UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO

MAESTRÍA EN TELECOMUNICACIONES

TÍTULO DE LA TESIS:

DEMODULACIÓN PSK Y FSK: IMPLEMENTACIÓN EN FPGAs

Previa la obtención del Grado Académico de Magíster en Telecomunicaciones

ELABORADO POR:
Ing. Felipe Andrés Zambrano García

Guayaquil, a los 10 días del mes Octubreaño 2014

SISTEMA DE POSGRADO

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el MagísterFelipe Andrés Zambrano Garcíacomo requerimiento parcial para la obtención del Grado Académico de Magíster en Telecomunicaciones.

Guayaquil, a los 10 días del mes Octubre año 2014

DIRECTOR DE TESIS

_____

MsC. Edwin Palacios Meléndez

REVISORES:

_____

MsC. Néstor Zamora Cedeño.

_____

MsC. Luis Córdova Rivadeneira

DIRECTOR DEL PROGRAMA

_____

MsC. Manuel Romero Paz

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO

DECLARACIÓN DE RESPONSABILIDAD

YO, FELIPE ANDRÉS ZAMBRANO GARCÍA

DECLARÓ QUE:

La tesis “DEMODULACIÓN PSK Y FSK: IMPLEMENTACIÓN EN FPGAs”, previa a la obtención del grado Académico de Magíster, ha sido desarrollada en base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes. Consecuentemente este trabajo es de nuestra total autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico de la tesis del Grado Académico en mención.

Guayaquil, a los 10 días del mes Octubre año 2014

EL AUTOR

_____

Ing. Felipe Andrés Zambrano García

SISTEMA DE POSGRADO

AUTORIZACIÓN

YO, FELIPE ANDRÉS ZAMBRANO GARCÍA

Autorizamos a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución de la Tesis de Maestría titulada: "DEMODULACIÓN PSK Y FSK: IMPLEMENTACIÓN EN FPGAs", cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 10 días del mes Octubre año 2014

EL AUTOR

_____

Ing. Felipe Andrés Zambrano García

Dedicatoria

Dedico este trabajo a Dios, por permitirme vivir este momento tan especial.

A mis padres por haberme enseñado a luchar por alcanzar todos los objetivos que me he propuesto.

A Sandra por ser mi apoyo incondicional en todo lo que hago y por ser una parte muy importante en mi vida.

Y a todos mis seres queridos que siempre han estado presente para darme su apoyo.

Agradecimientos

Agradezco a Dios por haberme guiado a lo largo de mi carrera.

Le doy gracias a mis padres por haberme inculcado valores, y por la oportunidad que me dieron para poder acceder a una excelente educación.

A mi tutor de tesis MSC. Edwin Palacios Meléndez, por su guía a lo largo de la elaboración de este trabajo.

Al director de la Maestría MSC Manuel Romero Paz, porsu constante apoyo.

Y a todas las personas que colaboraron de una u otra manera, hago extensivos mis más sinceros agradecimientos.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

Capítulo 2:

Capítulo 3:

Capítulo 4:

ÍNDICE DE TABLAS

Capítulo 2:Estado del Arte del Procesamiento de Señales ECG

Capítulo 3:Adquisición, almacenamiento, y transmisión del ECG

Resumen

El presente trabajo denominado “DEMODULACIÓN PSK Y FSK: IMPLEMENTACIÓN EN FPGAs”, se logró demostrar que a través de la programación de arreglos de compuertas (Field ProgrammableGateArray) se pueden realizar sistemas de modulación y demodulación digital como un campo específico de las Telecomunicaciones. Este trabajo pretende promover el uso de dispositivos electrónicos robustos como los FPGAs que permiten emular aplicaciones en diversos campos de las carreras de ingeniería de la Facultad de Educación Técnica para el Desarrollo.

En el capítulo 1 se realiza la introducción del trabajo de intervención, en el capítulo 2 se describe el estado del arte de las plataformas FPGAs, en el capítulo 3 se realiza el diseño de los demoduladores PSK y FSK, en el capítulo 4 se muestran las pruebas y resultados obtenidos y en el capítulo 5 se presentan las conclusiones y recomendaciones.

Abstract

This work called "PSK and FSK DEMODULATION: IMPLEMENTATION IN FPGAs", was able to show that through programming gate arrays (Field Programmable Gate Array) systems can perform digital demodulation and modulation as a specific field of Telecommunications . This work aims to promote the use of robust electronic devices such as FPGAs that allow emulating applications in various fields of engineering programs at the Faculty of Technical Education for Development.

In Chapter 1, the introduction of the intervention work is done in Chapter 2, the state of the art FPGA platforms is described in chapter 3 the design of PSK and FSK demodulators is done in Chapter 4 show tests and results and in chapter 5 the conclusions and recommendations.

Capítulo 1: Introducción del Trabajo de Intervención.

1.1. Justificación.

Los moduladores de datos, sobre todo los destinados a producir señales de salida de envolventes constantes, hay componentes "de alta influencia", en las que incluso muy pequeñas desviaciones son ideales en su comportamientopueden conducir a grandes degradaciones en el rendimiento general del sistema. En el desarrollo "habitual" de técnicas de modulación de datos tal como se presenta en la mayoría de los textos de comunicaciones, las diversas técnicas se presentan en orden de complejidad, empezando con el más simple.

Por lo tanto BPSK se presentará primero, a continuación QPSK, seguido m-PSK, y así sucesivamente. Debido a su relación con las representaciones de señales complejas de envolvente, la modulación en cuadratura juega un papel central en la simulación de sistemas como por ejemplo las comunicaciones inalámbricas y los modelos para los moduladores en cuadratura, y demoduladores sirven como bloques de construcción para la mayoría de otros tipos de moduladores y demoduladores de datos.

Por lo tanto, este proyecto de grado realiza el análisis de modulación por desplazamiento de fase en cuadratura (QPSK) y se utiliza las FPGAs para el desarrollo de modelos genéricos de demodulación. La discusión se traslada a modulación por desplazamiento de fase binaria (BPSK) y muestra cómo se modela este formato más simple usando los modelos de modulación en cuadratura genéricos. Un enfoque similar se toma entonces para el desarrollo de modelos desplazamiento de frecuencia (FSK).

1.2. Antecedentes.

En la Maestría en Telecomunicaciones de la Universidad Católica de Santiago de Guayaquil no se han encontrado trabajos de grado en la que se desarrollen diseños sobre FPGAs de sistemas de demodulación para PSK y FSK.Se ha encontrado técnicas de procesamiento digital de señales pero a nivel de diseño de filtros digitales.

En la Facultad de Educación Técnica para el Desarrollo (FETD) específicamente en la Carrera de Ingeniería en Telecomunicaciones se encontró trabajos desarrollados sobre FPGAs, pero que a nivel de posgrado serían de muy poco valor científico, aunque su aporte en pregrado es muy significativo. La FETD dispone de sistemas de entrenamiento FPGA, tales como Altera y Xilinx para la enseñanza a nivel de sistemas digitales y microprocesadores.

A nivel de publicaciones en revistas internacional, se han encontrado trabajos en las que se desarrollan investigaciones utilizando los sistemas FPGAs, tales como Altera, Xilinx, entre otros. Los dos primeros son los más empleados en proyectos de investigación.

1.3. Definición del problema

Necesidad de desarrollar sistemas de demodulación digital tanto para PSK como FSK sobre la plataforma de entrenamiento FPGA, debido a que no hay disponibilidad de sistemas experimentales o equipos que permitan comprobar experimentalmente los procesos de demodulación.

1.4. Objetivos

1.4.1. Objetivo General:

Desarrollar sistemas de demodulación PSK y FSK para ser implementadas sobre FPGAs.

1.4.2. Objetivos específicos:

- ✓ Describir el estado del arte de los FPGA.
- ✓ Describir los fundamentos básicos de la modulación y demodulación digital PSK y FSK.
- ✓ Desarrollar los diseños de demodulación PSK y FSK sobre FPGA.
- ✓ Mostrar los resultados obtenidos de las demodulaciones PSK y FSK implementadas sobre FPGA.
- ✓ Fomentar el uso de dispositivos electrónicos para futuros trabajos investigativos de la Maestría en Telecomunicaciones.

1.5. Hipótesis

Mediante el desarrollo de demoduladores PSK y FSK se pretenderá demostrar mediante el diseño de diagramas de bloques sobre FPGAs, que el funcionamiento de estos son similares a los que se encuentran disponibles en sistemas de comunicaciones pero a costos muy altos.

1.6. Metodología de investigación.

Para el desarrollo del proyecto previo a la obtención del grado académico de Magister en Telecomunicaciones, se procede a describir la metodología empleada:

Alcance:

El alcance del proyecto de grado es del tipo **Exploratorio y Explicativo,** debido a que se explora las demodulaciones PSK y FSK y sus técnicas necesarias para ser tratadas como procesamiento de señales digitales que han originado tal fenómeno, y finalmente se explica cómo funciona el proceso de demodulación a través de la plataforma de entrenamiento FPGA.

Paradigma:

Empírico-Analítico

Enfoque:

Cuantitativo

Diseño de la Investigación:

El diseño de investigación es experimental, debido a que manipularán las variables del diseño de los demoduladores PSK y FSK, también se puede considerar correlacional-causal, debido a que se describirá que la causalidad implica correlación, pero no toda correlación implica causalidad.

Capítulo 2: Estado del Arte de Field ProgrammableGateway Array.

2.1. Breve historia de FPGA.

Los circuitos lógicos se utilizan para construir equipos informáticos, así como muchos otros tipos de productos. Todos estos productos son ampliamente clasificados como de hardware digital. La razón de que el nombre digital se utiliza se pondrá de manifiesto más adelante del presente documento. Se deriva de la forma en que la información se representa en los ordenadores, tales como señales electrónicas que corresponden a los dígitos de información.

La tecnología utilizada para construir el hardware digital ha evolucionado dramáticamente durante las últimas cuatro décadas. Hasta la década de los años 60’s, los circuitos lógicos fueron construidos mediante componentes voluminosos, tales como transistores y resistencias que venían como partes individuales. La aparición de la mayoría de chips (circuitos integrados) permitía plantar un sinnúmero de transistores, y en general se implementaban circuitos completos pero en único chip.

En el principio estos circuitos tenían sólo unos pocos transistores, pero a medida de que mejoraba la tecnología se hizo más grande. Los chips o circuitos integrados eran fabricados en una oblea de silicio (véase la figura 2.1). La oblea era cortada para producir los chips individuales, que luego eran colocadas dentro de un tipo especial de paquete de chips.

En 1970 se puso en práctica toda la circuitería necesaria para implementar un sistema microprocesador en un solo chip. A pesar de que los primeros microprocesadores tenían la capacidad de una computadora modesta comparada con los estándares actuales, que para ese entonces se abría la puerta a la revolución de procesamiento de la información,

proporcionando los medios para la aplicación de los ordenadores personales asequibles.

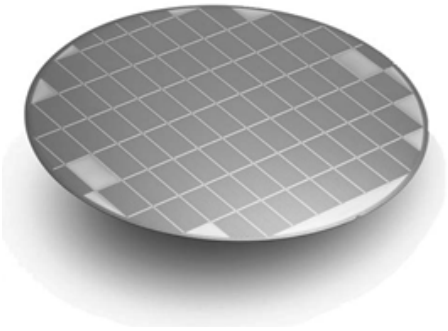


Figura 2. 1: Oblea de silicio
Fuente:

En 1980 el Presidente de Intel, Gordon Moore, señalaba que los avances tecnológicos de los circuitos integradostendría un acelerado crecimiento asombroso, que duplicarían los transistores de cada chip en promedio entre 1,5 y 2 años. Dicho crecimiento o fenómeno tecnológico se lo ha denominado “Ley de Moore” que continúa hasta la actualidad.

Para Karthik, S., Shreya, P., Srihari, P., &Viswanath, N. M. (2014), hoy en día, el uso de laboratorios remotos para la enseñanza a distancia en el campo de la electrónica está aumentando a un ritmo rápido. A diferencia de los laboratorios virtuales, que ofrecen simulaciones de software de procesos físicos, laboratorios remotos permiten a los usuarios trabajar de forma remota en experimentos reales, por lo que es más realista y poner a prueba su rendimiento. Pruebas experimentales es crucial en el campo de VLSI, donde los datos reales de validación son extremadamente importantes para la evaluar la efectividad del diseño.

2.2. Por qué FPGAs?

Las FPGAs llenan un vacío entre la lógica discreta y los PLDs más pequeños en el extremo inferior de la escala de complejidad y costosos

ASICs personalizados en el extremo superior. Consisten de una serie de bloques lógicos que se configuran mediante software. Los bloques de entrada y salida (E/S) rodean estos bloques lógicos. Ambos están conectados por interconexiones programables (véase la figura 2.2).

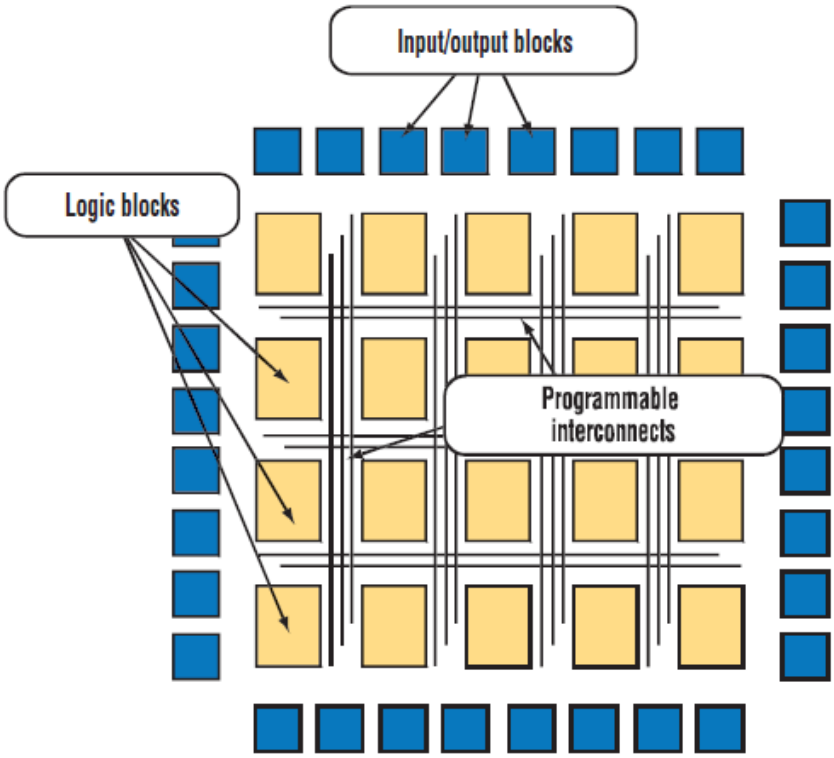


Figura 2. 2: Arquitectura flexible de bloques lógicos.
Fuente: Maliniak, D. (2009)

La tecnología de programación en una la FPGA determina el tipo de célula lógica básica y el esquema de interconexión. A su vez, las células lógicas y los esquemas de interconexión determinan el diseño de los circuitos de E/S, así como el esquema de programación. Hace apenas unos años, la mayoría de FPGAseran medidas en decenas de miles de puertas del sistema y funcionaban a una frecuencia de 40 MHz.

La mayoría FPGAs a menudo cuestan más de $ 150 para las partes más avanzadas de ese entonces. Hoy, sin embargo, las FPGAs ofrecen millones de puertas de capacidad lógica, funcionan a 300 MHz, pueden

costar menos de $ 10, y hasta ofrecen funciones integradas, como los procesadores y la memoria (véase la Tabla 2.1).

Tabla 2. 1:Recursos principales disponibles en dispositivos FPGA.

Features	Xilinx Virtex II Pro	Altera Stratix	Actel Axcelerator	Lattice ispXPGA
Clock management	DCM Up to 12	PLL Up to 12	PLL Up to 8	SysCLOCK PLL Up to 8
Embedded memory blocks	BlockRAM Up to 10 Mbits	TriMatrix memory Up to 10 Mbits	Embedded RAM Up to 338 kbits	SysMEM blocks Up to 414 kbits
Data processing	Configurable logic blocks and 18-bit by 18-bit multipliers Up to 125,000 logic cells and 556 multiplier blocks	Logic elements and embedded multipliers Up to 79,000 LEs and 176 embedded multipliers	Logic modules (C-Cell and R-Cell) Up to 10,000 R-Cells and 21,000 C-Cells	Based on programmable functional unit Up to 3844 PFUs
Programmable I/Os	SelectI/O	Advanced I/O support	Advanced I/O support	SysI/O
Special features	Embedded PowerPC 405 cores RocketI/O multi-gigabit transceiver	DSP blocks High-speed differential I/O and interface standards support	PerPin FIFOs for bus applications	SysHSI for high-speed serial interface

Fuente: Maliniak, D. (2009)

Las FPGAs ofrecen todas las funciones necesarias para implementar diseños más complejos. La gestión del reloj se ve facilitada por el chip PLL (bucle de enganche de fase) o circuitos DLL (bucle de enganche de retardo). Los bloques de memoria dedicados pueden ser configurados como RAMs básicos de un solo puerto, ROM, FIFO, o CAM.La capacidad de vincular lasFPGAscon placas madre, buses de alta velocidad, y memorias proporcionan soporte de varios estándares de terminación única y diferencial de bloques de E/S.

