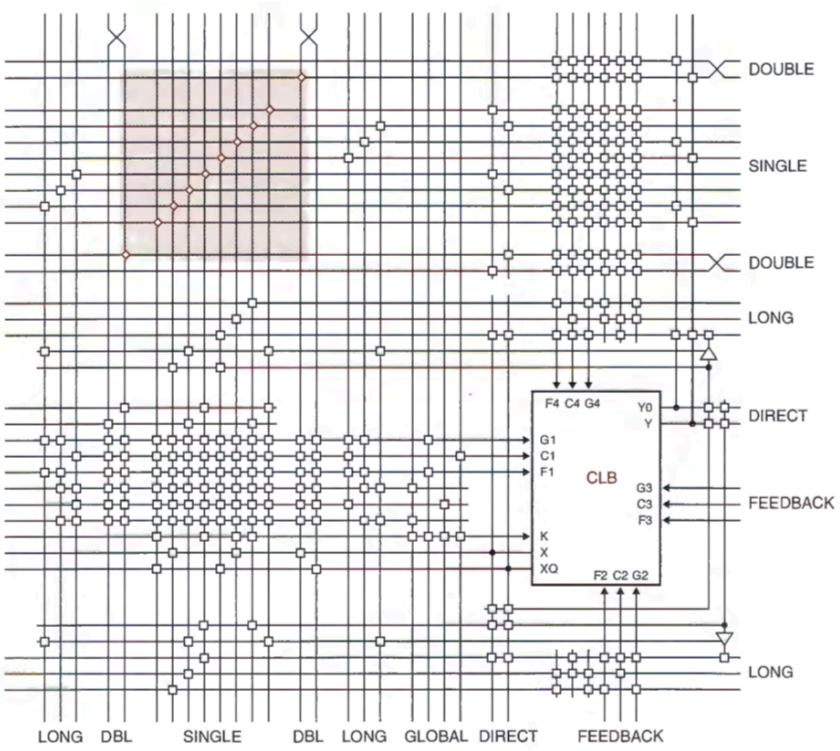

Figura 8. Estructura de interconexión en detalle

En la figura, el área sombreada en color se llama matriz de conmutación programable (PSM - programmable switch matrix).

