

Arquitectura de la FPGA Spartan III de Xilinx

Las FPGA Spartan III de Xilinx ¡Error! No se encuentra el origen de la referencia. están conformadas por un conjunto de Bloques Lógicos Configurables (Configurable Logic Blocks: CLBs) rodeados por un perímetro de Bloques Programables de entrada/salida (Programmable Input/Output Blocks: IOBs). Estos elementos funcionales están interconectados por una jerarquía de canales de conexión (Routing Channels), la que incluye una red de baja capacitancia para la distribución de señales de reloj de alta frecuencia. Adicionalmente el dispositivo cuenta con 24 bloques de memoria RAM de 2Kbytes de doble puerto, cuyos anchos de buses son configurables, y con 12 bloques de multiplicadores dedicados de 18 X 18 bits.

Los cinco elementos funcionales programables que la componen son los siguientes:

- Bloques de entrada/salida (Input/Output Blocks – IOBs): Controlan el flujo de datos entre los pines de entrada/salida y la lógica interna del dispositivo. Soportan flujo bidireccional más operación tri-estado y un conjunto de estándares de voltaje e impedancia controlados de manera digital.
- Bloques Lógicos configurables (Configurable Logic Blocks – CLBs): Contienen Look-Up Tables basadas en tecnología RAM (LUTs) para implementar funciones lógicas y elementos de almacenamiento que pueden ser usados como *flip-flops* o como *latches*.
- Bloques de memoria RAM (Block RAM): Proveen almacenamiento de datos en bloques de 18 Kbits con dos puertos independientes cada uno.
- Bloques de multiplicación que aceptan dos números binarios de 18 bit como entrada y entregan uno de 36 bits.
- Administradores digitales de reloj (Digital Clock Managers – DCMs): Estos elementos proveen funciones digitales auto calibradas, las que se encargan de distribuir, retrasar arbitrariamente en pocos grados, desfasar en 90, 180, y 270 grados, dividir y multiplicar las señales de reloj de todo el circuito.

Los elementos descritos están organizados como se muestra en la Figura 1. Un anillo de IOBs rodea un arreglo regular de CLBs. Atraviesa este arreglo una columna de Bloques de memoria RAM, compuesta por varios bloques de 18 Kbit, cada uno de los cuales está asociado con un multiplicador dedicado. Los DCMs están colocados en los extremos de dichas columnas.

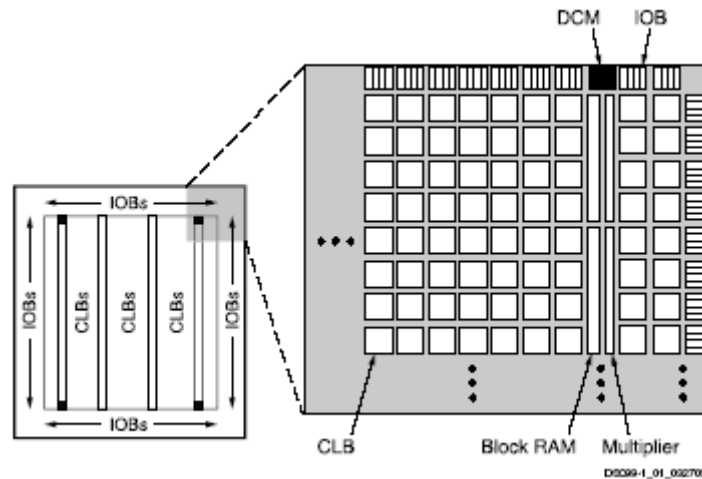


Figura 1: Arquitectura de la Spartan III

A continuación se hace una descripción más detallada de cada uno de los elementos funcionales de la FPGA, y luego se describe el proceso de configuración de la misma.

Bloques de entrada/salida IOB

Los bloques de entrada/salida (IOB) suministran una interfaz bidireccional programable entre un pin de entrada/salida y la lógica interna de la FPGA. Un diagrama simplificado de la estructura interna de un IOB aparece en la Figura 2. Hay tres rutas para señales en un IOB: la ruta de salida, la ruta de entrada y la ruta tri-estado. Cada ruta tiene su propio par de elementos de almacenamiento que pueden actuar tanto como registros o como latches. Las tres rutas principales son como sigue:

- La ruta de entrada, que lleva datos desde el *pad*, que está unido al pin del package, a través de un elemento de retardo opcional programable, directamente a la línea I. Después del elemento de retardo hay rutas alternativas a través de un par de elementos de almacenamiento hacia las líneas IQ1 e IQ2. Las tres salidas del IOB todas conducen a la lógica interna de la FPGA.
- La ruta de salida, que parte con las líneas O1 y O2, lleva datos desde la lógica interna de la FPGA, a través de un multiplexor y del *driver* tri-estado hacia el *pad* del IOB. En suma a esta ruta directa, el multiplexor da la opción de insertar un par de elementos de almacenamiento.
- La ruta tri-estado determina cuando el driver de salida está en alta impedancia. Las líneas T1 y T2 llevan datos desde la lógica interna a través de un multiplexor hacia el driver de salida. En suma a esta ruta directa, el multiplexor da la opción de entregar un par de elementos de almacenamiento.