Asimismo se ha encontrado FPGAscuyos recursos de construcción del sistema son: alta velocidad de serie de los bloques de E/S, módulos de cálculo, procesadores embebidos, y grandes cantidades de memoria. Ahora,

con la llegada de dispositivos mucho más grandes y la disminución de los costos por parte, las FPGAshan encontrado su camino desde banco de prototipos y en la producción (véase la Tabla 2.2).

Tabla 2. 2:Usos de FPGA.

	Emulation: 3%	Prototyping: 30%	Preproduction: 30%	Production: 37%
Time-to-market	Fairly high; fast compile times	Fairly high; fast compile times	Fairly high; fast compile times	Fairly high; fast compile times
Performance	Not stringent	Not stringent	Very critical	Very critical
Volume	Very low per application	Low per application	Moderately high per application	High per application

Fuente: Maliniak, D. (2009)

2.3. Visión general de las FPGAs.

Una matriz de puertas programables de campo (FPGA) es un circuito integrado que está diseñado para ser configurado después de la fabricación. Las FPGAs se pueden utilizar para implementar cualquier función lógica que un circuito integrado de aplicación específica (ASIC). Para los requisitos variables, una porción de FPGA puede también ser parcialmente reconfigurado mientras el resto de un FPGA todavía se está ejecutando.

A diferencia de otras tecnologías, que implementan hardware directamente en el silicio, cualquier error en el producto basado en FPGA finalmente se puede corregir fácilmente con sólo reprogramar el FPGA. Todas las actualizaciones futuras en el producto final también pueden actualizarse fácilmente con sólo descargar una nueva aplicación de flujo de bits. La facilidad de programación y depuración con FPGAs disminuye los costes no-recurrentes de ingeniería (NRE) y el tiempo de salida al mercado de los productos basados en FPGA generalmente.

Según Betz, V., Marquardt, A., & Rose, J. (1999) la reconfiguración de FPGAs se debe a sus componentes reconfigurables llamados bloques lógicos, los cuales están interconectados por una red de enrutamiento

reconfigurable. Mientras que para Marrakchi, Z., Mrabet, H., Farooq, U., &Mehrez, H. (2009) existen dos principales topologías de interconexión de enrutamiento:

a. Redde encaminamiento basado en árbol y
b. Redde enrutamiento basado en malla.

Una arquitectura FPGA basado en árbol, se crea mediante la conexión de bloques lógicos en grupos. Estos grupos están conectados de forma recursiva para formar una estructura jerárquica. En contrario, que la basada en malla de la arquitectura FPGA en bloques realiza interconexiones lógicas a través de una malla de 2-D de la red de enrutamiento.

Según Marrakchi, Z. (2008) la topología de interconexión basados en árbol ocupa menos área que la interconexión de topología basada en la malla, sin embargo, un FPGA basado en árboles sufre de problemas de escalabilidad de diseño. El diseño de un FPGA basado en malla es escalable y por lo tanto es comúnmente utilizado por los vendedores de FPGA comerciales como Xilinx (2014) y Altera (2014).

En la figura 2.3se muestra la arquitectura tradicional FPGA basada en malla. Donde los bloques lógicos configurables (CLBs) están dispuestos en una rejilla 2D y están interconectados por una red de encaminamiento programable. La entrada/salida (I/O) de bloques en la periferia del chip FPGA también están conectados a la red de encaminamiento programable.

Para Marrakchi, Z. (2008), la red de encaminamiento comprende de pistas de canal de enrutamiento horizontales y verticales. Las cajas de interruptores se conectan a las pistas de enrutamiento horizontales y verticales de la red de encaminamiento. Mientras que las cajas de conexión conectan la lógica y los pines del bloque de E/S con pistas de enrutamiento adyacentes.

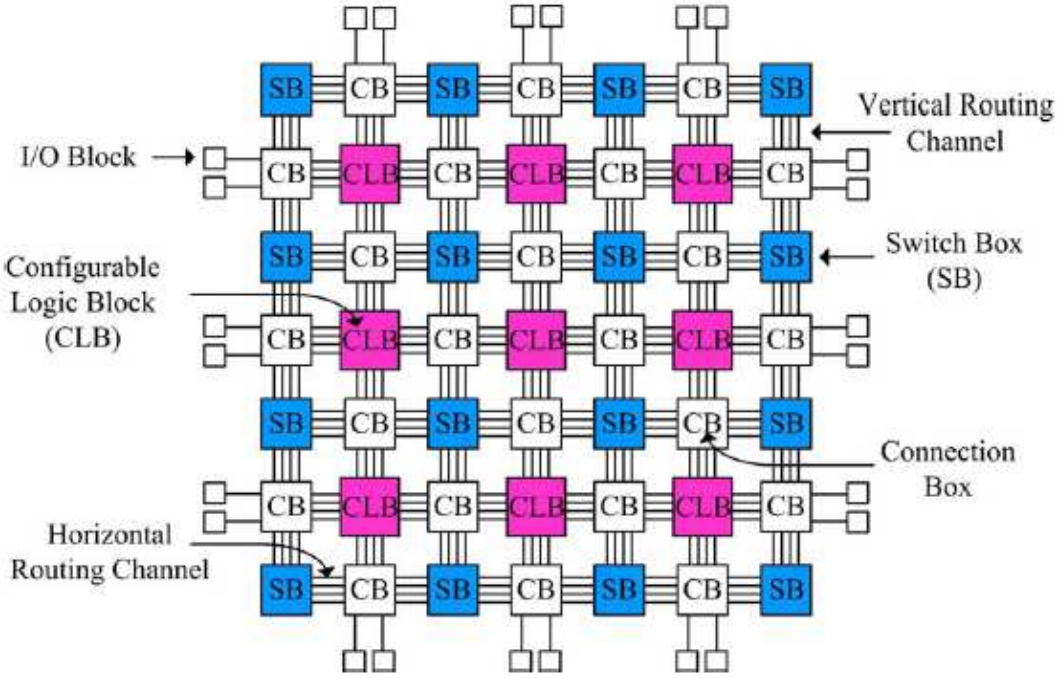


Figura 2. 3: Arquitectura FPGA basada en malla.
Fuente: Betz, V., et al (1999)

El flujo de software convierte un circuito de hardware de destino en CLB interconectados con las E/S según sea el caso, y luego se observa los mapas en la FPGA. El flujo de software también genera un flujo de bits, que se programa en el FPGA para ejecutar el circuito de hardware de destino.

2.3.1. Lógica configurable de bloque

Un bloque lógico configurable (CLB) es un componente básico de una FPGA que implementa la funcionalidad de la lógica de un diseño de aplicación de destino. Una CLB puede comprender de un único elemento de lógica básica (BLE), o un grupo de BLES interconectadas localmente. Una simple BLE consiste en una tabla de consulta (LUT), y de un Flip-Flop. Una LUT con k entradas (LUT-k) contiene los bits de configuración 2k; que se puede implementar cualquier función booleana de k-entrada.

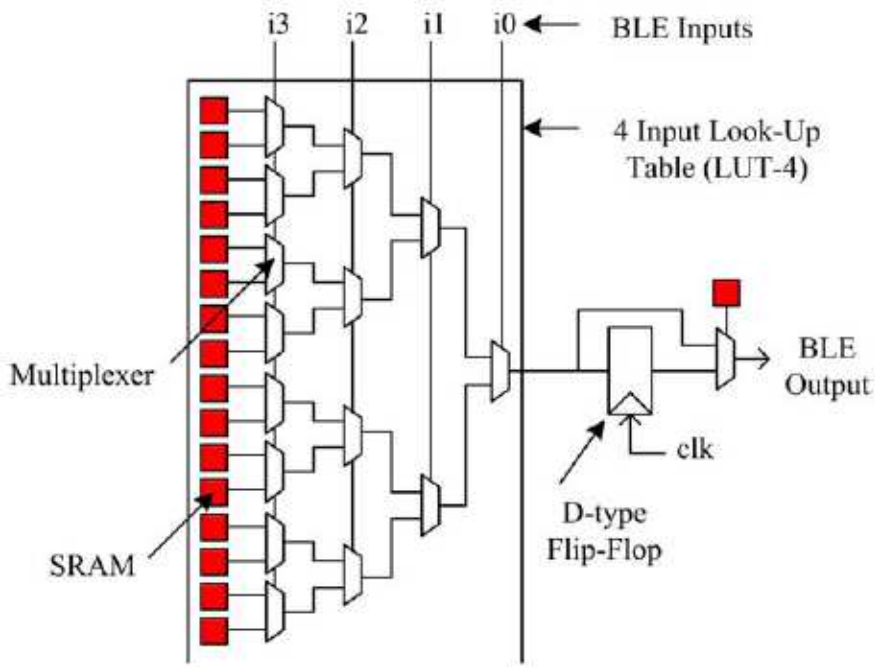


Figura 2. 4: Elementos de Lógica Básica (BLE).
Fuente: Betz, V., et al (1999)

En la figura 2.4se muestra una BLE sencilla que consta de una entrada de 4 Look-Up Table (LUT-4) y un Flip-Floptipo D. Donde el LUT-4 utiliza una SRAM (memoria estática de acceso aleatorio) de 16 bits para implementar cualquier función booleana 4 entradas. La salida de la LUT-4 está conectada a un Flip-Flop opcional. Un multiplexor selecciona la salida BLE para ser la salida de un Flip-Flop o la LUT-4.

Una tabla de búsqueda con mayor número de entradas, reduce el número total de LUTs necesarias para asignar un circuito de hardware. Donde la lógica funcional permite el mapeado de un solo LUT. Esto reduciría el tiempo la intercomunicación entre las LUTs, y en consecuencia la velocidad del circuito de hardware mejora.

Sin embargo, una LUT con mayor número de entradas aumenta su área de forma exponencial. Mientras que Ahmed, E. & Rose, J. (2000) manifiesta que han medido el efecto de la cantidad de entradas LUT en área,

velocidad y enrutabilidad de FPGAs. También, han concluido que las LUT de 4 entradas proporcionan un buen compromiso entre la velocidad y la densidad de FPGAs.

Un CLB puede contener un grupo de BLEs conectados a través de una red de enrutamiento local. En la figura 2.5se muestra un racimo de 4 BLEs; cada BLE contiene una LUT-4 y un Flip-Flop. La salida BLE es accesible a otros BLEs del mismo grupo a través de una red de enrutamiento local. El número de pines de salida de un grupo es igual al número total de BLEs en un clúster (con cada BLE que tiene una sola salida).

Considerando que el número de pines de entrada de un clúster puede ser menor o igual que a la suma de los pines de entrada requeridos por todos los BLEs del clúster. Las FPGAs modernas contienen típicamente de 4 a 10 BLEs en un solo clúster.

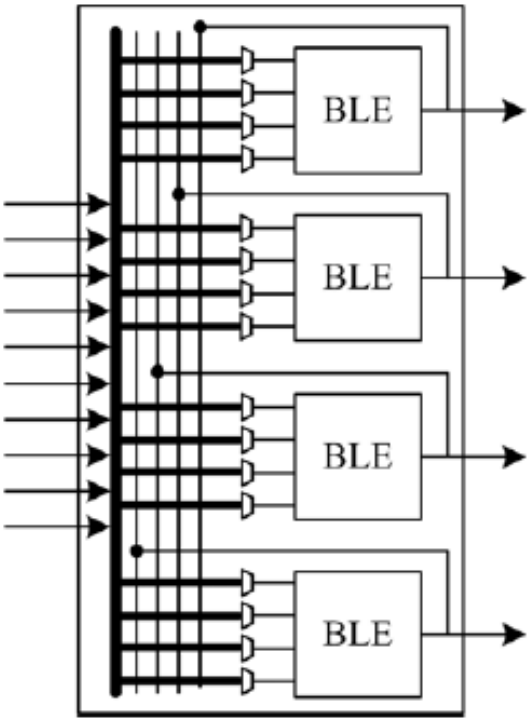


Figura 2. 5: Bloque lógico configurable (CLB) que tiene cuatro BLEs.
Fuente: Betz, V., et al (1999)

2.3.2. Enrutamiento de red

Betz, V., et al (1999) manifiestan que la red de encaminamiento de una FPGA ocupa entre un 80% y 90% del área del chip FPGA, mientras que el área lógica ocupa sólo entre el 10% y 20% de área. La flexibilidad de una FPGA depende principalmente de su red de encaminamiento programable. Una red de enrutamiento basado en FPGA de malla consta de pistas de enrutamiento horizontales y verticales que se interconectan a través de las cajas de interruptores (SB).

Los bloques lógicos están conectados a la red de enrutamiento a través de cajas de conexión (CB). La flexibilidad de un bloque de conexión (Fc), es el número de pistas de enrutamiento de canal adyacente que están conectados a la clavija de un bloque. La conectividad de los pines de entrada de bloques lógicos con el canal de enrutamiento adyacente se denomina como$F_c(in)$; la conectividad de los pines de salida de los bloques lógicos con el canal de enrutamiento adyacente se llama como$F_c(out)$.

Un$F_c(in)$ igual a 1.0, significa que todas las pistas de canal adyacente de enrutamiento están conectadas a la clavija de entrada del bloque lógico. Un $F_c(in)$igual a 0.5, significa que sólo el 50% de pistas del canal adyacente de enrutamiento están conectados a la clavija de entrada.La flexibilidad de la caja del interruptor (Fs), es el número total de pistas con el que cada pista que entra se conecta a la caja del interruptor.

Las pistas de enrutamiento conectados a través de una caja de interruptores puedeser bidireccional o unidireccional (también conocido como direccionales). En la figura 2.6 se muestran las cajas de conmutación bidireccional y unidireccional, con Fs igual a 3. Las pistas de entrada (o cables) en tanto éstos cambian, las cajas se conectan a otras 3 pistas de la misma caja de interruptores. La única limitación de la caja del interruptor

unidireccional es que su ancho de canal de enrutamiento debe ser en múltiplos de 2.

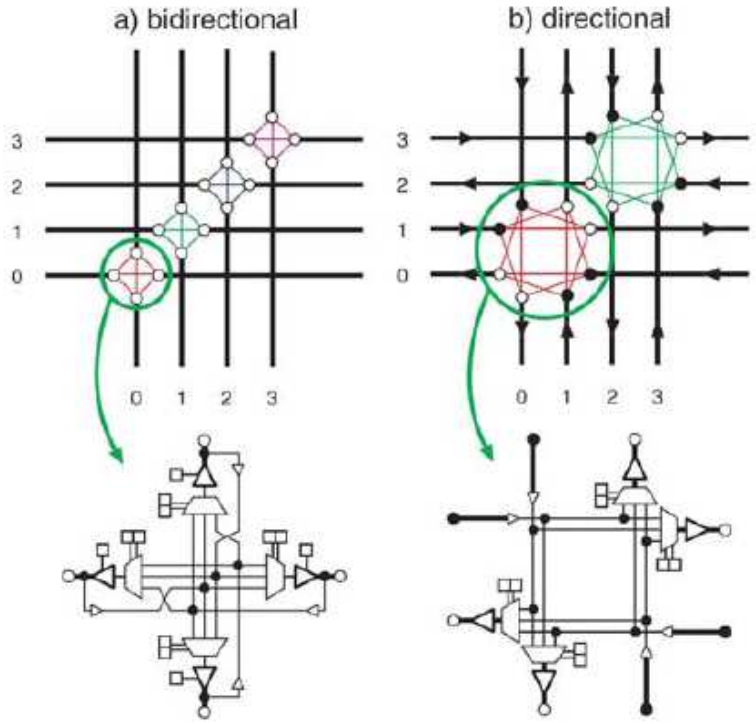


Figura 2. 6: Interruptor de bloque, con longitud de 1 alambre.
Fuente: Lemieux, G., Lee, E., Tom, M., &Yu, A. (2004)

Los cables Multi-longitud se crean para reducir el área y el retraso. En la figura 2.7 se muestra un ejemplo de los diferentes cables de longitud. Los segmentos de cable más largos abarcan múltiples bloques y requieren un menor número de interruptores, lo que reduce el área de enrutamiento y de retrasos. Sin embargo, también disminuyen la flexibilidad de enrutamiento, lo que reduce la probabilidad de encaminar un circuito de hardware con éxito.

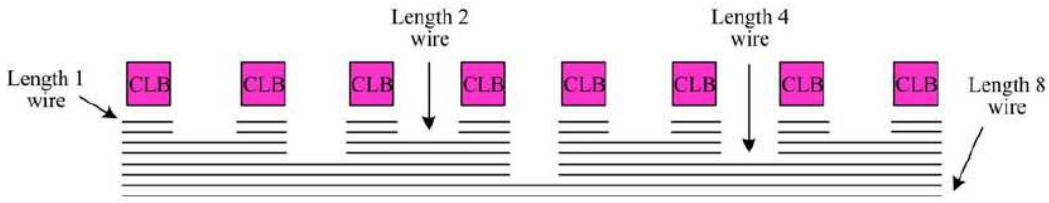


Figura 2. 7: Distribución del segmento de canal.
Fuente: Lemieux, G., et al (2004)

Las FPGAs comerciales modernas suelen utilizar una combinación de cables largos y cortos para equilibrar la flexibilidad, la zona y el retardo de la red de enrutamiento.En general, los pines de salida de un bloque se pueden conectar a cualquier pista de enrutamiento a través de transistores de paso. Cada transistor de paso forma una salida triestado que se puede activar o desactivar de forma independiente. Sin embargo, la técnica de cableado conductor solo se puede utilizar también para conectar los pines de salida de un bloque a las pistas de enrutamiento adyacentes.