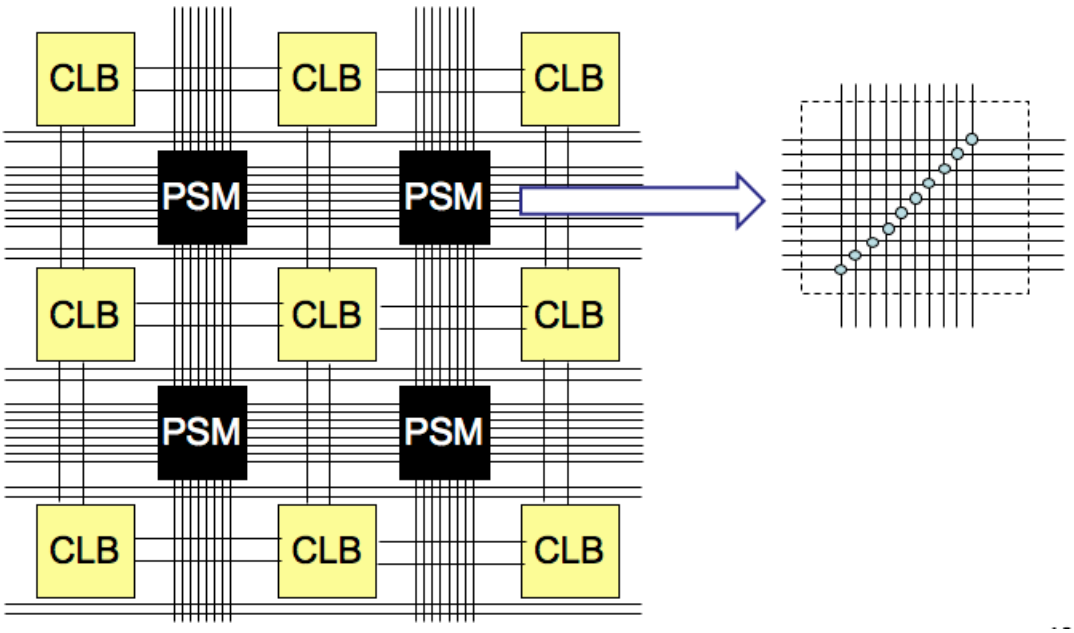

Figura 9. Estructura de interconexión

Cada uno de los diamantes en (a) es un elemento de conmutación programable (PSE) que puede conectar cualquier línea a cualquier otra, como se muestra en (b). Con cuatro líneas, hay seis posibles conexiones por pares como se muestra en (b), y el PSE tiene una puerta de transmisión para cada uno de ellos. Algunas, ninguna o todas las compuertas de transmisión

de un PSE pueden habilitarse, nuevamente, mediante bits de configuración almacenados en latches. Por lo tanto, son posibles muchos patrones de conexión diferentes, como se muestra en (c).

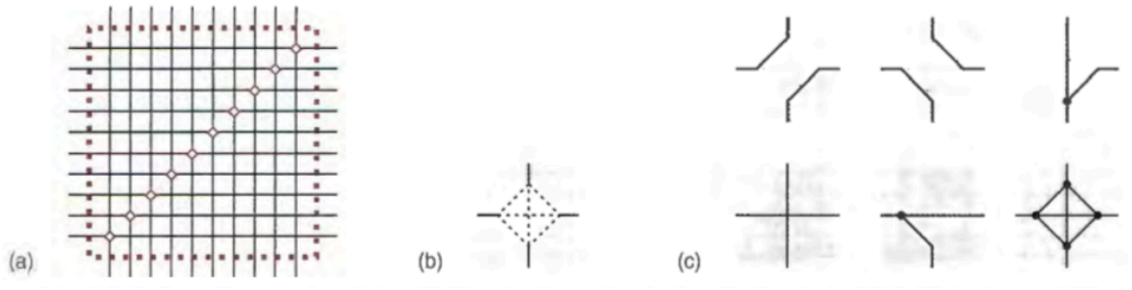

Figura 10. Configuraciones del PSE

APARTADO 2.04 ARQUITECTURAS AVANZADAS

La familia XC4000 es una arquitectura fuera de fabricación. Todas las familias de FPGA´s comparten en cierto grado esta arquitectura básica. Los nuevos diseños agregan elementos que permiten mayor capacidad de procesamiento o versatilidad. A continuación, se muestran algunos ejemplos de dichas mejoras.

II.04.1 MEMORIA RAM EN BLOQUE

Muchas aplicaciones requieren el uso de memoria, por lo que las FPGA incluyen trozos relativamente grandes de RAM incrustada llamada e-RAM o block RAM. Dependiendo de la arquitectura del componente, estos bloques pueden colocarse alrededor de la periferia del dispositivo, esparcidos por la cara del chip en relativo aislamiento u organizados en columnas, como se muestra en la Figura 11.