Todas las rutas de señales que entran al IOB, incluidas aquellas asociadas con los elementos de almacenamiento tienen una opción de inversión. Cualquier inversor colocado (en la programación) en estas rutas es automáticamente absorbido dentro del IOB.

Hay tres pares de elementos de almacenamiento en cada IOB, un par para cada uno de las tres rutas. Es posible configurar cada uno de esos elementos como un *flip-flop* D

gatillado por flanco (FD) o como un *latch* sensible a nivel (LD). Estos elementos son controlados con la misma red de distribución de relojes que se utiliza para todo el sistema.

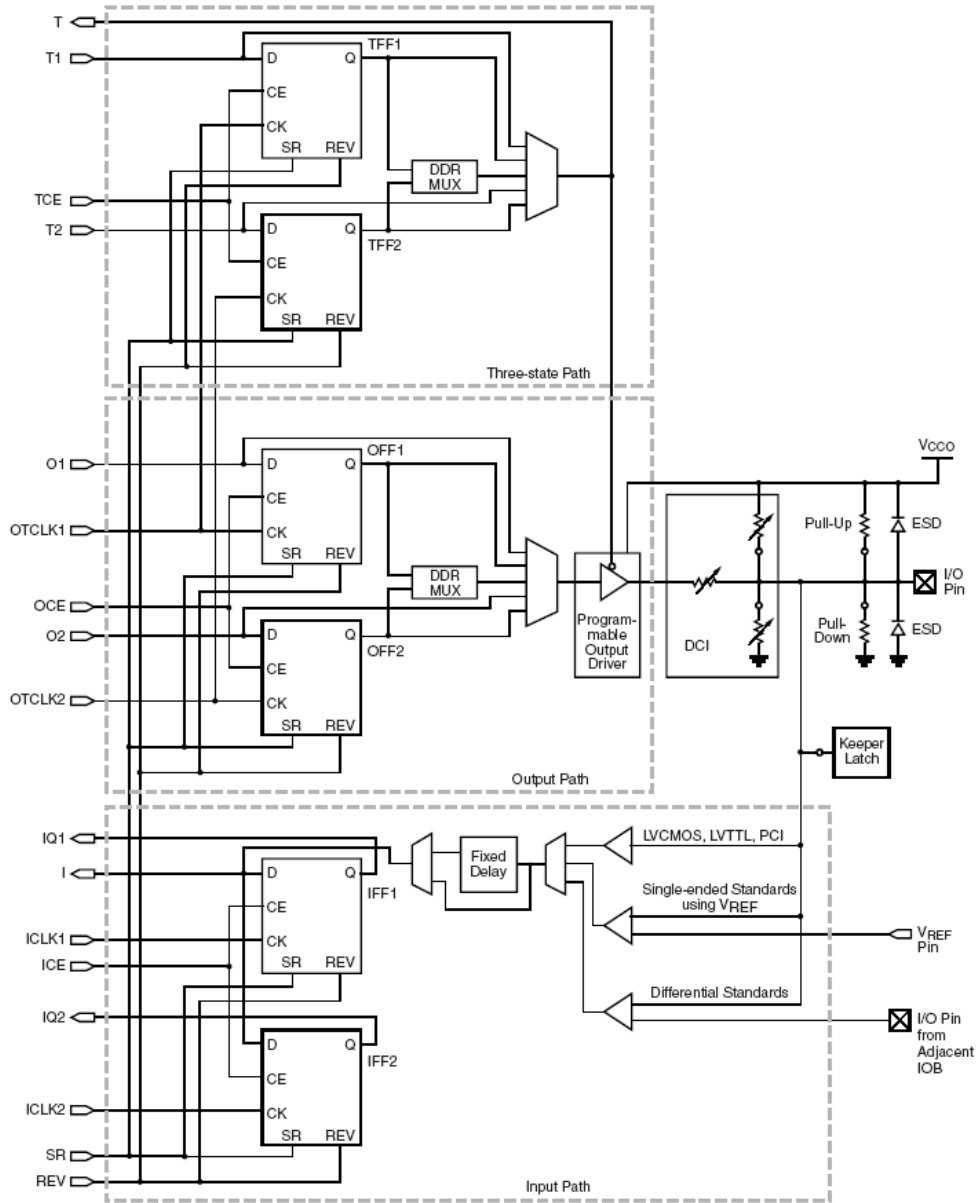


Figura 2: Diagrama simplificado de un IOB de la Spartan III

El par de elementos de almacenamiento tanto de la ruta de salida o de la del *driver* tri-estado pueden ser usados en conjunto, con un multiplexor especial para producir transmisión de doble tasa de datos (DDR). Esto se logra tomando datos sincronizados con el flanco de subida del reloj y convirtiéndolos en bits sincronizados tanto con el flanco de subida como con el de bajada. A esta combinación de dos registros y un multiplexor se le llama *flip flop* tipo D de doble tasa de datos (FDDR).