Para el cableado del único controlador, los elementos de tres estados no pueden utilizarse; la salida del bloque necesita ser conectado a la red vecina de encaminamiento a través de multiplexores en la caja del interruptor. Las modernas arquitecturas de FPGA comerciales se han movido hacia el uso de conductores únicos, pistas de enrutamiento de dirección.Lemieux, G., et al (2004) manifiesta que si el cableado unidireccional del único controlador es utilizado en lugar del cableado bidireccional, mejoraría en un 25% del área y un 9% en el retardo. Con esto se podrá lograr un 32% en el área de retardo. Todas estas ventajas se consiguen sin hacer ningún cambio importante en el flujo del FPGA CAD.

2.3.3. Flujograma para desarrollo de software.

Uno de los principales aspectos de la investigación de FPGAs es el desarrollo de flujogramas para desarrollar software, necesarios para mapear las aplicaciones de hardware en una FPGA. La eficacia y la calidad de una FPGA dependen en gran medida del flujogramasuministrado en una FPGA. El flujogramarealiza una descripción del diseño de ciertas aplicaciones en un lenguaje de descripción de hardware (HDL) y la convierte en una corriente de bits que es eventualmente programado en la FPGA.

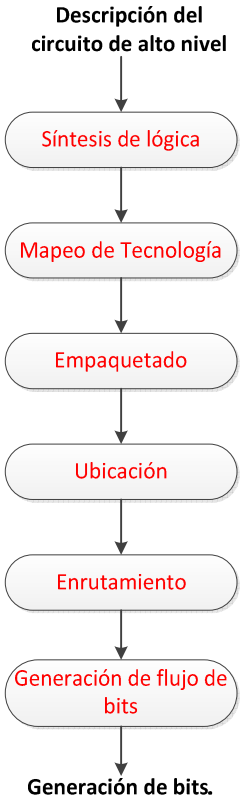


Figura 2. 8: Configuración del flujograma para FPGA.
Fuente: Lemieux, G., et al (2004)

En la figura 2.8se muestra un flujogramacompleto para la programación de un circuito cuya aplicación será en una FPGA basada en malla. Una breve descripción de varios módulos de software de flujo se describe a continuación:

a. Síntesis de lógica:

La síntesis lógica transforma una descripción HDL (VHDL o Verilog) en un conjunto de puertas booleanas y flip-flops. Las herramientas de síntesis transforman la descripción de trasferencia entre registros de lenguaje RTL de un diseño en una red jerárquica booleana.

Varias técnicas independientes de la tecnología se aplican para optimizar la red booleana. La función de costo típico de optimizaciones independientes de la tecnología, es el recuento total literal de la representación factorizada de la función lógica. El recuento literal se

correlaciona muy bien con el área de circuito. Más detalles de la síntesis de la lógica están fuera del alcance del presente trabajo investigativo.

b. Mapeo de Tecnología:

Después de la síntesis lógica, se realizan optimizaciones dependientes de la tecnología. Estas optimizaciones transforman la red booleana independiente de la tecnología, en una red de compuertasen la biblioteca de la tecnología dada. La descripción de las tecnologías para FPGAs transforman las redes booleanasal conjunto disponible de bloques en una FPGA.

Para una arquitectura FPGA tradicional, la red booleana se transforma en tablas de consulta y flip-flops. Los algoritmos del mapeo de tecnología optimizan una red booleana dada para un conjunto de diferentes funciones objetivo, incluyendo profundidad, el área y potencia. Para Cong, J., &Ding, Y. (2000) el algoritmo de flujogramaFlowMap, es una herramienta académica ampliamente utilizada para el mapeo de la tecnología FPGA.

También, es capaz de encontrar una solución de profundidad óptima en tiempo polinómico. Las versiones posteriores de FlowMaphan mejorado aún más para optimizar el área y tiempo de ejecución de una red booleana mientras se mantiene la misma profundidad. El resultado final del mapeo de tecnología FPGA es una red de dispositivos de E/S, LUTs y flip-flops.

c. Empaquetado (Packing).

Una FPGA basada en malla, consiste de una serie de bloques lógicos configurables (CLBs). Cada CLB se compone de un conjunto de elementos de lógica básica (BLES). Una BLE consiste en una tabla de

búsqueda y un Flip-Flop. La fase de empaquetado también llamado grupos de fase del cluster, consiste en una tabla de búsqueda y un Flip-Flop en una BLE, y de grupos diferentes de BLEs en unaagrupación de BLEs.

Estos BLEso grupos de BLEs se pueden asignar directamente sobre los CLB de una FPGA. El objetivo principal es la optimización para agrupar las tablas de consulta, los flip-flops y los BLEs, de tal manera que se reduce al mínimo la comunicación inter-cluster. El resultado final de embalaje es una red de E/S y CLB.

d. Ubicación (Placement).

El algoritmo de colocación determina la posición de los CLBs y las instancias de E/S en una lista llena de conexiones en la respectiva CLB y los bloques de E/S en la arquitectura FPGA. El objetivo principal del algoritmo de ubicación es colocar bloques conectados cerca unos de otros de manera que se requiere un mínimo de recursos de enrutamiento para enrutar sus conexiones.

El algoritmo de ubicación también puede servir para cumplir con otros requisitos arquitectónicos o de optimización, tales como el equilibrio de la densidad de alambre a través de FPGA. Existen tres tipos de algoritmos de ubicación comúnmente utilizados que son:

(i) Algoritmo de particionamientobasado en la ubicación.

El enfoque basada en la partición de la ubicación es generalmente adecuado para arquitecturas FPGA jerárquicas. El particionado se aplica de forma recursiva para distribuir instancias de listas de conexiones entre grupos. El objetivo es reducir la comunicación externa y fusionar las instancias altamente conectadas en el mismo grupo.

(ii) Algoritmo analítico de ubicación.

Los algoritmos analíticos de ubicación comúnmente utilizan una función objetivo cuadráticode longitud del cable. Aunque, este objetivo es sólo una medida indirecta de la longitud del hilo; su principal ventaja es que se puede minimizar de manera muy eficiente y por tanto es adecuado para el manejo de grandes problemas.

(iii) Simulación del algoritmo basada en la ubicación de hibridación.

El algoritmo de hibridación simulado utiliza el concepto de hibridación (recocido) para metales fundidos que son enfriados gradualmente para producir objetos de metal de alta calidad. El algoritmo de ubicación de hibridación es muy bueno en la aproximación de una solución aceptable para ubicar una lista de conexiones y ser colocado en una FPGA.

En la figura 2.9 se muestra un cuadro de límite de una red de 6terminales hipotéticos.

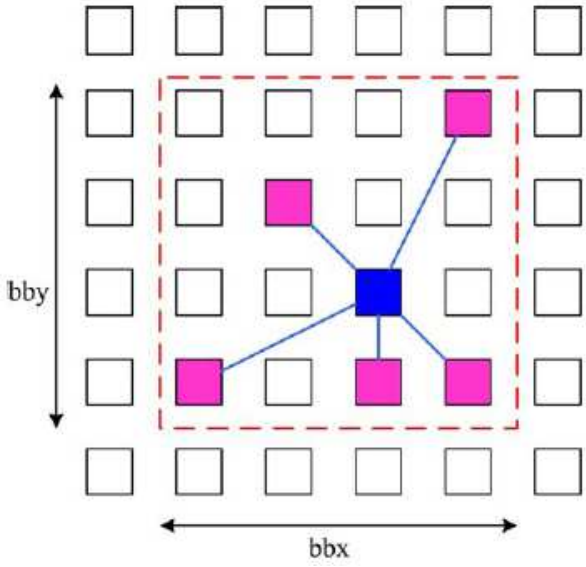


Figura 2. 9: Cuadro delimitador de una red de 6 terminales hipotética.
Fuente: Lemieux, G., et al (2004)

e. Enrutamiento (Routing).

Una vez que las instancias de la lista de conexiones se colocan en la FPGA, las conexiones entre las diferentes instancias se enrutan utilizando los recursos de enrutamiento disponibles. El problema de enrutamiento de FPGAs consta del enrutamiento de las señales (o redes) de una manera tal que no más de una señal utilizada como recurso de enrutamiento.

El algoritmo de enrutamiento se utiliza comúnmente para FPGAs. Con el fin de realizar el enrutamiento en una arquitectura FPGA, la arquitectura de enrutamiento se modela inicialmente como un grafo dirigido a los diferentes nodos que están conectados a través de los bordes.Cada alambre de enrutamiento de la arquitectura está representado por un nodo, y la conexión entre dos cables está representada por un borde.

En la figura 2.10 se muestra una pequeña parte de la arquitectura de enrutamiento en forma de un grafo dirigido. Cuando una lista de conexiones se enruta en el gráfico de encaminamiento FPGA, cada red (es decir, la conexión de una instancia de conductor con sus instancias receptor) se enruta utilizando un algoritmo de congestión impulsada.

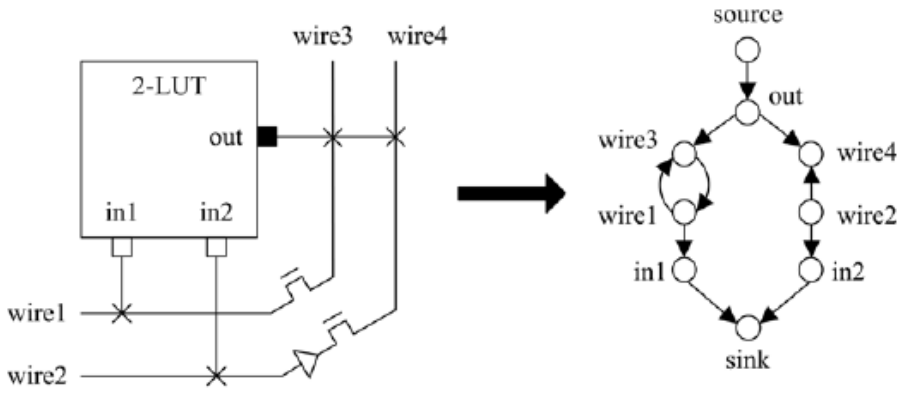


Figura 2. 10: Modelado de la arquitectura FPGA como grafo dirigido.
Fuente: Lemieux, G., et al (2004)

Una vez que todas las redes de una lista de conexiones son dirigidas, se dice entonces que una iteración de enrutamiento ha de ser completada. Al final de una iteración, puede haber conflictos entre diferentes redes que comparten los mismos nodos.

Los parámetros de congestión de los nodos se actualizan, y las iteraciones de enrutamiento se repiten hasta que el enrutamiento converge a una solución factible (es decir, todos los conflictos se resuelven) o el encaminamiento falla (es decir, número máximo de iteración se ha alcanzado, y algunos conflictos de enrutamiento siguen sin resolverse).

f. Generación de flujo de bits.

Cong, J., &Ding, Y. (2000) sostienen que una vez que la lista de conexiones se coloca y se encamina en una FPGA, la información de los flujos de bits son generados para la lista de conexiones. Este flujo de bits se programa en la FPGA utilizando un cargador de corriente de bits. El flujo de bits de una lista de conexiones contiene información sobre qué bits de la SRAM de un FPGA son programados a 0 o a 1.

El generador de flujo de bits, lee la descripción de las tecnologías, el empaquetado y la información de ubicación para poder programar los bits de las tablas de consulta.La información de enrutamiento de una lista de conexiones, se utiliza para programar correctamente los bits SRAM de cajas de conexión y de cajas de interruptores.

2.4. Tendencias de investigación en las FPGAs.

Los productos basados en FPGA son muy eficaces para la producción de volumen baja a mediana, siendo fáciles de programar y de depurar, y tienen menos costos NRE y menos tiempo de comercialización.Todas estas

grandes ventajas de una FPGA vienen a través de su capacidad de reconfiguración. Sin embargo, la mismareconfigurabilidad es la principal causa de sus desventajas.

La flexibilidad de FPGAs, se debe principalmente a su red de encaminamiento reprogramable que tiene entre 80% y 90% de toda el área de FPGA. El área de la lógica esta entre 10% y 20% de la FPGA. Debido a esta razón las FPGAs son mucho más grandes, más lentas y con mayor consumo de energía que los ASIC; por lo tanto no son adecuados para la producción de grandes volúmenes y de alto rendimiento o de bajo consumo de energía.

Tanto el hardware reconfigurable y la arquitectura FPGA tienen muchos campos de investigación activos. Un aspecto importante de la investigación en hardware reconfigurable, gira en torno a la disminución de los inconvenientes de FPGAs, con o sin poner en peligro a sus principales beneficios. A continuación se presentan algunas de las principales soluciones de las relaciones de intercambio que se han propuesto en la resolución de la zona, la velocidad, la potencia y/o problemas de producción de volumen de FPGAs.

a. Hard-Blocks

La densidad lógica de una FPGA se mejora mediante la incorporación Hard-blocks dedicados en una FPGA. Los Hard-Blockso también conocidas como pequeñas ASICs, en las FPGAs aumentan su velocidad y reducen su superficie total y el consumo de energía. Los Hard-blocks, pueden incluir multiplicadores, sumadores, memorias, unidades de punto flotante, etc.

En la figura 2.11 se muestra una arquitectura FPGA comercial que utiliza incrustado duros bloques.

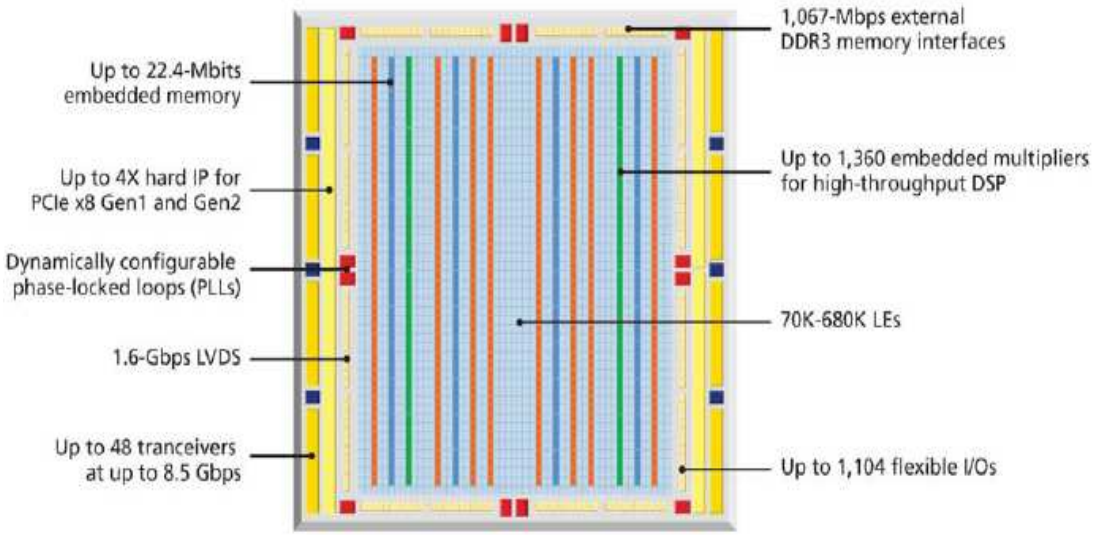


Figura 2. 11: Elementos de la arquitectura de la FPGA Stratix IV.
Fuente: Beauchamp, M., et al. (2006)

En este sentido Beauchamp, M., Hauck, S., Underwood, K., &Hemmert, K. (2006) sostienen que se han introducido unidades de punto flotante embebidos en FPGAs, mientras que Ho, C. H., Leong, W., Luk, W., Wilton, S., & López-Buedo, S. (2006) desarrollaron la metodología de bloque virtual incrustado para modelar bloques incrustados arbitrarios en las FPGAs comerciales existentes.

b. Aplicaciones específicas de FPGAs.

El tipo de bloques lógicos y la red de encaminamiento en una FPGA puedenoptimizarse para obtener ventajas de la zona de rendimiento y para un mejor dominio de aplicaciones dadas (aplicaciones de controlorientado al trayecto, aplicaciones orientadas al camino de datos, etc.).

W., Luk, et al. (2006) sostienen que estostipos de FPGAs pueden incluir diferentes variedades de hard-blocks (bloques duros) deseados, y la cantidad apropiada de la flexibilidad necesaria para las interconexiones de aplicaciones o de buses basados en datos que se

interconecta a base de bits. En la figura 2.12se muestra una matriz aritmética reconfigurable para aplicaciones multimedia.