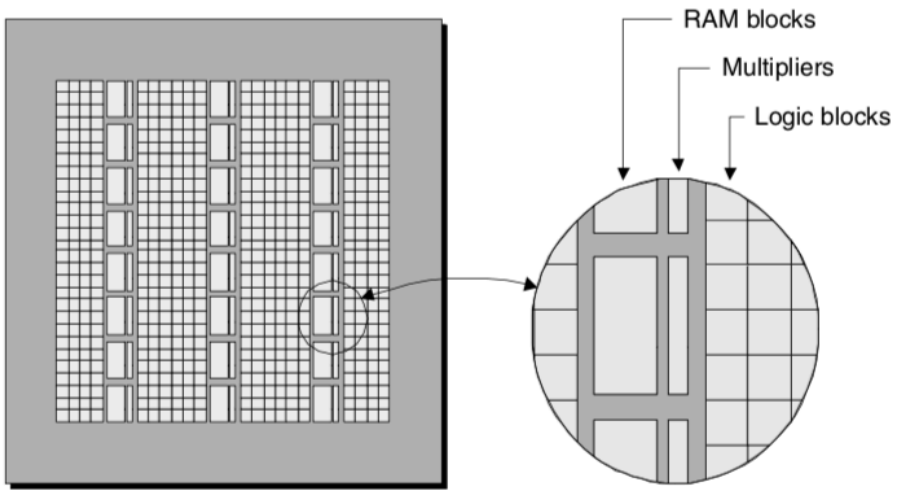

Figura 11. Configuraciones del PSE

II.04.2 MULTIPLICADORES EMBEBIDOS

Algunas funciones, como los multiplicadores, son intrínsecamente lentas si se implementan conectando juntos una gran cantidad de bloques lógicos programables. Dado que muchas aplicaciones requieren estas funciones, muchas FPGA incorporan bloques multiplicadores especiales cableados. Estos se encuentran típicamente muy cerca de los bloques de RAM incorporados introducidos en el punto anterior porque estas funciones a menudo se usan conjuntamente entre sí.

Del mismo modo, algunos FPGA ofrecen bloques sumadores dedicados. Una operación que es muy común en las aplicaciones de tipo DSP se llama multiply-and-accumulate (MAC). Como su nombre indica, esta función multiplica dos números y agrega el resultado a un total acumulado almacenado en un acumulador.

II.04.3 DSP SLICES

El concepto de utilizar FPGA para realizar operaciones DSP está bastante extendido actualmente. Por ello, las FPGA tienen segmentos DSP (DSP *slices*) para implementar funciones de procesamiento de señales. La operación de DSP que se usa más comúnmente es Multiply-Acumulate o MAC. Por tanto, un DSP slice esencialmente implementa una operación MAC.

Los segmentos DSP mejoran la velocidad y la eficiencia de muchas aplicaciones más allá del procesamiento digital de señales, como pueden ser *wide dynamic bus shifters*, generadores de direcciones de memoria, multiplexores de bus y registros de E/S mapeados en memoria.

A modo de ejemplo, Xilinx ha desarrollado diferentes versiones de DSP slices como el "DSP48".

La funcionalidad básica de la división DSP48E1 se muestra en la siguiente figura:

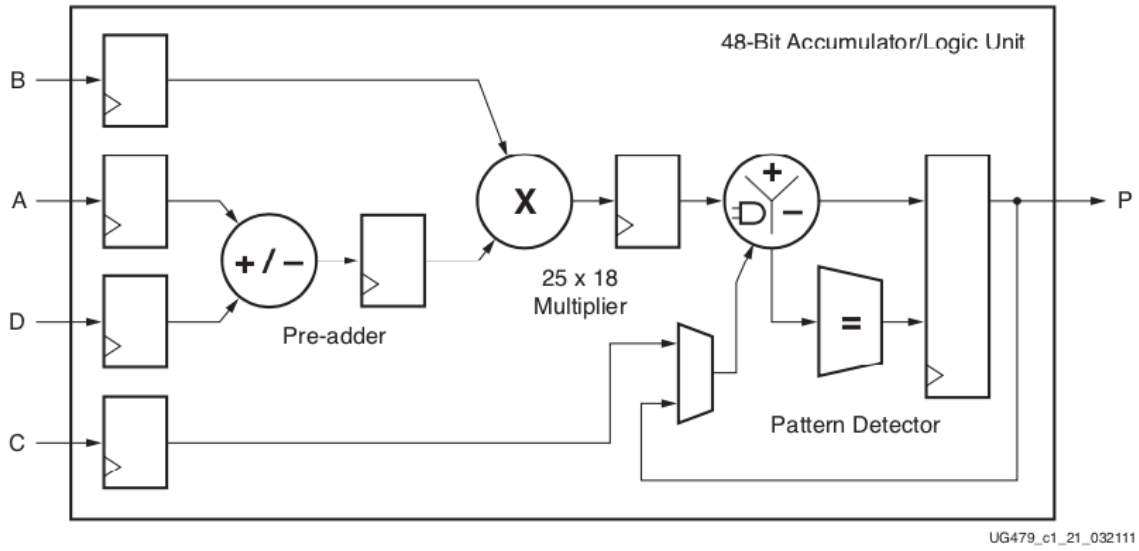

Figura 12. Estructura de un DSP Slice

Algunos aspectos destacados de la funcionalidad de DSP incluyen:

- Multiplicador de 25 × 18 (bits) en complemento a dos
- Acumulador de 48 bits:
  - Puede usarse como un contador sincronizado arriba/abajo
- Pre-sumador de ahorro de energía:
  - Optimiza las aplicaciones de filtro simétrico y reduce los requisitos del DSP slice
- Unidad aritmética de una sola instrucción-datos múltiples (SIMD):
  - Sumador/Restador/Acumulador dual de 24 bits o cuádruple de 12 bits
- Unidad lógica opcional:
  - Puede generar diez funciones lógicas diferentes sobre los dos operandos
- Detector de patrones:
  - Redondeo convergente o simétrico
  - Funciones lógicas de 96 bits de ancho cuando se utilizan junto con la unidad lógica
- Características avanzadas:
  - Canalización opcional y buses dedicados para conectar en cascada

Fuente: guía "7 Series DSP48E1 Slice" de Xilinx.

II.04.4 SEÑAL DE RELOJ

Todos los elementos síncronos dentro de una FPGA -por ejemplo, los registros configurados para actuar como flip-flops dentro de los bloques lógicos programables- necesitan ser accionados por una señal de reloj. Dicha señal de reloj típicamente se origina en el mundo exterior, entra en el FPGA a través de un pin de entrada de reloj especial, y luego se enruta a través del dispositivo y se conecta a los registros apropiados

Clock Tree

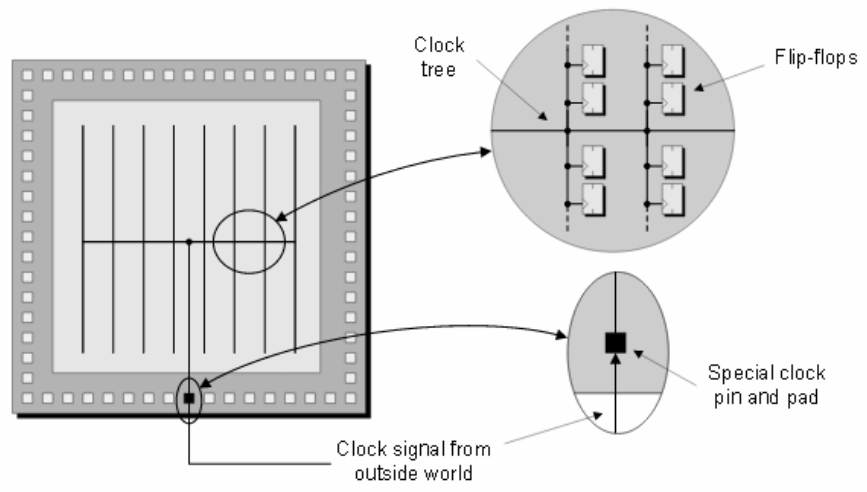

Figura 13. Configuraciones del PSE

La figura anterior muestra un "árbol de reloj" porque la señal del reloj principal se ramifica una y otra vez (se pueden considerar las flip-flops como las "hojas" al final de las ramas). Esta estructura se utiliza para garantizar que todos los flip-flops vean sus versiones de la señal del reloj lo más cerca posible. Si el reloj se distribuyera como una sola pista larga que manejara todas las flip-flops una tras otra, entonces el flip-flop más cercano al pin del reloj vería la señal del reloj mucho antes que la del final de la cadena. Esto se conoce como “skew”, y puede causar todo tipo de problemas (incluso cuando se utiliza un árbol de reloj, habrá una cierta cantidad de “skew” entre los registros en una rama y también entre las ramas).