Cada IOB cuenta además con otros elementos, entre los cuales cuentan las resistencias de *Pull-Up* y de *Pull-Down*, que tienen el objetivo de establecer niveles altos o bajos respectivamente en las salidas de los IOBs que no están en uso; un circuito de retención (Keeper) del último nivel lógico que se mantiene, después de que todos los drivers han sido apagados, lo que es útil para cuidar que las líneas de un bus no floten, cuando los drivers conectados están en alta impedancia; un circuito de protección para descargas electro estáticas (protección ESD), que utiliza diodos de protección.

Finalmente cada IOB cuenta con un control para el *slew rate* y para la corriente de salida máxima. El primero otorga la posibilidad de elegir una tasa alta de cambio de nivel (con bajo *slew rate*) o una tasa máxima menor, pero con un control de transiente, para la utilización de los puertos en la integración a buses, donde al pasar de alta impedancia a un nivel de voltaje suele producirse transiciones inesperadas. El segundo entrega siete niveles deferentes de corrientes máximas tanto para el estándar CMOS como para el TTL, lo que permite adaptarse a dispositivos que necesitan mayores corrientes para su activación; en el caso del estándar LVCMOS a 2.5V el rango de corrientes es de 2 a 24 mA.(2, 4, 6, 8, 12, 16, 24 mA).

Los IOB soportan 17 estándares de señales de salida de terminación única y seis de señal diferencial; también cuentan con un sistema integrado, para coincidir con la impedancia de las líneas de transmisión que llegan a la FPGA, llamado Control Digital de Impedancia (DCI), el que permite elegir hasta 5 tipos diferentes de terminaciones, utilizando una red de resistencias internas que se ajustan en serie o en paralelo, dependiendo de las necesidades del estándar elegido.

Bloques de Lógica Configurable (CLB)

El bloque básico de la red que compone la FPGA es la *slice*. Existen dos tipo de slice, éstas se diferencian en algunos elementos, pero son muy parecidas, (ver más adelante). Luego estas *slices* se organizan en los bloques lógicos elementales, que son los que se describen a continuación.

Los Bloques de Lógica Configurable (CLBs) constituyen el recurso lógico principal para implementar circuitos síncronos o combinacionales. Cada CLB está compuesta de cuatro slices interconectadas entre si, tal como se muestra en la Figura 3.

Las cuatro slices que componen un CLB tienen los siguientes elementos en común: dos generadores de funciones lógicas, dos elementos de almacenamiento, multiplexores de función amplia, lógica de *carry* y compuertas aritméticas, tal como se muestra en la Figura 4. Los dos pares de *slices* usan estos elementos para entregar funciones lógicas y aritméticas de ROM. Además de lo anterior, el par de la izquierda soporta dos funciones adicionales: almacenamiento de datos usando RAM distribuida y corrimiento de datos con registros de 16 bits.

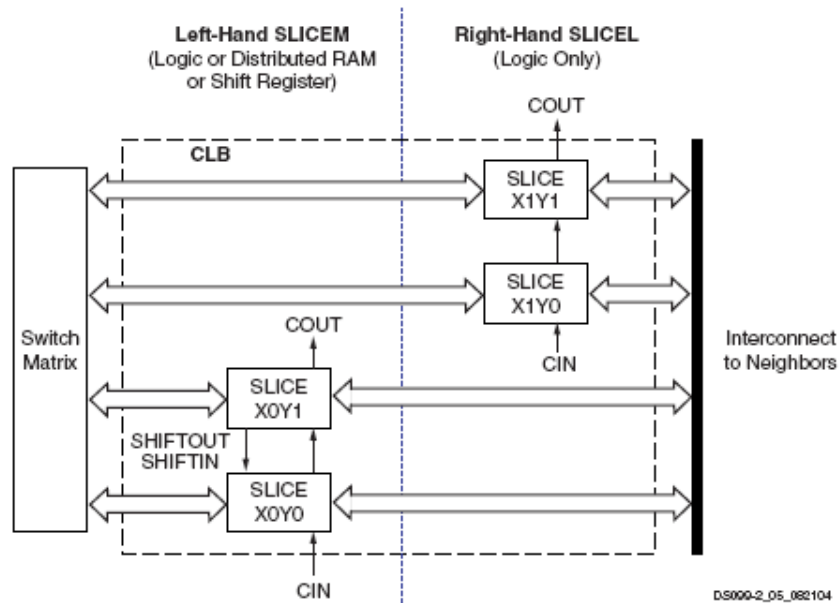


Figura 3: Arreglo de slices en un CLB

La Figura 4 es un diagrama de una slice del par del lado izquierdo, por lo tanto representa un súper conjunto de los elementos y conexiones que se encuentran en las slices.

El generador de funciones basado en RAM –también conocido como *Look-Up Table* (LUT)- es el recurso principal para implementar funciones lógicas dentro de la FPGA. Más aún, las LUTs en cada par de slices del lado izquierdo pueden ser configuradas como RAM distribuida o como un registro de corrimiento de 16 bits. Los generadores de funciones ubicados en las porciones superiores e inferiores de la slice son referidos como “G-LUT” y “F-LUT” respectivamente en la Figura 4.