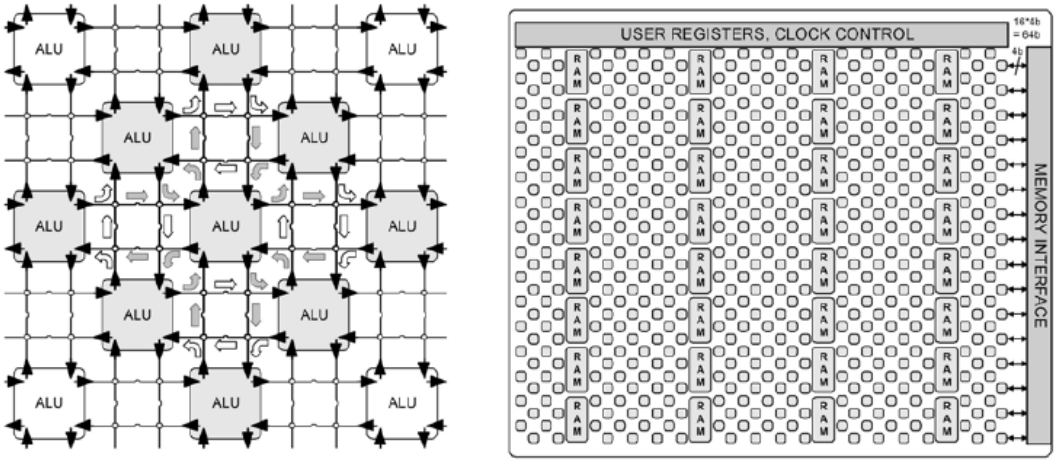


Figura 2. 12:Matriz aritmética reconfigurable para aplicaciones multimedia.
Fuente: Beauchamp, M., et al. (2006)

c. ASIC estructuradas para FPGAs.

La facilidad de diseño y creación de prototipos con FPGAs puede explotarse para diseñar rápidamente una aplicación de hardware en una FPGA. Más tarde, las mejorías en cuanto al área (zona), la velocidad, la potencia y la producción en volumen se pueden alcanzar mediante la migración de la aplicación de diseño de FPGA a otras tecnologías comoASICs estructurados.

En este sentido, Altera proporciona la facilidad para migrar diseños de aplicaciones basadas en Stratix IV a HardCopy IV (2012). El eASICNextreme(eASIC, 2014) utiliza una FPGA como caudal de diseño para asignar un diseño de aplicación enLUTs programables de SRAM, que están interconectadas a través de la programación de máscara de pocas capas de enrutamiento superiores.

Para Compton, K. &Hauck, S. (2007) el cASIC explora el espacio de diseño entre ASIC y FPGA; los núcleos ASIC configurables están diseñados para ejecutar un conjunto dado de diseños de aplicaciones

que a veces son exclusivos. El nivel lógico (Tierlogic, 2012) es un proveedor de FPGA recientemente lanzado y que ofrece dispositivos FPGA basadas en SRAM 3D para la creación de prototipos y producción temprana.

d. Procesadores mediante FPGAs.

Para Jones, A. K., Hoare, R., Kusic, D., Fazekas, J., & Foster, J. (2005) una cantidad considerable de área de una FPGA puede ser salvada, mediante la aplicación de una parte de la ruta de control de circuitos en un microprocesador, y solamente el cálculo del camino de datos intensivo de un circuito se implementa en las FPGAs. Una FPGA está conectada a un microprocesador de diferentes maneras:

(i) Un procesador sencillo (blando) se implementa con recursos configurables de una FPGA.

(ii) Unprocesador se incorpora en una FPGA como un bloque duro (hard-blocks) dedicado o como procesadores PowerPC embebido en Xilinx Virtex-4,

(iii) UnaFPGA es unida con la tubería de un procesador para ejecutar instrucciones de hardware personalizados.

En la figura 2.13se ilustra un procesador VLIW que soporta instrucciones de hardware específicos.

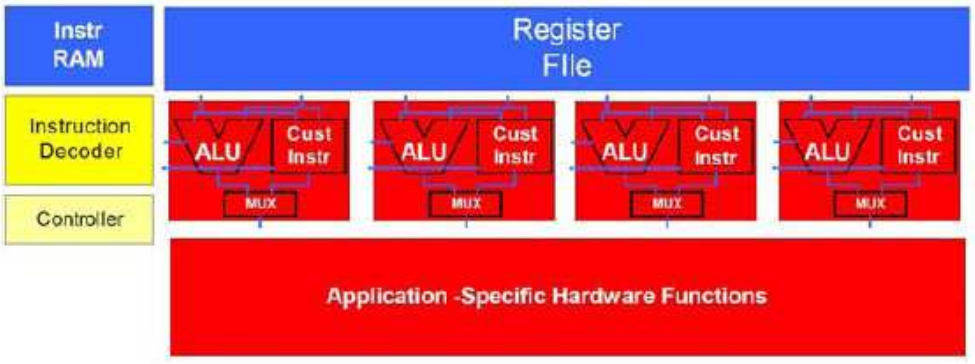


Figura 2. 13: Matriz aritmética reconfigurable para aplicaciones multimedia.
Fuente: Jones, A. K., et al. (2005)

e. Señales de tiempo multiplexados.

Essen, B. V., Wood, A., Carroll, A., Friedman, S., Panda, R., Ylvisaker, B., Ebeling, C., &Hauck, S. (2009) manifiestan que en lugar de utilizar una pista de enrutamiento dedicado para el encaminamiento de una sola señal, un alambre de encaminamiento es multiplexada en el tiempo y utilizadas por diferentes señales en diferentes momentos.

De esta manera, una considerable cantidad de recursos de enrutamiento puede reducirse logrando así ganancias en área. El tiempo de multiplexación se maneja mediante la adición de circuitos de hardware especial. Estos recursos adicionales hacen que el tiempo de multiplexaciónsea menos atractivo para las arquitecturas FPGA comerciales, donde se utilizan generalmente solo bits de enrutamiento.

f. FPGAs multiplexados en el tiempo

Para Miyamoto, N., &Ohmi, T. (2008) la densidad de la capacidad o de la lógica de FPGAs se incrementa mediante la ejecución de diferentes partes de un circuito en una FPGA en un modo de multiplexación de tiempo. Un diseño de la aplicación se divide en distintos subcircuitos, y cada subcircuito se ejecuta como un contexto individual de la FPGA.

La información de estado de cada subcircuito se guarda en un registro ante un nuevo contexto que se ejecuta en una FPGA. Tabula (Tabula, 2011) es un proveedor de FPGAsque tiene poco tiempo en el mercado que ofrece FPGAs multiplexados en el tiempo.

2.5. Empaquetado, colocación y enrutamiento versátil.

Para el empaquetado, colocación y enrutamiento versátil para FPGAs (comúnmente conocido como VPR), donde el entorno de exploración basado en FPGA de malla es muy utilizado para fines académicos. También permite

explorar arquitecturas FPGA basadas en malla mediante el empleo de un enfoque empírico.

Los circuitos de referencia son tecnológicamente mapeados, colocados y enrutadas en arquitecturasFPGAs deseadas. Más tarde, el área y los retardos de las FPGAs son medidos para determinar los mejores parámetros de arquitecturas. Las diferentes herramientas CAD en VPR están altamente optimizadas para garantizar resultados de alta calidad; como herramientas CAD sencillas, que pueden conducir a conclusiones inexactas de arquitecturas de FPGAs. Los modelos de área y de retardo son suficientemente precisos para comparar el efecto de diferentes cambios en la arquitectura.

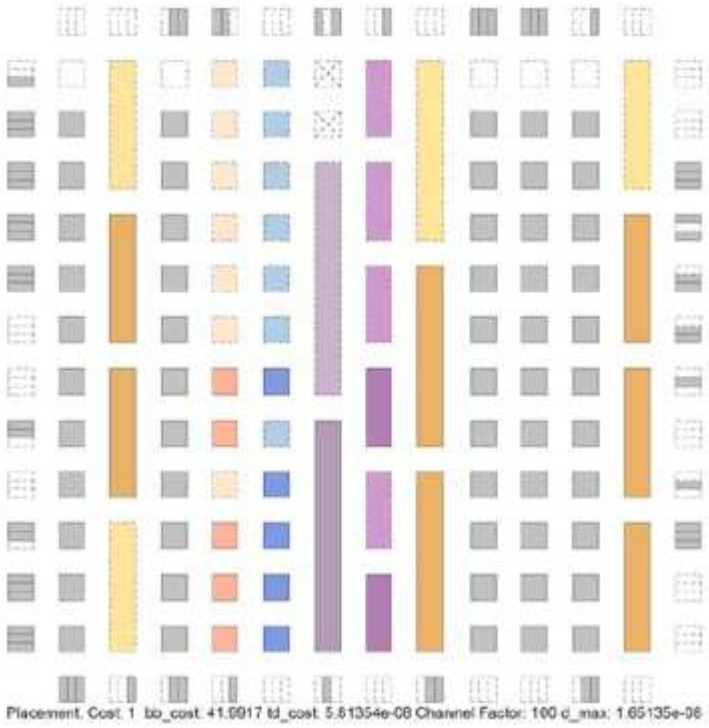


Figura 2. 14: FPGA heterogénea en VPR.
Fuente: Jones, A. K., et al. (2005)

La última versión de VPR conocido como VPR 5.0 es desarrollado mediante hard-blocks (bloques duros, tales como bloques multiplicadores y de memoria) y de cables de enrutamiento de único controlador. El hard-

blocks estárestringido para una columna de ancho de rejilla, y que dicha columna puedeestar compuesta de un mismo tipo de bloque.En la figura 2.14se ilustra una FPGA heterogénea con 8 tipos diferentes de bloques, conocida como VPR 5.0. También produce una optimización de los modelos eléctricos para una amplia gama de arquitecturas de distintas tecnologías de proceso.

2.6. Marco para la exploración de arquitecturas reconfigurables.

También conocido como Madeo, es una suite de diseño para la exploración de arquitecturas reconfigurables. Incluye un entorno de modelado que soporta multi-granularidad de arquitecturas heterogéneas con topologías irregulares. El marco madeo inicialmente permite modelar una arquitectura FPGA. Las características de la arquitectura se representan como un modelo abstracto común. Una vez definida la arquitectura, las herramientas de diseño CAD de Madeo se pueden utilizar para asignar una lista de red de destino en la arquitectura.

Madeo incorpora algoritmos de ubicación y enrutamiento (el mismo que el utilizado por VPR), un generador de flujo de bits, un simulador de lista de conexiones, y un generador de distribución física. También apoya la prospección de arquitecturas y prototipadode FPGAs. Varias FPGAs, incluyendo algunas arquitecturas comerciales (tales como la familia Xilinx Virtex) han sido modelados usando Madeo. La disposición física se produce como descripción VHDL.

2.7. Arquitectura de Altera.

Altera Stratix IV, es una familia de arquitecturas FPGAs basadas en malla. En la figura 2.11 se pudo observar el diseño de arquitectura mundial de Stratix IV. La estructura lógica consiste en bloques de matriz lógica (LAB),

bloques de memoria y bloques de procesamiento de señales digitales (DSP). Las LABs se distribuyen simétricamente en filas y columnas y son utilizados para implementar lógica de propósito general. Los bloques DSP se utilizan para implementar los multiplicadores de precisión completa de diferentes granularidades. Los bloques tanto de memoria como los DSPsson colocados en las columnas a igual distancia entre sí. Los pines de E/Sse encuentran a lo largo de la periferia del dispositivo. Los bloques de matriz lógica (LAB) y los módulos lógicos de adaptación (ALMS) son los componentes básicos del dispositivo Stratix VI de construcción.

Pueden ser utilizados para configurar las funciones lógicas, funciones aritméticas, y registrar las funciones. Cada LAB consiste en diez ALM, llevan cadenasaritméticas, señales de control de LAB, interconexión local y registro de líneas de conexión en cadena. La estructura interna de un LAB se ilustra en la Figura 2.15.

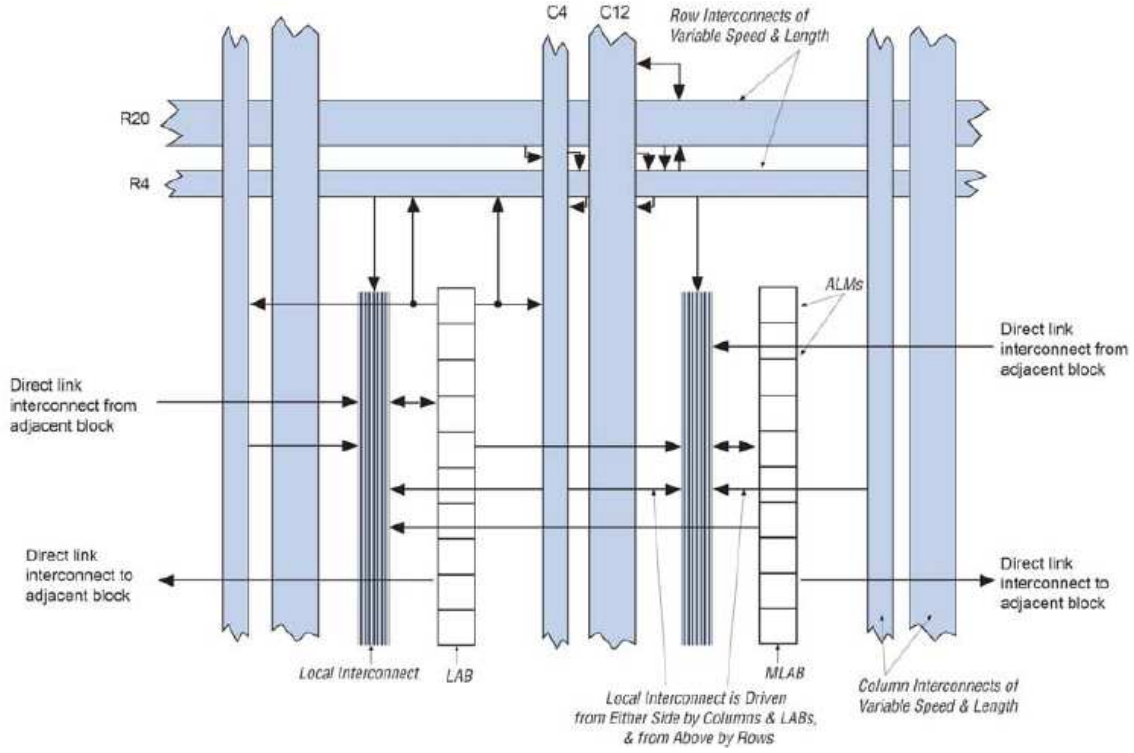


Figura 2. 15:Estructura LAB de Stratix IV.
Fuente: Jones, A. K., et al. (2005)

La interconexión local conecta lasALMsa una misma LAB. El enlace directo permite a una LAB impulsar la interconexión local, tanto del LAB izquierdo o derecho afín. La cadena de registro es conectada a la salida del registro ALM y del registro ALM adyacente en una LAB. Una memoria LAB(MLAB) es un derivado de LAB para ser utilizado como un sencillo LAB, o como una memoria estática de acceso aleatorio (SRAM).

Cada ALM en una MLAB se puede configurar como bloques de 64x1, o32x2, resultando una configuración de bloque SRAM sencilla de doble puerto de 64x10 o 32x20. MLAB y los bloques LAB siempre coexisten como pares en familias Stratix IV.Los bloques DSP en Stratix IV son optimizados para aplicaciones de procesamiento de señales, tales como Respuesta Finita de Impulso (FIR), Respuesta de Impulsos Infinitos (IIR), Transformada Rápida de Fourier (FFT) y codificadores, etc.

Los dispositivos StratixIV tienen desde dos hasta siete columnas de bloques DSP que pueden implementar: multiplicación, añadir multiplicación, acumular multiplicación (MAC) y el cálculo dinámico o funciones de cambio lógicas. El bloque DSP soporta operaciones de multiplicación de 9x9, 12x12, 18x18 y 36x36. Los dispositivos Stratix IV contienen tres diferentes tamaños de SRAM integrados.

Los tamaños de memoria incluyen bloques de 640 bits de matriz lógica de memoria (MLABs), 9Kbit de bloques M9K y de 144 Kbit de bloques M144K. Los MLABs han sido optimizados para implementar filtros de líneas de retardo, pequeños búferes FIFO, y registros de desplazamiento. Los bloques M9K pueden ser utilizados para aplicaciones de memoria de propósito general, y los bloques M144Ksuelen utilizarse para almacenar códigos para un procesador, para almacenamiento de paquetes en búfer.

2.8. Altera HardCopy

Altera da provisión para migrar las aplicaciones basadas en la FPGA a ASIC estructurado. Su ASIC estructurado se denomina HardCopy. El tema principal es diseñar, probar e incluso enviar inicialmente un diseño utilizando un la FPGA. Más tarde, el circuito de aplicación que se asigna en la FPGA se puede perfectamente emigrar a HardCopy para la producción de alto volumen.

Los últimos dispositivos de HardCopyIV, ofrecen compatibilidad pin a pin con el prototipo Stratix IV, haciendo caeren los reemplazos para las FPGAs. Por lo tanto, la misma placa base y softwares desarrollados para los ensayos de prototipos y de campo pueden ser retenidos, permitiendo el menor riesgo y tiempo de comercialización más rápido para la producción de alto volumen. Por otra parte, cuando un circuito de aplicación migra desde el prototipo Stratix IV FPGA para HardCopyVI, doblan en rendimiento de la lógica de núcleo y el consumo de energía se reduce en un 50%.