El árbol de reloj se implementa utilizando pistas especiales y está separado de la interconexión programable de propósito general. El escenario mostrado arriba es realmente muy simplista. En realidad, hay varios pines de reloj disponibles (los pines de reloj no utilizados se pueden emplear como pines de E/S de propósito general), y hay varios dominios de reloj (árboles de reloj) dentro del dispositivo.

Clock Managers

En lugar de configurar un pin de reloj para conectarse directamente a un árbol de reloj interno, ese pin se puede usar para manejar una función especial de cableado (bloque) llamada un administrador de reloj (DCM -Digital Clock Manager-) que genera una serie de relojes hijo:

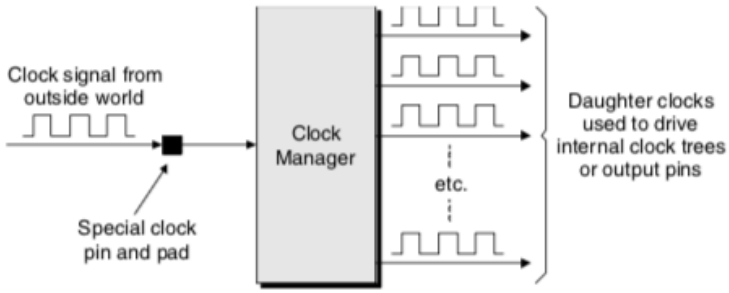

Figura 14. Relojes hijos

El DCM (Digital Clock Manager) es un circuito que maneja las señales de reloj de la FPGA. Se encarga de sincronizar todos los relojes y de:

- Eliminar el “jitter” (pulsos desiguales): supongamos que la señal de reloj tiene una frecuencia de 1 MHz. En un entorno ideal, cada flanco del reloj del mundo exterior llegaría exactamente una millonésima de segundo después de su predecesor. En el mundo real, sin embargo, los flancos del reloj pueden llegar un poco temprano o un poco tarde. Como una forma de visualizar este efecto -conocido como jitter- imagine si tuviéramos que superponer múltiples flancos uno encima del otro; el resultado sería un reloj "borroso"

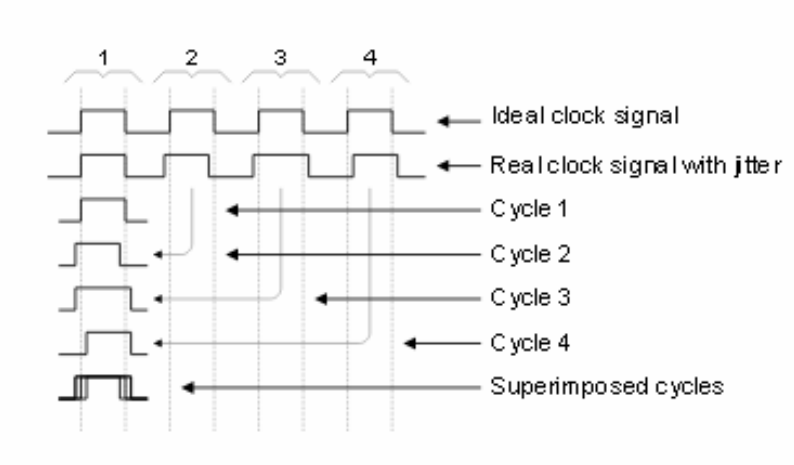

Figura 15. Skew

- Crear relojes múltiplos o divisores del original.