El elemento de almacenamiento, el cual es programable tanto como un *flip flop* tipo D o como un *latch* sensible a nivel, provee un medio para sincronizar datos a una señal de reloj, entre otros usos. Estos elementos de almacenamiento, que se encuentran en las porciones superiores e inferiores de la slice son llamados “FFY” y “FFX”, respectivamente.

Los multiplexores de función amplia combinan las LUTs para permitir operaciones lógicas más complejas, cada slice tiene dos de éstos, en la Figura 4 corresponden a F5MUX y F1MUX.

La cadena de *carry*, en combinación con varias compuertas lógicas dedicadas, soportan implementaciones rápidas de operaciones matemáticas. La cadena de *carry* entra a la slice como CIN y sale como COUT. Cinco multiplexores controlan la cadena: CYINIT, CY0F y CYMUXF en la porción inferior, así como CY0G y CYMUXG en la porción superior. La lógica aritmética dedicada incluye compuertas XOR y AND en cada porción de la slice.

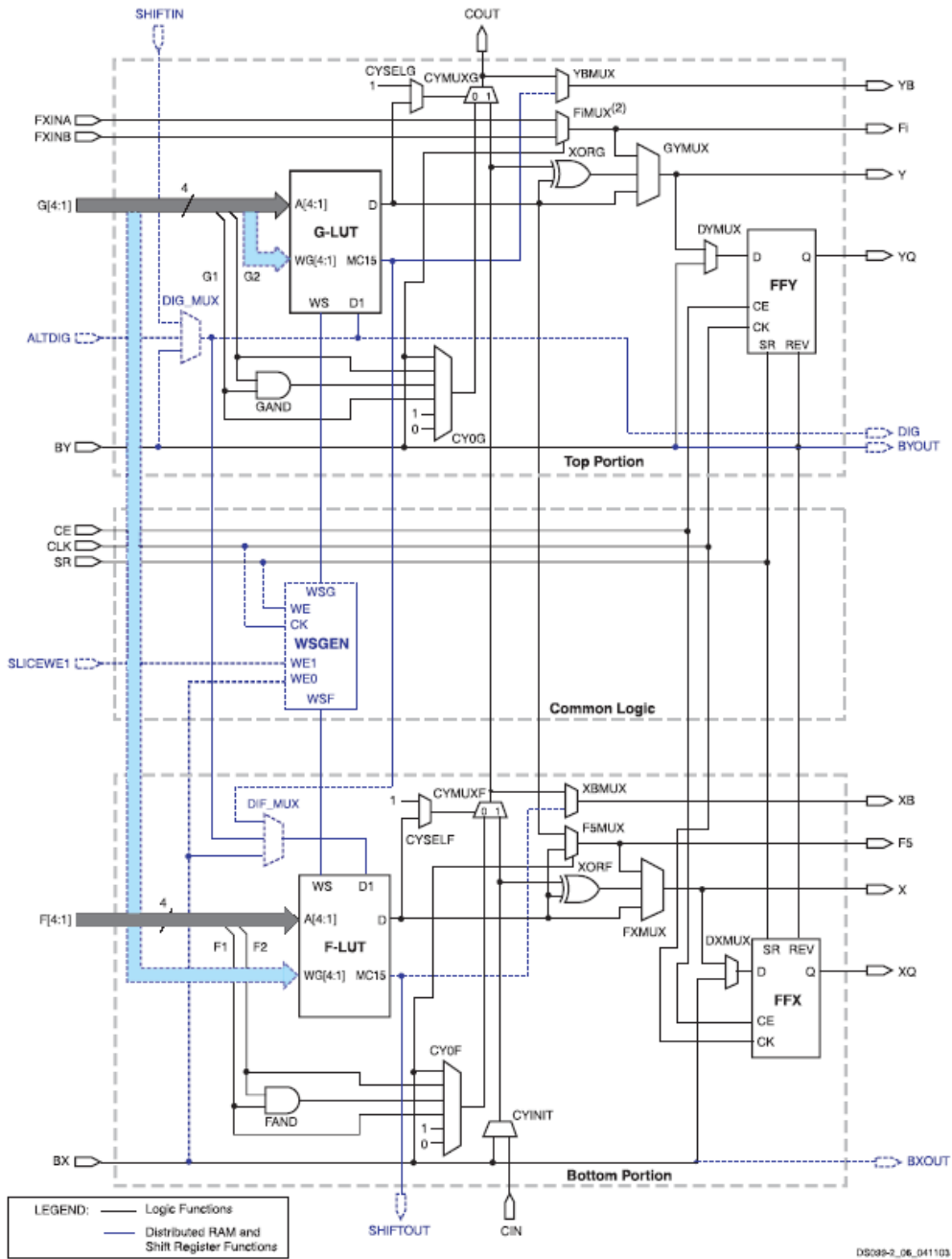


Figura 4: Diagrama simplificado de una slice del lado izquierdo de un CLB

Con un rol central en la operación de cada *slice* se encuentran dos rutas de datos casi idénticas. Para la descripción que prosigue se usan los nombres de la parte inferior de la Figura 4. La ruta básica tiene su origen en la matriz de *switches* de interconexión colocada fuera del CLB. Cuatro líneas, F1 a F4 entran en la slice y se conectan directamente a la LUT. Una vez dentro de la slice, la ruta de los 4 bits inferiores pasa a través de un generador de funciones F que realiza operaciones lógicas. La ruta de salida del generador de funciones, D, ofrece cinco posibles rutas posibles:

- Salir de la *slice* por la línea X y volver a interconectarse.