Los elementos básicos delógica en una FPGA basadas en SRAM comprenden de LUTs y flip-flops. La funcionalidad lógica se implementa en LUTs que están opcionalmente registradas. Por otro lado, la unidad lógica básica de HardCopy se denomina como HCell. Es similar a la celda lógica la FPGA (LAB) en el sentido de que el tejido consiste en un patrón regular que está formado por una o más celdas básicas en una matriz de dos dimensiones. La diferencia es que HCellno trabaja en la configuración.

Una matriz de HCellsy una red de enrutamiento de propósito general que las interconecta, se establecen en las capas inferiores de un chip. En la figura 2.16 se ilustra la comunicación entre un plano y la FPGA estructurada de base ASIC compatible. Existe una comunicación uno a uno para el nivel de diseño entre las MRAM, bucles de enganche de fase (PLL), memorias incrustadas, transceptores, y bloques de E/S.

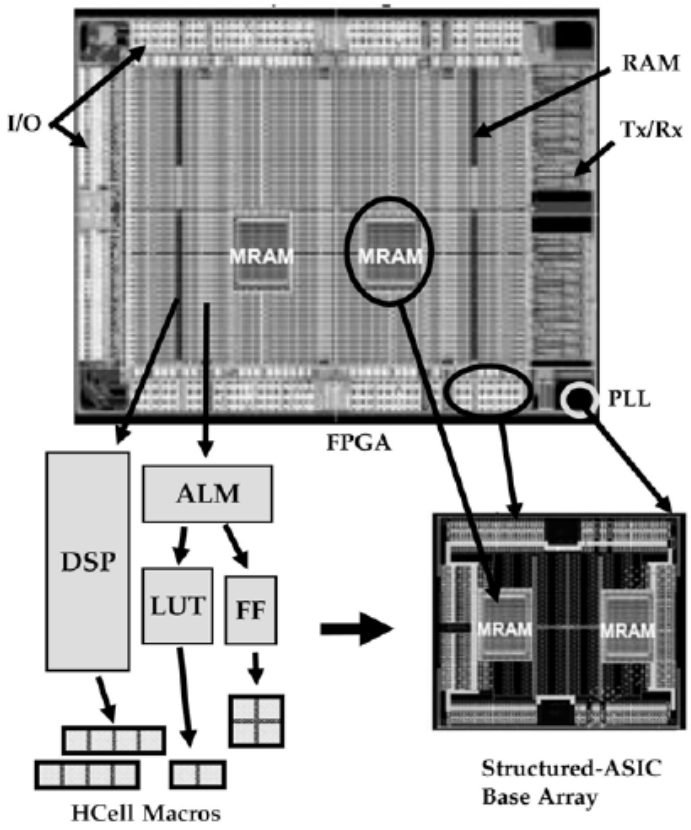


Figura 2. 16: Estructura LAB de Stratix IV.
Fuente: Jones, A. K., et al. (2005)

2.9. Núcleos ASIC configurables (cASIC)

Los cASICs son dispositivos reconfigurables que pueden implementarse en un conjunto limitado de circuitos, que operan a veces mutuamente excluyentes. Los cASICs están concebidos como aceleradores de los sistemas específicos de dominio en un chip, y no están diseñados para reemplazar totalmente el único chip ASIC. El anfitrión ejecutará códigos de software, mientras que las secciones de cálculo intensivo se pueden descargar a una o más cASICs.

Por esa razón, los cASICs implementan solamente circuitos de ruta de datos y por lo tanto sólo es compatible con bloques de texto completo (como multiplicadores de 16 bits de ancho, sumadores, RAMs, etc.). Dado que el

conjunto de circuitos soportados por un cASICson limitados y son significativamente más pequeños que una aplicación FPGA. Como los recursos de hardware son compartidos entre diferentes listas de conexiones, los cASICs son inclusive más pequeños que la suma de las áreas de una ASIC basados en células estándar de circuitos individuales.

La generación automática de núcleos cASICocurre en dos fases. La fase lógica determina las necesidades de computacionales de un conjunto de conexiones de aplicación. Los diferentes componentes computacionales son generados, incluyendo ALU, RAM, multiplicadores, registros, etc. Estos recursos lógicos son compartidos por todas las listas de conexiones de aplicación. Los componentes lógicos se ordenan adecuadamente a lo largo del camino de datos unidimensional de manera que requiera un mínimo de recursos de enrutamiento.

La fase de enrutamiento crea hilos y multiplexores para conectar los componentes lógicos y bloques de E/S. El objetivo de la fase de enrutamiento, es minimizar el área mediante el intercambio de hilos entre las diferentes listas de conexiones. Diferentes algoritmos heurísticos son utilizados para maximizar el intercambio de hilos. Los experimentos muestran que el hardware ASIC configurable es en promedio 12.3 veces más pequeña que una solución FPGA con multiplicador incorporado, y 2.2 veces más pequeña que una implementación estándar de células de los circuitos individuales.

2.10. La comparación de arquitecturas FPGA

Las FPGAsdebe ser programado por los usuarios para conectar los recursos del chip de la manera adecuada para implementar la funcionalidad deseada. Con los años, diversas tecnologías han surgido para satisfacer diferentes necesidades. Algunos FPGAs sólo se pueden programarse una

vez. Estos dispositivos emplean la tecnología antifusible. Los dispositivos basados en Flash se pueden programar y reprogramar de nuevo después de la depuración. Otros todavía se pueden programar de forma dinámica gracias a la tecnología basada en SRAM. Cada uno tiene sus ventajas y desventajas (véase la tabla 2.3).

Tabla 2. 3:Ventajas y desventajas de memorias SRAM, Flash y antifusible.

Feature	SRAM	Antifuse	Flash
Reprogrammable?	Yes (in-system)	No	Yes (in-system or offline)
Reprogramming speed (including erasure)	Fast	Not applicable	3X SRAM
Volatile?	Yes	No	No (but can be if required)
External configuration file?	Yes	No	No
Good for prototyping?	Yes	No	Yes
Instant-on?	No	Yes	Yes
IP security	Poor	Very good	Very good
Size of configuration cell	Large (six transistors)	Very small	Small (two transistors)
Power consumption	High	Low	Medium
Radiation hardness?	No	Yes	No

Fuente: Maliniak, D. (2009)

La mayoría de las FPGAs modernas se basan en células de configuración SRAM, que ofrecen el beneficio de reprogramación ilimitada. Cuando está encendida, se puede configurar para realizar tareas determinadas, como una prueba de la placa o del sistema, y luego reprogramarla para realizar su tarea principal. Por otro lado, sin embargo, las FPGAs basadas en SRAM deben reconfigurarse cada vez que su sistema host está encendido, y se requiere circuitería externa adicional para hacerlo.Además, los archivos de configuración utilizados para programar las FPGAs son almacenados en la memoria externa.

Las FPGAs basadas en antifusibleno son programables en el sistema, sino más bien se programan fuera de línea utilizando un programador de dispositivos. Una vez configurado el chipno puede ser alterado. Sin embargo, la configuración del dispositivo es no volátil sin necesidad de memoria externa. Además de eso, es prácticamente imposible de realizar

ingeniería inversa de su programación. A menudo trabajan como sustitutos de los ASICs en pequeños volúmenes.

Capítulo 3: Diseño de los demoduladores PSK y FSK en FPGA

3.1. Síntesis del diseño de Flujo para FPGAs.

El ciclo de diseño se inicia con la verificación del comportamiento del diagrama de bloques que estamos interesados en implementar. Por lo general, se utiliza la plataforma Simulink de Mathworks para este propósito. Existen varias herramientas de software que apoyan el diseño de los componentes individuales y la integración de sistemas para verificar el diseño mediante simulación.

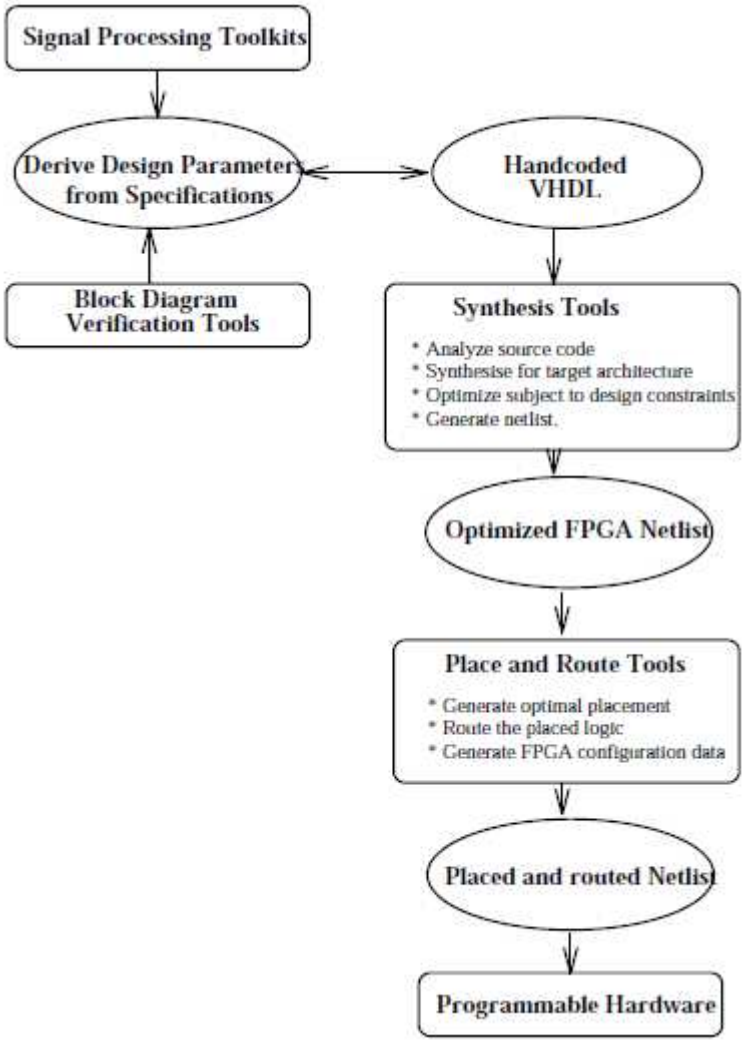


Figura 3. 1: Diagrama de la síntesis del diseño de flujo para FPGAs.
Fuente: Maliniak, D. (2009)

La síntesis consiste en analizar el código VHDL, sintetizar la arquitectura objetivo, optimizar las restricciones de diseño, así como las directivas de ubicación o especificaciones de retardo, y la generación de una lista de conexiones para optimizar la FPGA. En la figura 3.1 se explica el flujo de manera gráfica.

3.2. Demodulador BPSK.

El demodulador BPSK tiene una velocidad de datos de 100 kbps, aunque las tasas más altas son bastante factibles. La frecuencia de portadora se selecciona para que sea de 500 kHz. La señal modulada Binaria PSK se muestreaa 8 MHz y se alimenta como entrada digital para el diseño. La señal PSK tiene la característica de que la fase de la onda portadora cambia a la velocidad de datos.

La fase de la señal será uno de los valores de M para una PSK M-aria, donde las fases se diferencian por 360/M°. Para la presente implementación BPSK se tuvo que cambiar la fase de la portadora en 180°. La conmutación de fase se produce basándose en el bit de datos transmitido.Eldemodulador debería ser capaz de distinguir entre un “1” o un “0” basándose en la modulación de entrada.

3.2.1. Fundamentos teóricos de PSK.

Gil V., P., Pomares B., J., & Candelas H., F. A. (2010) manifiestan que la modulación por desplazamiento de fase (PSK)es ampliamente utilizada en la transmisión de datos y muy adecuado para las comunicaciones de datos síncronos. Según Paz P., H. (2009) para un ancho de banda ilimitado PSK da la tasa de error de bit más bajo para una energía transmitida dada por bit. También es eficiente en el uso de ancho de banda.

El sistema PSK básico, para datos binarios, transmite una de las dos fases de una señal de portadora, en función del sentido del bit transmitido. Así, un “1” setransmite por el símbolo$Acos\,(\omega_c t)$, mientras que un “0” se transmite como$-Acos\,(\omega_c t)$. La inversión de signo corresponde a un desplazamiento de fase de 180°, de ahí el nombre de modulación por desplazamiento de fase. Por tanto, la señal modulada recibida es:

$$s(t) = k * d(t)\cos(2\pi f_c t + \theta)$$

$$d(t) \in \{-1, +1\}$$

La función básica del demodulador BPSK se puede visualizar en la figura 3.2. En el receptor, una portadora de referencia se crea a partir de la señal de entrada. Esta portadora recuperada está en la misma frecuencia que la de la portadora original, pero desprovista de cualquier cambio de fase en el sentido de que la portadora de referencia tiene una fase constante. La portadora recuperada se mezcla con la señal de entrada modulada para llevar la señal a banda base. La señal de banda base se pasa por un filtro pasa bajo, que permite filtrar las componentes de alta frecuencia derivadas de mezclar el resultado. La decisión de “0 o 1” se hace sobre la base de esta salida.

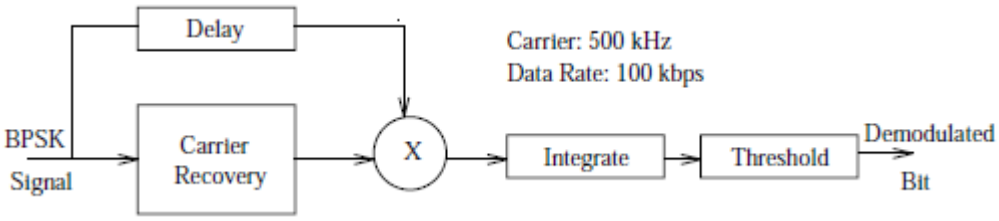


Figura 3. 2: Diagrama de bloques del modulador BPSK.
Fuente: Paz P., H. (2009)

Hay numerosos métodos para generar la portadora de referencia a partir de la señal de entrada modulada. Elegimos un sistema de recuperación de la portadora de bucle abierto para su implementación. El objetivo del esquema de recuperación de la portadora es generar una

portadora de referencia, exactamente con la misma frecuencia que la de la portadora original y que tiene una fase constante, pero puede tener una diferencia de fase constante con el soporte original. Cuando se producen cambios de datos, la compañía estará en fase de uno de los símbolos (cero o uno) y totalmente fuera de fase para el otro símbolo, de modo que cuando mezclamos las dos señales podemos tener dos niveles de amplitud, para lo cual se puede tomar una decisión.

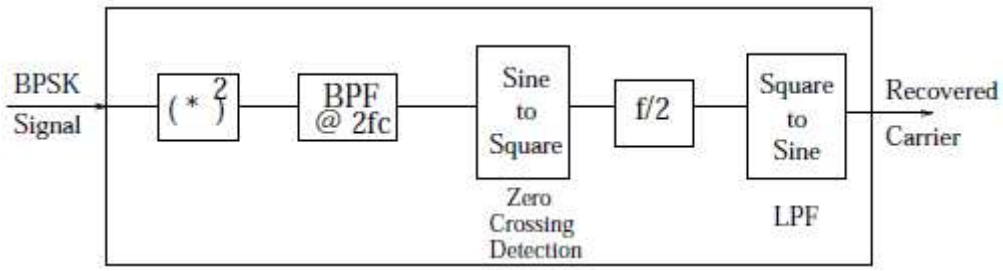


Figura 3. 3: Diagrama de bloques para la recuperación de portadora.
Fuente: Paz P., H. (2009)

En la figura 3.3 se muestra el esquema de recuperación de la portadora, para lo cual se toma la entrada como una forma de onda modulada en fase. El primer módulo de la recuperación de la portadora es un dispositivo de ley cuadrática, esto permite eliminar la modulación actual de la señal de entrada y produce algunos componentes de frecuencia a dos veces la frecuencia de la portadora. Con la aprobación de esta salida a través del filtro de paso de banda seleccionamos dos veces el componente de soporte. La salida del filtro de paso de banda será entonces:

$$c(t) = k * d^2(t) \sin(2\pi f_c t + 2\theta)$$

Para tener una referencia limpia de la frecuencia portadora, necesitamos dividir la frecuencia de la onda en la salida del filtro pasa banda. Para ello convertimos la onda senoidal en una onda cuadrada por transformación de cruce por cero, ya que es más fácil de la mitad de la frecuencia de la onda cuadrada. Entonces, tenemos una onda cuadrada de

frecuencia igual a la de la portadora. Al pasar a través de un filtro de paso bajo que tiene un punto de corte a la frecuencia portadora, podemos aprovechar el hecho de que una onda cuadrada de una frecuencia particular, se puede ver como una superposición de ondas senoidales de diferentes frecuencias y seleccionar la onda sinusoidal de la frecuencia fundamental.

La salida del circuito de recuperación de portadora siempre estará bloqueada en la frecuencia de entrada para derivas menores en la frecuencia de la entrada. La deriva del circuito puede tolerar sin perder la sincronización y dependerá de la anchura de banda del filtro pasa banda.

Tenemos entonces la modulación de fase de entrada y una portadora recuperada que tienen la misma frecuencia. Un retardo se usa para compensar el retardo a través de la línea de derivación en el circuito de recuperación de portadora, de manera que la señal de referencia puede ser exactamente en fase con uno de los símbolos. Luego multiplicamos estas dos señales. Para los intervalos de símbolos para el que la entrada está en fase con la portadora de referencia, la multiplicación dará exactamente la misma que la cuadratura de la señal de entrada produciría. La onda resultante sería todos positivos y tienen componentes de frecuencia principalmente dos veces dela portadora.