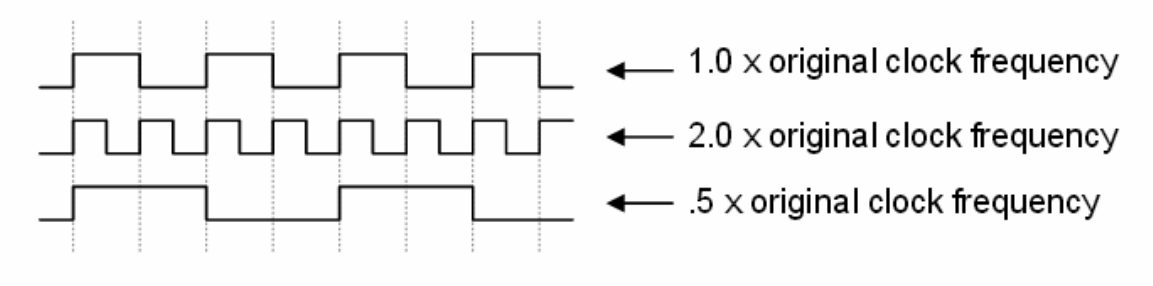

Figura 16. Relojes múltiplos o divisores del original

- Crear relojes desplazados en fase.

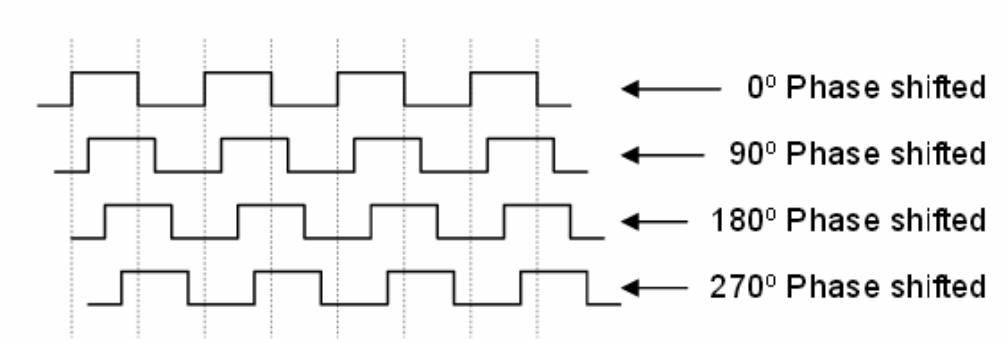

Figura 17. Relojes desplazados en fase

- Eliminar retardos mediante realimentación.

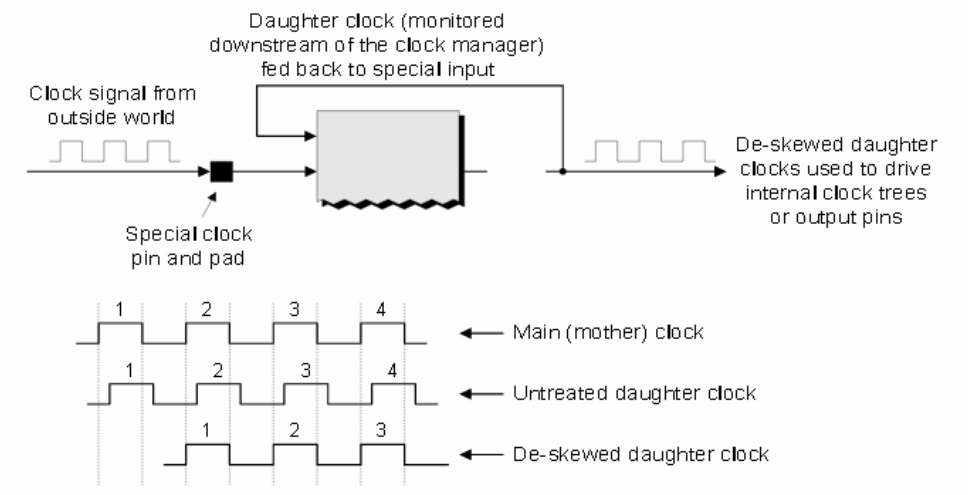

Figura 18. Eliminación de retardo

II.04.5 E/S CONFIGURABLES

Consideremos por un momento un producto electrónico desde la perspectiva de los arquitectos e ingenieros que diseñan la placa de circuito. Dependiendo de lo que están tratando de hacer, los dispositivos que están utilizando, el entorno en el que funcionará la placa, y así sucesivamente, seleccionarán un estándar particular que se utilizará para transferir señales de datos.