- Dentro de la *slice*, X sirve como entrada al DXMUX que alimenta la entrada de datos D, correspondiente al elemento de almacenamiento FFY. La salida Q de este elemento maneja la ruta XQ que sale de la *slice*.
- Controlar el multiplexor CYMUXF de la cadena de *carry*.
- Con la cadena de *carry*, servir como una entrada a la compuerta XORF, que realiza operaciones aritméticas y produce el resultado en X.
- Manejar el multiplexor F5MUX para implementar funciones lógicas más anchas que 4 bits. Las salidas D de los F-LUT y G-LUT sirven de entradas de datos para este multiplexor.

En suma a los caminos lógicos principales descritos recién, existen dos rutas de *bypass* que entran a la *slice* como BX y BY. Una vez dentro de la FPGA, BX en la parte de debajo de la *slice* (o BY en la parte superior) puede tomar cualquiera de varias ramas diferentes:

- Hacer *bypass* de la LUT y del elemento de almacenamiento, luego salir de la *slice* como BXOUT y volver a interconectarse.
- Hacer *bypass* a la LUT, y luego pasar a través del elemento de almacenamiento, para luego salir como XQ.
- Controlar el multiplexor F5MUX.
- Servir como una entrada a la cadena de *carry* vía los multiplexores.
- Manejar la entrada DI de la LUT.
- BY puede controlar la entrada REV de FFY y de FFX.
- Finalmente, el multiplexor DIG_MUX puede derivar la ruta BY hacia la línea DIG que sale de la *slice*.

Cada una de las dos LUTs (F y G) de una *slice* tiene cuatro entradas lógicas (A1–A4) y una única salida D. Esto permite programar cualquier operación lógica booleana de cuatro variables en este dispositivo. Además, los multiplexores de función amplia pueden usarse para combinar LUTs dentro del mismo CLB o incluso a través de diferentes CLBs, haciendo posible funciones con mayor número de variables.

Las LUT de ambos pares de *slices* dentro de un CLB no sólo soportan las funciones descritas, si no que también pueden funcionar como ROM (Read Only Memory) con datos inicializados al momento de configurar la FPGA. Las LUTs del lado izquierdo de cada CLB soportan además dos funciones adicionales: primero, es posible programarlas como RAM distribuida, lo que permite contar con espacios de memoria de 16 bits en cualquier parte de la topología de la FPGA. Segundo, es posible programar una de estas LUTs como un registro de desplazamiento de 16 bits, con lo que se pueden producir retardos de hasta 16 bits o combinaciones de varias LUTs pueden producirlos de cualquier largo de bits.

Bloques dedicados de memoria RAM

La Spartan III tiene 24 bloques de 18 Kbits de memoria RAM. El ancho del bus de datos versus el de direcciones (relación de aspecto) de cada bloque es configurable y se

puede combinar varios de éstos para formar memorias más anchas o de mayor profundidad.

Tal como se muestra en la Figura 5, los bloques de RAM tienen una estructura de doble puerto. Dos puertos idénticos llamados A y B permiten acceso independiente al mismo rango de memoria, que tiene una capacidad máxima de 18.432 bits – o 16.384 cuando no se usan las líneas de paridad. Cada puerto tiene su propio set de líneas de control, de datos y de reloj para las operaciones síncronas de lectura y escritura. Estas operaciones tienen lugar de manera totalmente independiente en cada uno de los puertos.

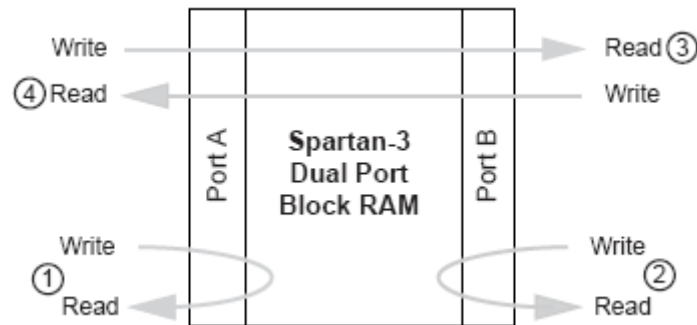


Figura 5: Diagrama de un bloque de RAM dedicado de la Spartan III

Multiplicadores dedicados

La Spartan III provee multiplicadores embebidos que aceptan palabras de 18 bits como entrada y entregan productos de 36 bits. Los buses de entrada de estos multiplicadores aceptan datos en complemento dos (tanto 18 bits con signo, como 17 bits sin signo). Para cada bloque de RAM hay un multiplicador inmediatamente colocado y conectado; dicha proximidad permite manejo eficiente de los datos.