Para intervalos de símbolo para el que la entrada es totalmente fuera de fase con respecto a la portadora de referencia, la multiplicación siempre daría un resultado negativo y adquieren componentes de frecuencia similares a los del primer caso. Con la aprobación de esta salida a través de un filtro pasa bajo, una onda cuadrada de conmutación entre los lados positivos y negativos de la media puede ser detectada, está media se utiliza como un umbral para tomar una decisión bits.

3.2.2. Pormenores de implementar el demodulador BPSK

A continuación se detallan los pormenores para poder implementar la demodulación PSK.

a. Formato de los Datos.

Para este caso sería complemento a dos, que permite representar las señales en dominio digital debido a su capacidad para manejar números negativos inherentemente sin un bit de signo adicional. Los números positivos y negativos se pueden distinguir por el bit más significativo del número dado.

b. Multiplicadores de coeficiente constantes.

Una clase principal de bloques de DSP (Procesamiento de Digital de Señales) utilizado para las comunicaciones, es el filtro. La aplicación defiltros, se reduce a retrasar, multiplicar y acumular. Los multiplicadores utilizados en la mayoría de los filtros son multiplicadores reducidos, en el sentido de que uno de los operandos es fijo. Utilizando multiplicadores de coeficiente constantes en lugar de los multiplicadores generales, producen un ahorro significativo en los recursos de hardware para el circuito.

Un coeficiente multiplicador constante puede implementarse como una tabla de búsqueda. La salida se determina para todos los valores posibles de la entrada y se almacena como una memoria ROM. Por ejemplo, para una entrada de 8 bits y una constante de 8 bits, habría 256 entradas en la tabla de la memoria ROM, cada entrada debe ser 16 bits de ancho. Para tomar un enfoque práctico, utilizamos una técnica híbrida, donde la tabla de consulta se almacena sólo para$k * 0, \ldots k * F$, donde k representa la constante de entrada.

Los bits de entrada se agrupan en grupos de 4 bits. La búsqueda se realiza con cada grupo de cuatro para obtener productos parciales, que se añaden para obtener el producto final. Para el ejemplo anterior, la entrada de 8 bits se divide en dos grupos de 4 bits cada uno (el medio byte superior y el inferior), y dos consultas de tabla se realizan para obtener dos sumas parciales. El producto parcial resultante del nibble (medio byte) superior se desplaza 4 por izquierda y se añade al producto parcial resultante de la nibble inferior para obtener el producto final.

Multiplicación de complemento de dos también es fácil de implementar, mediante el uso de una tabla de búsqueda suscrita.

c. Dispositivo de ley cuadrática.

El primer bloque del programa de recuperación de la portadora es el dispositivo de ley cuadrática. Esto se implementa como un multiplicador normal. La entrada es un número de 8 bits en complementado ados. Esta es la entrada a todo el sistema como tal. Para la salida, la precisión total sería de 15 bits de ancho, pero se trunca a 8 bits para que sea ideal en cascada con el filtro de pasa banda.

El efecto de cuadratura elimina la modulación presente en la entrada. El espectro de frecuencia de la salida ideal contendría sólo la componente de corriente continua y el doble de la componente de frecuencia pero en la práctica debido a una cierta cantidad finita de corriente directa (DC) que puede estar presente en la señal de la frecuencia portadora. En la tabla 3.1 se muestra en detalle el módulo del dispositivo de ley cuadrática.

Tabla 3. 1: Especificaciones de implementación para el dispositivo.

Tipo de parte	4013PG223-5		
Uso de CLB	62 disponible de 576 ... 10%		
Señal de reloj máxima	17,75 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	1s	24s	26s

d. Filtro pasa banda

La respuesta de frecuencia de la señal que es la entrada del filtro pasa banda se muestra en la figura 3.4, donde la frecuencia de interés es el doble de la frecuencia de portadora. En este caso para una portadora de 500 kHz la frecuencia de la banda de paso debe estar centrada en 1 MHz. Para implementar esto se utiliza una ventanaRemez de intercambio.

El filtro resultante es un FIR (respuesta impulsional finita) 21-tap. El ancho de banda de 6 dB para el filtro es 500 kHz y un ancho de bandanulo 950 kHz. El diseño tiene una entrada de complemento a dos de 8 bits y da una salida de 12 bits de ancho, que también son del formato de complemento a dos.

La multiplicación en el filtro se realiza utilizando multiplicadores de coeficiente constante que no requiere de mucho espacio en comparación con los multiplicadores genéricos. En la figura 3.4 se muestra la magnitud de respuesta del filtro pasa banda tanto el teórico como el práctico (implementación).En la tabla 3.2 se muestra los detalles del diseño a implementarse.

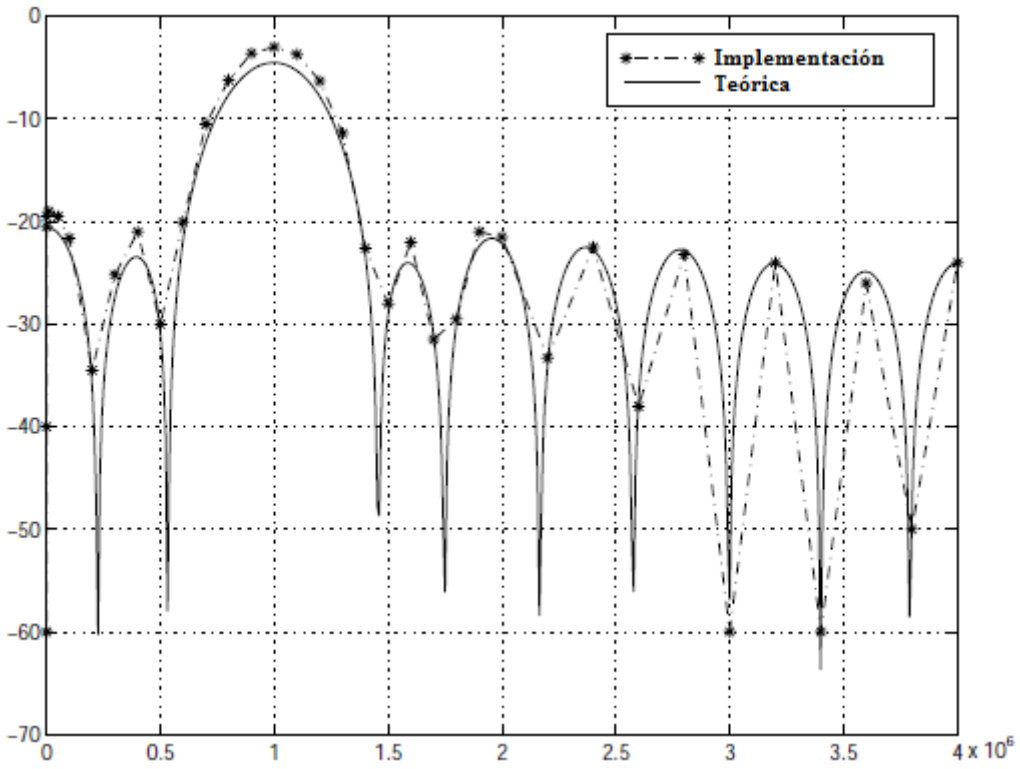


Figura 3. 4: Magnitud de respuesta del filtro pasa banda.
Elaborado: El Autor.

Tabla 3. 2: Especificaciones de implementación para el filtro pasa banda.

Tipo de parte	4013PG223-5		
Uso de CLB	351 disponible de 576 ... 60%		
Señal de reloj máxima	10,32 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	5s	181s	109s

e. Divisor de frecuencia.

La onda sinusoidal de la banda de paso se convierte en una onda cuadrada tomando el bit más significativo. Esta onda cuadrada, se utiliza el bit de la entrada al divisor de frecuencia. La lógica utilizada de la división de frecuencia funciona de la siguiente manera: el flanco ascendente de la onda de entrada se realiza el seguimiento y

la transición, desde el estado actual de la salida forzada. La salida permanece bloqueada para el borde descendente de la entrada, de ahí que se duplica el periodo de tiempo o la frecuencia dividida para dos. En la tabla 3.3 se muestran los valores que toma el divisor de frecuencia.

Tabla 3. 3: Especificaciones de implementación para el divisor de frecuencia.

Tipo de parte	4013PG223-5		
Uso de CLB	4 disponible de 576 ... 1%		
Señal de reloj máxima	74,6 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	1s	13s	1s

f. Convertidor de onda cuadrática a sinusoidal.

La salida del divisor de frecuencia es una onda cuadrada que tiene exactamente la misma frecuencia que la de la portadora. El objetivo es generar una onda sinusoidal de la onda cuadrada. En otras palabras, la entrada del convertidor de onda cuadrática a sinusoidal, es la onda cuadrada de frecuencia portadora. La entrada es solamente 1 bit y una salida de 8 bits.

La onda cuadrada se puede visualizar para ser una superposición de ondas sinusoidales de frecuencia fundamental junto con otros armónicos impares. Mediante la aplicación del filtrado pasa bajo a un espectro de frecuencia, tales producirían una onda sinusoidal de frecuencia fundamental. En lo que se refiere a la aplicación de los multiplicadores, estos pueden ser eliminados. Esto se debe a que la entrada es 1 bit y por lo tanto los términos del producto será la multiplicación de los coeficientes, ya sea con 1 o -1.

El filtro tiene una frecuencia de corte de poco un poco mayor a la frecuencia fundamental, que para el proyecto es 500 kHz. El filtro elegido para la aplicación es una ventana Kaiser 16-tap. En la tabla 3.4 se muestran las especificaciones para el convertidor.

Tabla 3. 4: Especificaciones de implementación para el convertidor.

Tipo de parte	4013PG223-5		
Uso de CLB	193 disponible de 576 ... 33%		
Señal de reloj máxima	18,9 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	1s	60s	1s

g. Recuperación de portadora.

El conjunto de la recuperación de la portadora está integrada y probada, en la tabla 3.5 se muestran las especificaciones de la implementación.

Tabla 3. 5: Especificaciones de implementación para la recuperación de portadora.

Tipo de parte	4013PG223-5		
Uso de CLB	530 disponible de 1024 ... 51%		
Señal de reloj máxima	10,14 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	8s	285s	327s

h. Filtrado de datos.

La función del filtrado de datos es para suavizar la forma de onda de banda base y rechazar las componentes de frecuencia más altas que resultan debido a la combinación o mezcla de ambas. La entrada es de 8

bits y la salida es de 12 bits. El filtro está diseñado para tener un punto de corte de 6 dB de 150 kHz y un ancho de bandanula de 500 kHz. El filtro resultante es un filtro FIR 19 –tap y la ventana utilizada fue del tipoKaiser. En la figura 3.5 se muestra la gráfica comparativa de la magnitud de respuestas del filtro diseñado y del implementado sobre una FPGA. Las especificaciones del filtrado de datos se muestran en la tabla 3.6.

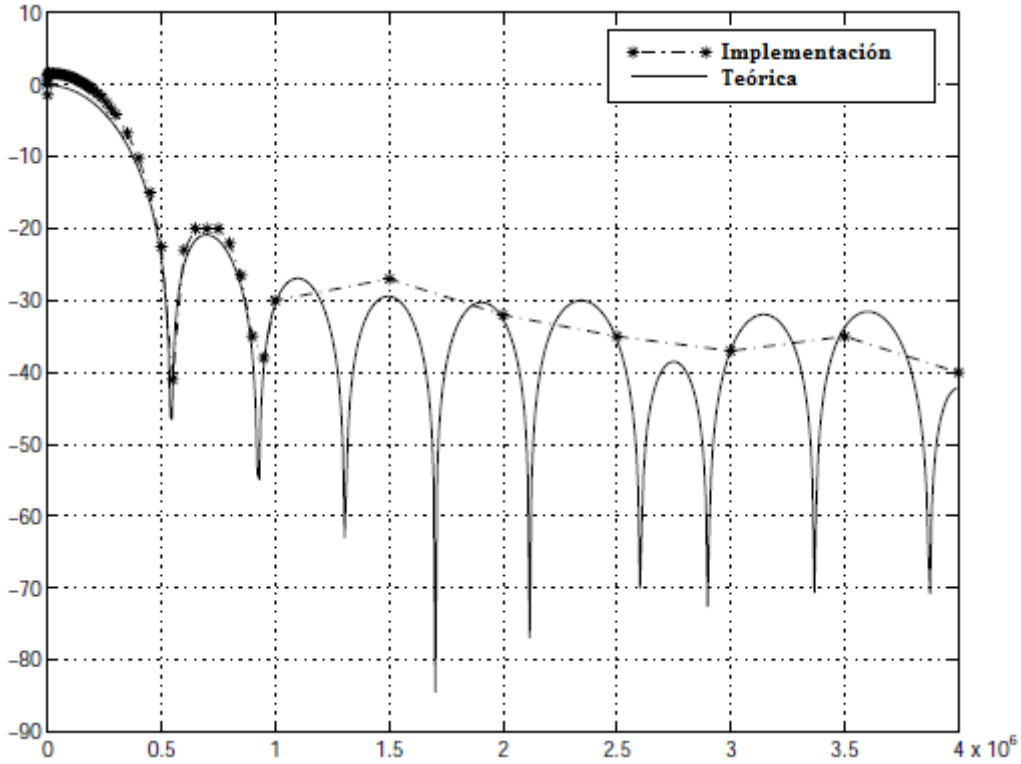


Figura 3. 5: Magnitud de respuesta del filtrado de datos.
Elaborado: El Autor.

Tabla 3. 6: Especificaciones de implementación para el filtrado de datos.

Tipo de parte	4013PG223-5		
Uso de CLB	378 disponible de 576 ... 57%		
Señal de reloj máxima	10,3 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	5s	166s	117s

3.2.3. Integración y pruebas del demodulador BPSK.

Todos los módulos anteriormente discutidos se integran en el diagrama de bloques que se muestra en la figura 3.6, que permite generar la señal modulada BPSK. Esta toma como entrada la frecuencia portadora del generador de señales, datos en serie a partir del Probador de la tasa de error binario (BERT) y proporciona una señal modulada. Esta entrada analógica se muestrea usando un convertidor A/D y la salida digital se alimenta una entrada al demodulador. La demodulación se realiza para emitir un flujo de bits de datos que se alimenta de nuevo en el terminal de datos recibido del BERT. Las especificaciones para la demodulación BPSK se muestran en la tabla 3.7.

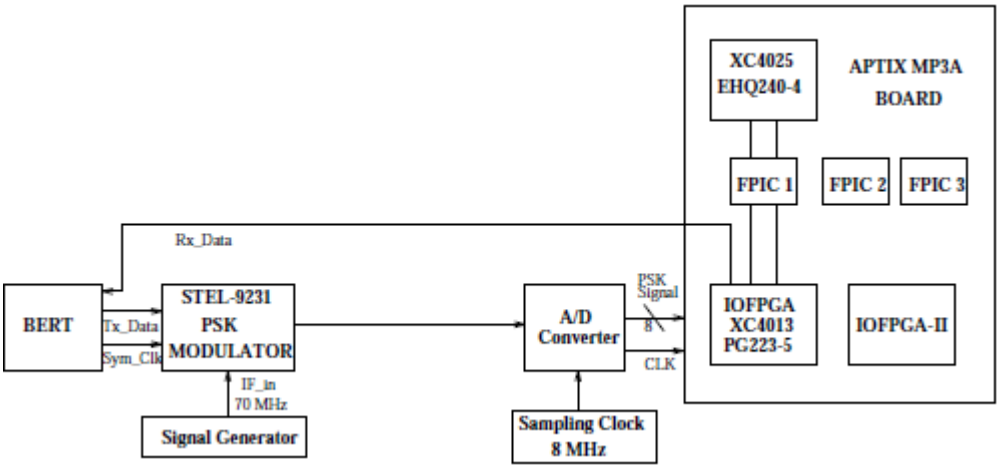


Figura 3. 6: Magnitud de respuesta del demodulador BPSK.
Elaborado: El Autor.

Tabla 3. 7: Especificaciones de implementación para el filtrado de datos.

Tipo de parte	4013PG223-5		
Uso de CLB	898 disponible de 1024 ... 87%		
Señal de reloj máxima	10,07 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	35s	704s	683s

La implementación anterior, produce una tasa de error de bits de 10^9 o superior. La prueba se llevó a cabo durante más de 3 horas sin un solo error de bit. El diseño utilizado emplea una velocidad de datos de 512 kbps, una portadora modulada a 5 MHz y muestreada a 20 MHz.

3.3. Demodulador BFSK.

Las señales FSK tienen múltiples tonos en ellos, cada tono se puede caracterizar por el número de ciclos de reloj que se tarda entre dos cruces por cero para una frecuencia de muestreo dada. Por ejemplo, una señal de 400 kHz se muestrea a 8 MHz y tendrá 20 muestras entre dos cruces por cero y para una señal de 600 kHz se tendrían alrededor de 13 muestras.