El problema es que existe una gran variedad de estos estándares, y sería doloroso tener que crear FPGAs especiales para acomodar cada variación. Por esta razón, las E/S de propósito general de un FPGA se pueden configurar para aceptar y generar señales que se ajusten a cualquier estándar requerido. Estas señales de E/S de propósito general se dividirán en varios bancos; supondremos que ocho de esos bancos están numerados del 0 al 7 (Figura 4-22). El punto interesante es que cada banco se puede configurar individualmente para admitir un estándar de E/S particular. Por lo tanto, además de permitir que el FPGA funcione con dispositivos que usan múltiples estándares de E/S, esto permite que el FPGA se use realmente para interactuar entre diferentes estándares de E/S.

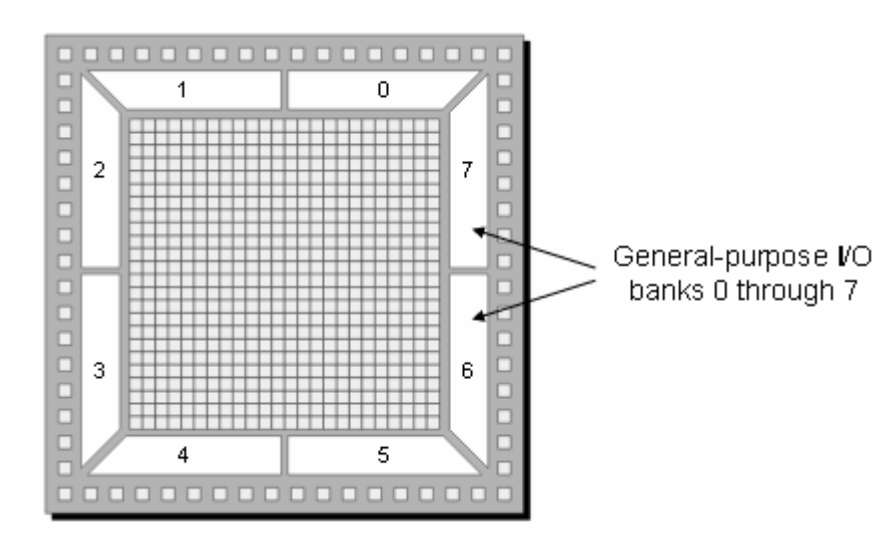

Figura 19. Configuración GPIO

II.04.6 TRANSCEPTORES GIGABIT

La forma tradicional de mover grandes cantidades de datos entre dispositivos es usar un bus, una colección de señales que llevan datos similares y realizan una función común. El problema es que esto requiere una gran cantidad de pines en el dispositivo y muchas pistas conectando los dispositivos juntos. El enrutamiento de estas pistas para que todas tengan la misma longitud e impedancia se vuelve cada vez más costoso a medida que las placas crecen en complejidad.

Por este motivo, los FPGA de gama alta actuales incluyen bloques transceptores gigabit cableados especiales. Estos bloques usan un par de señales diferenciales (lo que significa un par de señales que siempre llevan valores lógicos opuestos) para transmitir (TX) datos y otro par para recibir datos (RX).