Digital Clock Manager (DCM) y red de distribución de relojes

La Spartan III tiene 4 bloques para el control de todos los aspectos relacionados con la frecuencia, la fase y el *skew* de la red de relojes de la FPGA. Cada DCM tiene cuatro componentes funcionales: El Delay-Locked Loop (DLL), El Sintetizador Digital de Frecuencia (DFS) y el Desplazador de fase (PS). Además incluye cierta lógica para *status*.

La Figura 6 muestra un diagrama de bloques de este elemento funcional de la FPGA.

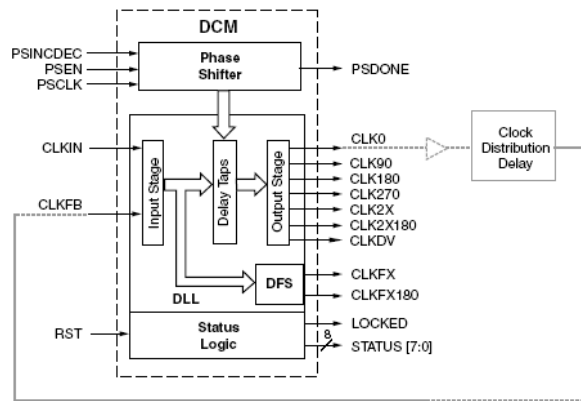


Figura 6: Diagrama de bloques de uno de los cuatro DCMs de la Spartan III

El DCM realiza tres funciones principales:

- Eliminación de *skew* de reloj: El concepto de *skew* describe el grado al cual las señales de reloj pueden, bajo circunstancias normales, desviarse del alineamiento de la fase cero. Ello ocurre cuando pequeñas diferencias en los retardos de las rutas causan que la señal de reloj llegue a diferentes puntos del circuito en tiempos diferentes. Este *skew* de reloj puede incrementar los requerimientos de *set-up time* y de *hold time*, lo que puede perjudicar el desempeño de aplicaciones de alta frecuencia. El DCM elimina el *skew* de reloj alineando la salida de la señal de reloj que genera con otra versión de la misma señal que es retroalimentada. Como resultado se establece una relación de cero desfase entre ambas señales.
- Síntesis de frecuencia: Provisto de una señal de reloj de entrada, el DCM puede generar diferentes relojes de salida. Ello se logra multiplicando y/o dividiendo la frecuencia del reloj de entrada.
- Corrimiento de fase: El DCM puede producir desfases controlados de la señal de reloj de entrada y producir con ello relojes de salida con diferentes fases.

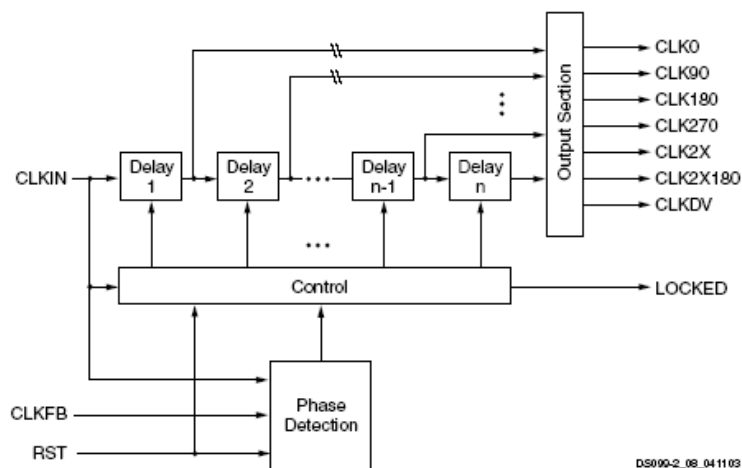


Figura 7: Diagrama funcional del Delay-Locked Loop (DLL)

El DLL tiene como principal función eliminar el *skew* de reloj. La ruta principal del DLL consiste en una etapa de entrada, seguida por una serie de elementos de retardo discreto o *taps*, los cuales conducen a una etapa de salida. Esta ruta, junto con lógica

para detección de fase y control conforman un sistema completo con retroalimentación, tal como se muestra en la Figura 7.

La señal CLK0 es entregada a la red de distribución de señales de reloj de la FPGA, que sincroniza todo los registros del circuito que ha sido configurado. Estos registros pueden ser tanto internos como externos a la FPGA. Luego de pasar por dicha red, la señal de reloj retorna al DLL a través de la entrada CLKFB. El bloque de control del DLL mide el error de fase entre ambas señales, que es una medida del *skew* de reloj que toda la red introduce. El bloque de control activa el número apropiado de elementos de retardo para cancelar el *skew* de reloj. Una vez que se ha eliminado el desfase, se eleva la señal LOCKED, que indica la puesta en fase del reloj con respecto a la retroalimentación.