Así un contador puede reajustar para cada cruce por cero, tendría dos valores discretos en la salida. Un umbral en algún lugar entre el 20 y el 13 limitaría difícilmente la salida a niveles digitales. Por lo tanto, se debe demodular la señal FSK. El diagrama de bloques para el demodulador FSK se muestra en la figura 3.7. Para demostrar esto, los dos portadores se eligen para tener frecuencias de marca y espaciado entre 400 y 600 kHz. Para aumentar la tolerancia al ruido de los muestreos puede incrementarse para que los valores del contador estén más separados y así disminuya la probabilidad de un error.

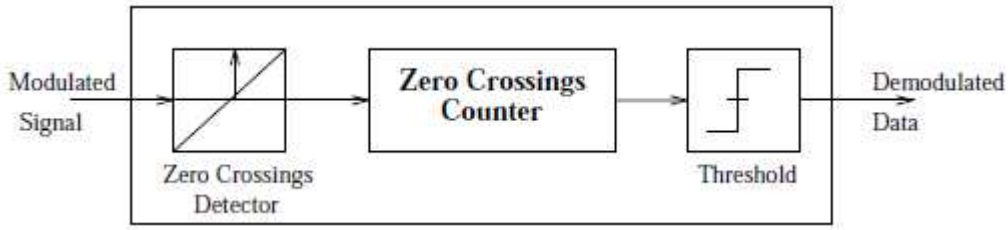


Figura 3. 7: Diagrama de bloques del demodulador BFSK.
Elaborado: El Autor.

Así como se realizó la configuración de pruebas del demodulador BPSK la del demodulador BFSK es similar su configuración. Las

especificaciones de la implementación del demodulador BFSK se muestran en la tabla 3.8

Tabla 3. 8: Especificaciones de implementación para el demodulador BFSK.

Tipo de parte	4013PG223-5		
Uso de CLB	4 disponible de 576 ... 1%		
Señal de reloj máxima	63,4 MHz		
Tiempo del CPU sobre Ultra Sparc 1	Partición	Ubicación	Enrutamiento
	1s	4s	5s

Capítulo 4: Pruebas y Resultados.

4.1. Análisis deldemoduladorBPSK.

En el presente capítulo se analizará el demodulador diseñados sobre FPGA, pero se le añadirá la presencia de ruido gaussiano blanco aditivo (AWGN). En la figura 4.1 se muestra la configuración general de prueba para la caracterización del demodulador BPSK en presencia de AWGN.

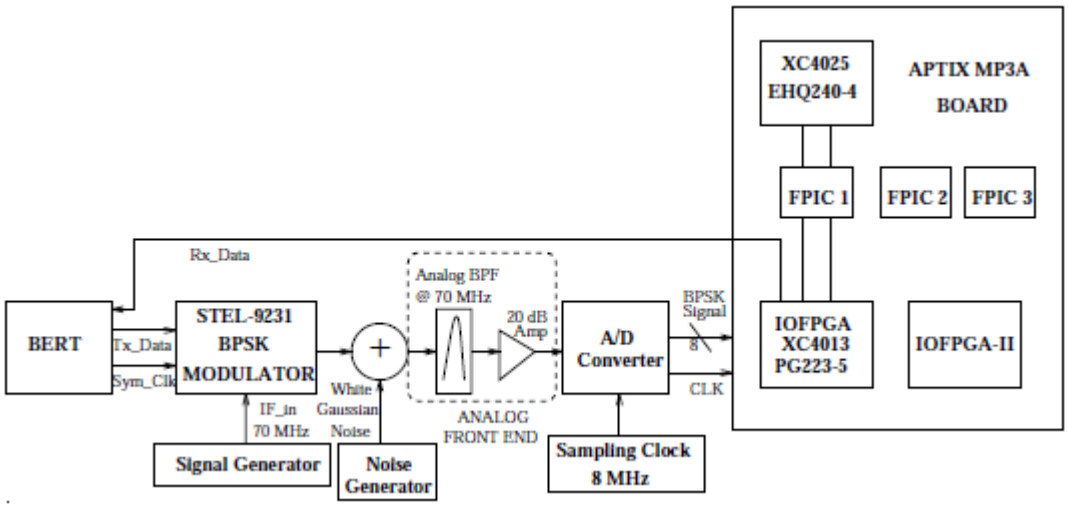


Figura 4. 1: Configuración del demodulador BPSK en presencia de AWGN.
Fuente: El Autor.

4.2. Circuitos frontales analógicos.

Con el fin de probar el receptor en presencia de ruido gaussiano blanco aditivo, fue desarrollado un extremo frontal analógico. La relación señal/ruido (S/N) se calcula en la entrada del circuito frontal analógico. El extremo comprende un filtro de paso de banda a 70 MHz con un ancho de banda de 5 MHz (ancho de banda de 3 dB) y un amplificador de 20 dB para empujar los niveles de amplitud en el rango de funcionamiento del demodulador digital.

La señal se reduce a la segunda etapa de frecuencia intermedia (IF) de 500 kHz, por un muestreo adecuado de la sub IFde 70 MHz, por un reloj de

8,75 MHz de manera que forma imágenes en cada 500 kHz y la primera imagen se utiliza para el resto del tratamiento.

4.3. Cálculo de la relación energía a ruido (Eb/No).

Estamos interesados en el trazado de la variación de la tasa de error de bit media (BER) con el cambio de la relación energía a ruido (Eb/No). El mismo se calcula de la siguiente manera:

$$E_b = PT_b$$

DondeP representa la potencia de la señal recibida de datos en vatios y No el nivel de PSD de un solo lado del ruido. Para las pruebas realizadas la potencia de señal es aproximadamente -25 dB, que es de aproximadamente -75 dBm/Hz para una velocidad de datos de 100 kbps. La potencia de ruido se varía en consecuencia a fin de obtener una proporción de 0 a 15 dB.

4.4. Comparativa entre BER y Eb/No.

Las curvas estándar BER en comparación con las curvas de Eb/No son graficadas y comparadas tanto teórica como implementada, tal como se muestran en las figuras4.2 y 4.3. La expresión teórica para la probabilidad de la BER en función de Eb/No está dada por las siguientes fórmulas:

a. Para señales BPSK:

$$P(E) = Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$$

b. Para señales QPSK:

$$P(E) = Q\left(\sqrt{\frac{E_b}{2N_o}}\right)$$

En la cual Q es la función de error.

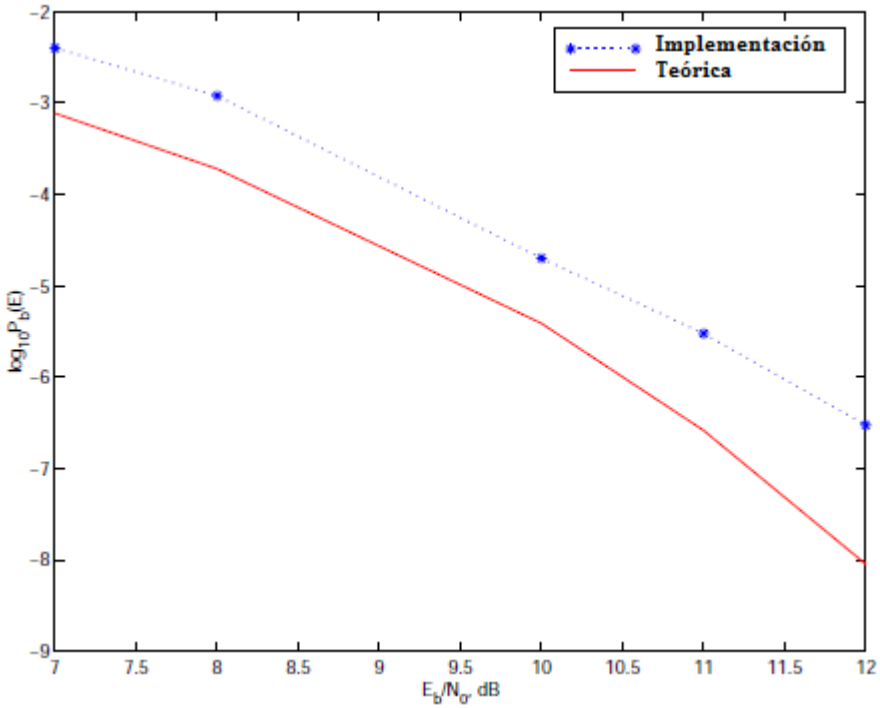


Figura 4. 2: Curvas BER y Eb/No de BPSK teórico e implementado.
Fuente: El Autor.

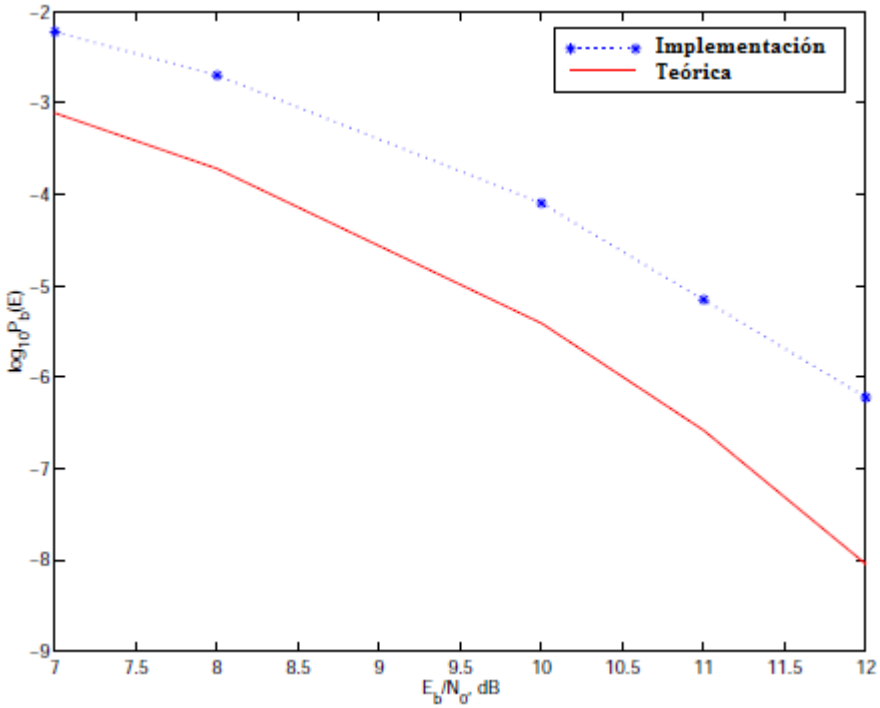


Figura 4. 3: Curvas BER y Eb/No de QPSK teórico e implementado.
Fuente: El Autor.

Para comparar el rendimiento del receptor QPSK con relación al receptor BPSK se muestran las curvas en la figura 4.4.Como era de esperar el rendimiento de receptor de BPSK es mejor que la de la QPSK.Teóricamente se espera que el rendimiento descienda en aproximadamente 3 dB de SNR, lo que significa que para la misma probabilidad de error, QPSK requiere que la SNR o Eb/No se equivalentemente a 3 dB más de lo que se necesitaría para el receptor BPSK.

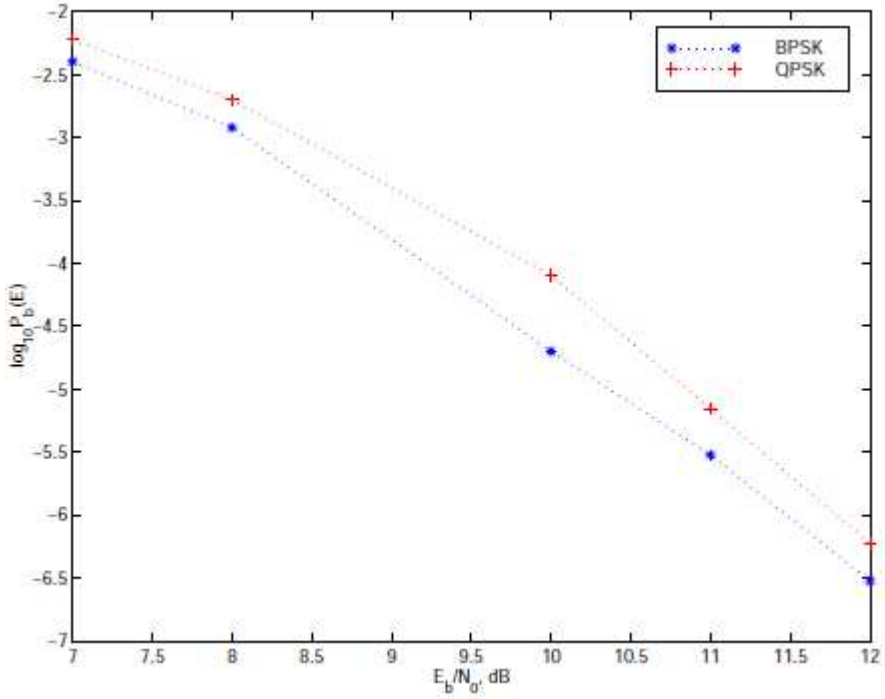


Figura 4. 4: Curvas BER y Eb/No comparadas entre BPSK y QPSK.
Fuente: El Autor.

Capítulo 5: Conclusiones y Recomendaciones.

5.1. Conclusiones.

1. Existen diversos demoduladores desarrollados por empresas internacionales prestigiosas, en nuestro país no se ha desarrollado ni construido demoduladores. El presente trabajo logró demostrar que se puede utilizar otros dispositivos electrónicos como los sistemas FPGAs para emular demoduladores PSK y FSK.

2. A través del estado del arte de FPGAs se demostró que este dispositivo permite desarrollar diferentes aplicaciones en cualquier área incluida las telecomunicaciones, aunque existen muchas tarjetas de entrenamiento FPGA la más robusta es la producida por Altera.

3. A través de las pruebas realizadas al diseño de demoduladores QPSK y QFSK se pudo observar que la FPGA permite realizar reconfiguraciones durante su ejecución, así como el esquema de modulación de entrada. Las capacidades ofrecidas por la plataforma reconfigurable se ha demostrado que puede ser una opción prometedora para un sistema de procesamiento de señal o comunicación más robusto.

5.2. Recomendaciones.

1. Mediante las plataformas de entrenamiento FPGA de Altera o de cualquier otra marca se pueden desarrollar diferentes trabajos investigativos en el área de las telecomunicaciones, tales como modulaciones analógicas y digitales, procesamiento de imágenes, sistemas OFDM, etc.

2. Adquirir más dispositivos que permitan desarrollar proyectos de investigación y que ayuden a fortalecer al programa de Maestría en Telecomunicaciones de la Universidad Católica de Santiago de Guayaquil.

Referencias Bibliográficas

Ahmed, E. & Rose, J. (2000). *The Effect of LUT and Cluster Size on Deep-submicron FPGA Performance and Density.* Proceedings of the International Symposium on Field Programmable Gate Arrays, pages 3–12

Altera (2014). Página web disponible online: http://www.altera.com

Beauchamp, M., Hauck, S., Underwood, K., &Hemmert, K. (2006) *Embedded floating-point units in FPGAs.* FPGA, pages 12–20

Betz, V., Marquardt, A., &Rose, J. (1999).*Architecture and CAD for Deep-Submicron FPGAs.*Kluwer Academic Publishers.

Compton, K. &Hauck, S. (2007). *Automatic Design of Area Efficient Configurable ASIC Cores.* IEEE Transaction on Computers, 56(5):662–672.

Cong, J., &Ding, Y.(2000).*Structural Gate Decomposition for Depth-Optimal Technology in LUT-Based FPGA Designs.* ACMTransactions on DesignAutomation of Electronic Systems, 5(3)

eASIC, (2014). Disponibleenlínea: http://www.easic.com/

Essen, B. V., Wood, A., Carroll, A., Friedman, S., Panda, R., Ylvisaker,B., Ebeling, C., &Hauck, S. (2009). Static Versus Scheduled Interconnect in CoarseGrained Reconfigurable Arrays. InternationalConference on Field Programmable Logic andApplications, pages 268–275.

Gil, P., Pomares, J.,& Candelas H., F. A. (2010). Redes y Transmisión de Datos. Textos docentes / Universidad de Alicante.

Ho, C. H., Leong, W., Luk, W., Wilton, S., &Lopez-Buedo, S. (2006). *Virtual Embedded Blocks: A Methodology for Evaluating Embedded Elements in FPGAs.*FCMM, páginas 35–44.

HardCopy IV (2012). Disponible en línea: http://www.altera.com/support/devices/hardcopy-iv/dev-hciv.jsp

Jones, A. K., Hoare, R., Kusic, D., Fazekas, J., &Foster, J. (2005).*An FPGA-based VLIW Processor with Custom Hardware Execution.*Proceedings of the International Symposium on Field Programmable Gate Arrays, pages 107–117.

Karthik, S., Shreya, P., Srihari, P., &Viswanath, N. M. (2014). *Remote Field-Programmable Gate Array (FPGA) LAB*. International Journal of Research in Engineering and Technology, IJRET. Volume: 03, Issue: 04. Páginas842 – 845.

Lemieux,G., Lee,E., Tom,M., &Yu, A. (2004).*Directional and Single Driver Wires in FPGA Interconnect.*IEEE International Conference on Field-Programmable Technology (ICFPT), pages 41–48.

Maliniak, D. (2009).*Basics of FPGA Design.*Electronic Design Automation Editor.