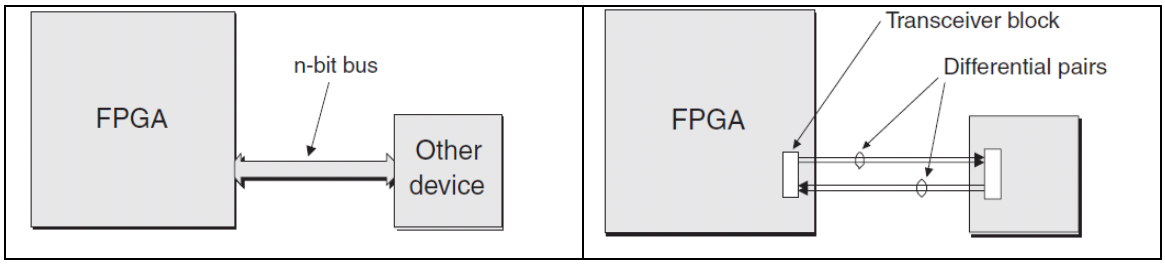

Figura 20. Transceptores Gogabit

Sección III.Artix-7

APARTADO 3.01 CLB

LUT (tabla de búsqueda) es una pequeña SRAM asíncrona que se utiliza para implementar la lógica combinacional, mientras que FF (Flip-Flop) es una celda de memoria de un solo bit utilizada para mantener el estado.

Las LUT son generalmente de solo lectura y su contenido solo se puede cambiar durante la configuración de FPGA. Pero en Xilinx FPGAs normalmente la mitad de los LUT se pueden escribir, por lo que se pueden usar para implementar muchas RAM pequeñas (la llamada "RAM distribuida"). Los flip-flop se pueden escribir y, de hecho, es su objetivo principal.

El SLICEM tiene la función RAM, pero el SLICEL es solo una LUT. La LUT / RAM en SLICEM se puede entender como de escritura, y el LUT en SLICEL es de solo lectura. El "LUT escribible" también se puede usar como un registro de desplazamiento con un múltiplex 16:1

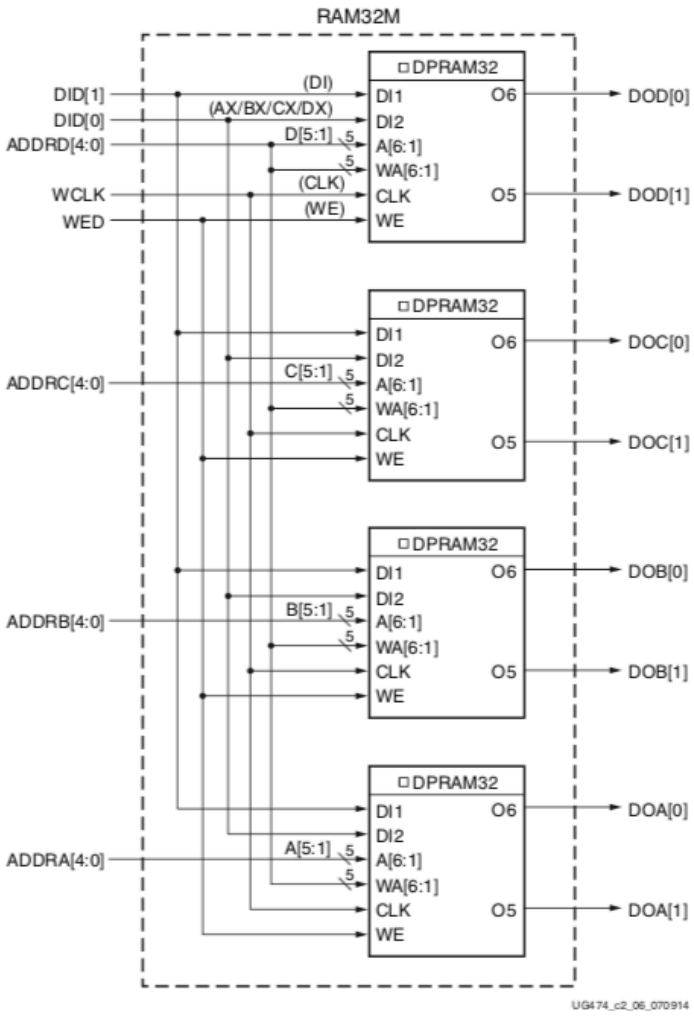

Figura 21. 32 X 2 Quad Port Distributed RAM (RAM32M)