Las señales de reloj tienen una red dedicada especial para su distribución. Esta red tiene ocho entradas globales, por medio de *buffers*. La red tiene baja capacitancia y produce muy bajo *skew* de reloj, lo que la hace adecuada para conducir señales de alta frecuencia. Tal como se muestra en la Figura 8, las entradas GCLK0 a GCLK3 están puestas en la parte inferior de la oblea de la FPGA, mientras que las entradas GCLK4 a GCLK7 están colocadas en la parte superior. Se puede conducir cualquiera de dichas entradas hacia cada uno de los CLBs, por medio de las líneas principales, que se observan en negro grueso en la Figura 8. Las líneas más delgadas representan líneas que conducen hacia los elementos síncronos de cada una de las *slices* de los CLBs.

Las entradas a la red se distribuyen a través de 4 multiplexores 2-1 a cada lado de la red, los que también conducen las señales provenientes de los DCMs. Con el propósito de minimizar la disipación de potencia dinámica en la red de distribución de relojes, el *software* de síntesis automáticamente deshabilita aquellas líneas que no son utilizadas en el diseño.

Esta red de distribución de señales de reloj es completamente independiente de la malla de interconexiones entre CLBs que se describe en el siguiente punto (II.4.3.6).

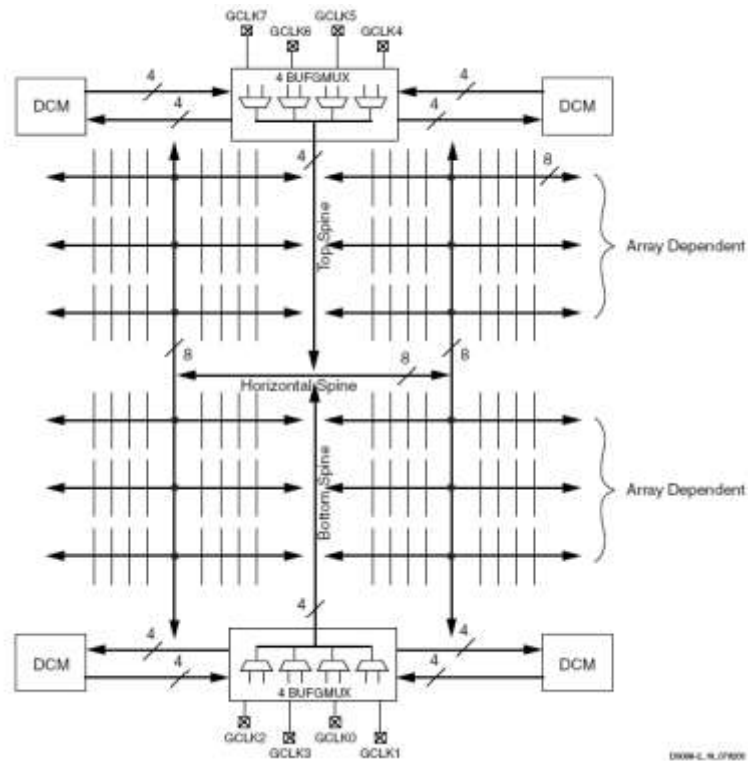


Figura 8: Red de distribución de señales de reloj de la Spartan III

Red de interconexiones de la FPGA

La red de interconexión conduce las señales entre varios elementos funcionales de la Spartan III. Hay cuatro tipos de interconexiones: *Long lines*, *Hex lines*, *Double lines* y *Direct lines*.

Long lines son aquellas que conectan una salida de cada seis CLBs (Figura 9a). Debido a su baja capacitancia, estas líneas son adecuadas para conducir señales de alta frecuencia. Si las ocho entradas para las redes de reloj están ocupadas, estas líneas son adecuadas como alternativa. *Hex lines* son las que conectan una salida de cada tres CLBs (Figura 9b). Son líneas que ofrecen mayor conectividad que las anteriores, pero un poco menos de capacidad en alta frecuencia. Las *Double lines* conectan todos los otros CLBs (Figura 9c), lo que las hace conexiones más flexibles. Las *Direct lines* entregan conexiones directas de cada CLB hacia cada uno de sus ocho vecinos (figura 8d). Estas líneas son usadas más a menudo para conducir una señal proveniente de un CLB de origen hacia una *Double line*, *Hex line* o *Long line* y desde esa ruta larga hacia otra *Direct line* que llevará la señal hacia el CLB de destino.

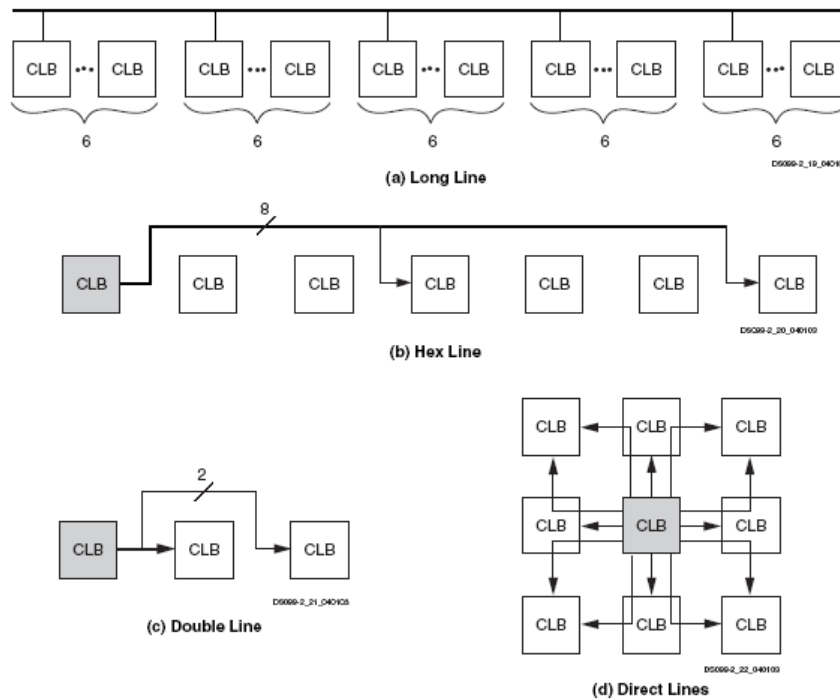


Figura 9: Tipos de interconexiones entre CLBs en la Spartan III

Proceso de configuración de la FPGA Spartan III

La FPGA Spartan III se programa por medio de la carga de los datos de configuración en celdas de memoria estática, las que colectivamente controlan todos los elementos funcionales y los recursos de interconexión. Luego de aplicar alimentación a la oblea, se escribe la trama de configuración en dicha memoria utilizando uno de los siguientes modos: Maestro - Paralelo, Esclavo - Paralelo, Maestro - Serial, Esclavo - Serial o *Boundary-Scan* (JTAG). Estos modos difieren en el origen del reloj (proviene de la FPGA en los modos Maestro y es externo en los modos Esclavo), y en la forma en que se escriben los datos, por lo que los modos paralelos son más rápidos.

El modo *Boundary-Scan* utiliza pines dedicados de la FPGA y cumple con los estándares IEEE 1149.1 *Test Access Port* e IEEE 1532 para dispositivos *In-System Configurable* (ISC). Este modo está siempre disponible en la FPGA y al activarlo se desactivan los otros modos ya mencionados.

El proceso de configuración de la FPGA ocurre en tres etapas. Primero la memoria interna de configuración es borrada. Luego los datos de configuración son cargados en dicha memoria, y finalmente la lógica es activada por un proceso de partida.

Flujo de diseño para la configuración de la FPGA

El flujo de diseño para generar la configuración de una FPGA está compuesto principalmente por cuatro etapas: diseño lógico, síntesis, implementación y generación del archivo de salida. En el caso de

las FPGA de Xilinx existe un paquete de *software* que reúne herramientas para llevar a cabo cada una de estas etapas, subetapas y procesos de simulación en varios niveles de profundidad. Esta utilidad se llama Xilinx Integrated Software Environment (ISE) 7.1i. ¡Error! No se encuentra el origen de la referencia. y está disponible, en una versión bastante completa y gratuita, con el nombre de WebPack ISE 7.1i. ¡Error! No se encuentra el origen de la referencia..

El diseño lógico se realiza mediante un lenguaje de descripción de hardware tal como VHDL o Verilog (ver punto II.4.5). En este trabajo se ha utilizado el primero para describir cada uno de los módulos del diseño, y también se ha utilizado una herramienta de más alto nivel, que de manera gráfica permite juntar los distintos bloques en un esquemático y unirlos con buses y conexiones unitarias. Esta herramienta es parte de del paquete ISE y se vale de la característica jerárquica del mismo VHDL, mediante la cual se puede crear componentes y unirlos usando recursos del lenguaje.

Una vez descrito el sistema, la siguiente etapa consiste en sintetizarlo. Este proceso se realiza de forma automática y sigue directivas de configuración, en las que se determinan los algoritmos preferidos de síntesis. La salida de éste es una *netlist*, que es un archivo que contiene una lista de conexiones, una lista de instancias y para cada instancia, una lista de señales conectadas a los terminales de dicha instancia. Además contiene información de atributos del diseño. En este caso la *netlist* es una descripción a nivel de compuertas lógicas del sistema descrito. La herramienta que realiza este proceso en el caso del paquete ISE se llama Xilinx Synthesis Tool (XST) ¡Error! No se encuentra el origen de la referencia. y su netlist es un archivo de formato NGC (Native Generic Circuit).

La *netlist* de formato NGC es la entrada para el proceso de implementación, el cual se subdivide en tres etapas: *Translate*, *Mapping* y *Place and Route*. En la primera el archivo NGC es convertido a un formato estándar llamado NGD, por una herramienta llamada NGDBuild, que no sólo acepta archivos de salida del sintetizador XST, si no que también otros formatos provenientes de otros sintetizadores alternativos. En la segunda etapa se mapea el diseño lógico contenido en el archivo NGD, en los componentes físicos reales, con que cuentan las *slices* de la FPGA. En la tercera, se determina la topología de colocación y de interconexión de los elementos ya mapeados. Esta etapa es un proceso iterativo, que tiene alto costo computacional y puede demorar decenas de minutos.

Finalmente, en la etapa de generación, otra herramienta genera un archivo de configuración, el que es descargado a la memoria de la FPGA y que contiene la trama de bits que produce la configuración adecuada.