Marrakchi, Z. (2008). *Exploration and Optimization of Tree-based FPGA Architectures.* PhD. Thesis: http://www-asim.lip6.fr/publications/

Marrakchi, Z., Mrabet, H., Farooq, U., &Mehrez, H. (2009).*FPGA Interconnect Topologies Exploration.*Hindawi Publishing Corporation, 2598:1–13.

Miyamoto, N., &Ohmi, T.(2008), *Delay Evaluation of 90nm CMOS Multi-Context FPGA with Shift-Register-type Temporal Communication Module for Large-Scale Circuit Emulation.* IEEE InternationalConference on Field-ProgrammableTechnology (ICFPT), pages 365–368.

Paz P., H. (2009). *Sistemas de Comunicaciones Digitales*. Colección de Electrónica – Escuela Colombiana de Ingeniería.

Tabula (2011). Disponible en línea: http://www.tabula.com

Xilinx (2014). Página web Disponible online: http://www.xilinx.com

Anexo 1

Configuring Altera FPGAs

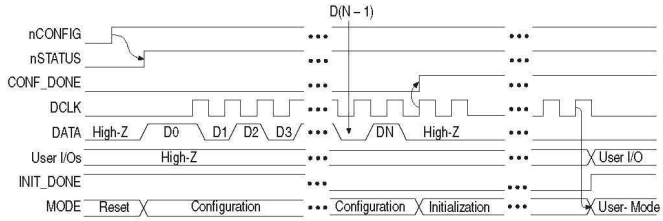
CF51001-3.2

This document describes the types of configuration schemes for Altera® FPGAs.

Device Configuration Overview for Passive Schemes

During device operation, Altera FPGAs store configuration data in SRAM cells. Because SRAM memory is volatile, the SRAM cells must be loaded with configuration data each time the device powers up. After the device is configured, its registers and I/O pins must be initialized. After initialization, the device enters user mode for in-system operation. Figure 1 shows the waveform of the configuration pins during configuration, initialization, and user mode.

Figure 1. Configuration Cycle Waveform



The low-to-high transition of nCONFIG on the FPGA begins the configuration cycle. The configuration cycle consists of 3 stages—reset, configuration, and initialization. While nCONFIG is low, the device is in reset. When the device comes out of reset, nCONFIG must be at a logic high level in order for the device to release the open-drain nSTATUS pin. After nSTATUS is released, it is pulled high by a pull-up resistor and the FPGA is ready to receive configuration data. Before and during configuration, all user I/O pins are tri-stated. Stratix® series, Arria® series, and Cyclone® series have weak pull-up resistors on the I/O pins which are on, before and during configuration.

To begin configuration, nCONFIG and nSTATUS must be at a logic high level. You can delay configuration by holding the nCONFIG low. The device receives configuration data on its DATA0 pins. Configuration data is latched into the FPGA on the rising edge of DCLK. After the FPGA has received all configuration data successfully, it releases the CONF_DONE pin, which is pulled high by a pull-up resistor. A low to high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin.

101 Innovation Drive
San Jose, CA 95134
www.altera.com

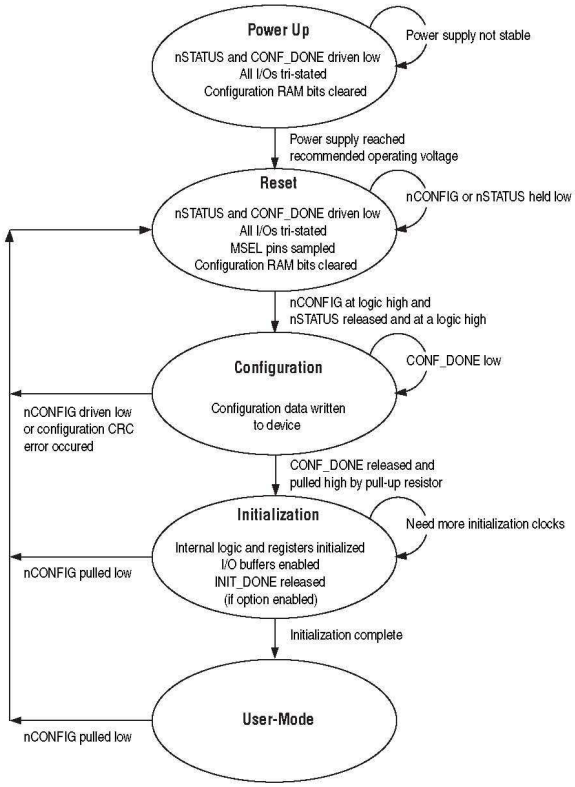
ISO 9001:2008 Registered

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user mode. During initialization, internal logic and I/O registers are initialized and I/O buffers are enabled. When initialization is finished, the INIT_DONE pin is released and pulled high by an external pull-up resistor. After entering user mode, the user I/O pins will no longer have a weak pull up and will function as assigned in your design. After configuration, you must not leave the DATA0 pins floating. Drive the pin high or low, whichever is convenient, on your board.

You can initiate a reconfiguration by toggling the nCONFIG pin from high to low and then back to high. When nCONFIG is pulled low, nSTATUS and CONF_DONE are also pulled low and all I/O pins are tri-stated. After nCONFIG and nSTATUS return to a logic high level, configuration begins.

Figure 2 shows a simple state diagram of the configuration process.

Figure 2. Configuration Cycle State Machine



Selecting a Configuration Scheme

You can load the configuration data for Altera devices using an active, passive, or JTAG configuration scheme. When using an active configuration scheme with a serial configuration (EPCS) or quad-serial configuration (EPCQ) device, the target FPGA generates the control and synchronization signals. When both devices are ready to begin configuration, the EPCS or EPCQ device sends data to the FPGA.

When you use any passive configuration scheme, the Altera device is incorporated into a system with an Altera configuration device or an intelligent host, such as a microprocessor, that controls the configuration process. The configuration device or host supplies configuration data from a storage device such as a configuration device, a hard disk, RAM, or other system memory. When you use passive configuration scheme, you can change the target device's functionality while the system is in operation by reconfiguring it.

Altera devices support a number of configuration schemes. After you have decided on the appropriate configuration scheme for your system, you need to drive the dedicated mode select control pins, `MSEL`, of the FPGA to set the configuration mode.

For more information about how to set the `MSEL` pins for your target device, refer to the configuration chapter in the appropriate device handbook.

Active Serial Configuration

You can perform an active serial (AS) configuration using EPCS or EPCQ devices. During AS configuration, the FPGA device is the master and the EPCS or EPCQ device is the slave. Configuration data is transferred one bit per clock cycle.

Passive Serial Configuration

You can perform a passive serial (PS) configuration using an Altera download cable, an Altera configuration device, or an intelligent host, such as a microprocessor. During PS configuration, configuration data is transferred from a storage device, such as a configuration device or flash memory, to the FPGA on the `DATA0` pin. This configuration data is latched into the FPGA on the rising edge of `DCLK`. Configuration data is transferred one bit per clock cycle.

Fast Passive Parallel Configuration

You can perform a fast passive parallel (FPP) configuration using an Altera configuration device or an intelligent host, such as a microprocessor. During FPP configuration, configuration data is transferred from a storage device, such as a configuration device or flash memory, to the FPGA on the `DATA[7..0]` pins. This configuration data is latched into the FPGA on the rising edge of `DCLK`. Configuration data is transferred one byte per clock cycle.

JTAG Configuration

You can perform a JTAG configuration using an Altera download cable or an intelligent host, such as a microprocessor. JTAG configuration uses the IEEE Std 1149.1 JTAG interface pins and supports the Jam™ Standard Test and Programming Language (STAPL) standard.

Document Revision History

Table 1 lists the revision history for this document.

Table 1. Document Revision History

Date	Version	Changes
September 2014	3.2	Added EPCQ device support.
August 2012	3.1	Updated Figure 1.
January 2012	3.0	Reorganizing the content for this document.
October 2008	2.2	■ Updated Table 1-1 and Table 1-2. ■ Updated Figure 1-2. ■ Updated "Introduction", "Device Configuration Overview for Passive Schemes", and "Selecting a Configuration Scheme" sections. ■ Added "Active Parallel Configuration" and "Document Revision History" sections.

Anexo 2

XILINX®

Spartan-6 FPGA Data Sheet: DC and Switching Characteristics

DS162 (v3.1) June 27, 2014 **Product Specification**

Spartan-6 FPGA Electrical Characteristics

Spartan®-6 LX and LXT FPGAs are available in various speed grades, with -3 having the highest performance. The DC and AC electrical parameters of the Automotive XA Spartan-6 FPGAs and Defense-grade Spartan-6Q FPGAs devices are equivalent to the commercial specifications except where noted. The timing characteristics of the commercial (XC) -2 speed grade industrial device are the same as for a -2 speed grade commercial device. The -2Q and -3Q speed grades are exclusively for the expanded (Q) temperature range. The timing characteristics are equivalent to those shown for the -2 and -3 speed grades for the Automotive and Defense-grade devices.

Spartan-6 FPGA DC and AC characteristics are specified for commercial (C), industrial (I), and expanded (Q) temperature ranges. Only selected speed grades and/or devices might be available in the industrial or expanded temperature ranges for Automotive and Defense-grade devices. References to device names refer to all available variations of that part number (for example, LX75 could denote XC6SLX75, XA6SLX75, or XQ6SLX75). The Spartan-6 FPGA -3N speed grade designates devices that do not support MCB functionality.

All supply voltage and junction temperature specifications are representative of worst-case conditions. The parameters included are common to popular designs and typical applications.

Available device and package combinations can be found at:

- DS160: *Spartan-6 Family Overview*
- DS170: *Automotive XA Spartan-6 Family Overview*
- DS172: *Defense-Grade Spartan-6Q Family Overview*

This Spartan-6 FPGA data sheet, part of an overall set of documentation on the Spartan-6 family of FPGAs, is available on the Xilinx website at http://www.xilinx.com/support/documentation/spartan-6.htm.

Spartan-6 FPGA DC Characteristics

Table 1: **Absolute Maximum Ratings**(1)

Symbol	Description		Units
V_{CCINT}	Internal supply voltage relative to GND	–0.5 to 1.32	V
V_{CCAUX}	Auxiliary supply voltage relative to GND	–0.5 to 3.75	V
V_{CCO}	Output drivers supply voltage relative to GND	–0.5 to 3.75	V
V_{BATT}	Key memory battery backup supply (LX75, LX75T, LX100, LX100T, LX150, and LX150T only)	–0.5 to 4.05	V
V_{FS}	External voltage supply for eFUSE programming (LX75, LX75T, LX100, LX100T, LX150, and LX150T only)(2)	–0.5 to 3.75	V
V_{REF}	Input reference voltage	–0.5 to 3.75	V

Table 1: **Absolute Maximum Ratings(1) *(Cont'd)***

Symbol	Description					Units
V_{IN} and V_{TS}(3)	I/O input voltage or voltage applied to 3-state output, relative to GND(4)	All user and dedicated I/Os	Commercial	DC	–0.60 to 4.10	V
				20% overshoot duration	–0.75 to 4.25	V
				8% overshoot duration(5)	–0.75 to 4.40	V
			Industrial	DC	–0.60 to 3.95	V
				20% overshoot duration	–0.75 to 4.15	V
				4% overshoot duration(5)	–0.75 to 4.40	V
			Expanded (Q)	DC	–0.60 to 3.95	V
				20% overshoot duration	–0.75 to 4.15	V
				4% overshoot duration(5)	–0.75 to 4.40	V
		Restricted to maximum of 100 user I/Os	Commercial	20% overshoot duration	–0.75 to 4.35	V
				15% overshoot duration(5)	–0.75 to 4.40	V
				10% overshoot duration	–0.75 to 4.45	V
			Industrial	20% overshoot duration	–0.75 to 4.25	V
				10% overshoot duration	–0.75 to 4.35	V
				8% overshoot duration(5)	–0.75 to 4.40	V
			Expanded (Q)	20% overshoot duration	–0.75 to 4.25	V
				10% overshoot duration	–0.75 to 4.35	V
				8% overshoot duration(5)	–0.75 to 4.40	V
T_{STG}	Storage temperature (ambient)				–65 to 150	°C
T_{SOL}	Maximum soldering temperature(6) (TQG144, CPG196, CSG225, CSG324, CSG484, and FTG256)				+260	°C
	Maximum soldering temperature(6) (Pb-free packages: FGG484, FGG676, and FGG900)				+250	°C
	Maximum soldering temperature(6) (Pb packages: CS484, FT256, FG484, FG676, and FG900)				+220	°C
T_j	Maximum junction temperature(6)				+125	°C

Notes:

1. Stresses beyond those listed under Absolute Maximum Ratings might cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time might affect device reliability.
2. When programming eFUSE, $V_{FS} \leq V_{CCAUX}$. Requires up to 40 mA current. For read mode, V_{FS} can be between GND and 3.45 V.
3. I/O absolute maximum limit applied to DC and AC signals. Overshoot duration is the percentage of a data period that the I/O is stressed beyond 3.45V.
4. For I/O operation, refer to UG381: *Spartan-6 FPGA SelectIO Resources User Guide*.
5. Maximum percent overshoot duration to meet 4.40V maximum.
6. T_{SOL} is the maximum soldering temperature for component bodies. For soldering guidelines and thermal considerations, see UG385: *Spartan-6 FPGA Packaging and Pinout Specification*.

Table 2: **Recommended Operating Conditions(1)**

Symbol	Description			Min	Typ	Max	Units
V_{CCINT}	Internal supply voltage relative to GND	-3, -3N, -2	Standard performance(2)	1.14	1.2	1.26	V
		-3, -2	Extended performance(2)	1.2	1.23	1.26	V
		-1L	Standard performance(2)	0.95	1.0	1.05	V
V_{CCAUX}(3)(4)	Auxiliary supply voltage relative to GND	V_{CCAUX} = 2.5V(5)		2.375	2.5	2.625	V
		V_{CCAUX} = 3.3V		3.15	3.3	3.45	V
V_{CCO}(6)(7)(8)	Output supply voltage relative to GND			1.1	–	3.45	V
V_{IN}	Input voltage relative to GND	All I/O standards (except PCI)	Commercial temperature (C)	–0.5	–	4.0	V
			Industrial temperature (I)	–0.5	–	3.95	V
			Expanded (Q) temperature	–0.5	–	3.95	V
		PCI I/O standard(9)		–0.5	–	V_{CCO} + 0.5	V
I_{IN}(10)	Maximum current through pin using PCI I/O standard when forward biasing the clamp diode.(9)		Commercial (C) and Industrial temperature (I)	–	–	10	mA
			Expanded (Q) temperature	–	–	7	mA
	Maximum current through pin when forward biasing the ground clamp diode.			–	–	10	mA
V_{BATT}(11)	Battery voltage relative to GND, T_j = 0°C to +85°C (LX75, LX75T, LX100, LX100T, LX150, and LX150T only)			1.0	–	3.6	V
T_j	Junction temperature operating range	Commercial (C) range		0	–	85	°C
		Industrial temperature (I) range		–40	–	100	°C
		Expanded (Q) temperature range		–40	–	125	°C

Notes:

1. All voltages are relative to ground.
2. See *Interface Performances for Memory Interfaces* in Table 25. The extended performance range is specified for designs not using the standard V_{CCINT} voltage range. The standard V_{CCINT} voltage range is used for:
   - Designs that do not use an MCB
   - LX4 devices
   - Devices in the TQG144 or CPG196 packages
   - Devices with the -3N speed grade
3. Recommended maximum voltage droop for V_{CCAUX} is 10 mV/ms.
4. During configuration, if V_{CCO_2} is 1.8V, then V_{CCAUX} must be 2.5V.
5. The -1L devices require V_{CCAUX} = 2.5V when using the LVDS_25, LVDS_33, BLVDS_25, LVPECL_25, RSDS_25, RSDS_33, PPDS_25, and PPDS_33 I/O standards on inputs. LVPECL_33 is not supported in the -1L devices.
6. Configuration data is retained even if V_{CCO} drops to 0V.
7. Includes V_{CCO} of 1.2V, 1.5V, 1.8V, 2.5V, and 3.3V.
8. For PCI systems, the transmitter and receiver should have common supplies for V_{CCO}.
9. Devices with a -1L speed grade do not support Xilinx PCI IP.
10. Do not exceed a total of 100 mA per bank.
11. V_{BATT} is required to maintain the battery backed RAM (BBR) AES key when V_{CCAUX} is not applied. Once V_{CCAUX} is applied, V_{BATT} can be unconnected. When BBR is not used, Xilinx recommends connecting to V_{CCAUX} or GND. However, V_{BATT} can be unconnected.

Table 3: **eFUSE Programming Conditions(1)**

Symbol	Description	Min	Typ	Max	Units
V_{FS}(2)	External voltage supply	3.2	3.3	3.4	V
I_{FS}	V_{FS} supply current	–	–	40	mA
V_{CCAUX}	Auxiliary supply voltage relative to GND	3.2	3.3	3.45	V
R_{FUSE}(3)	External resistor from R_{FUSE} pin to GND	1129	1140	1151	Ω
V_{CCINT}	Internal supply voltage relative to GND	1.14	1.2	1.26	V
t_j	Temperature range	15	–	85	°C

Notes:

1. These specifications apply during programming of the eFUSE AES key. Programming is only supported through JTAG.The AES key is only supported in the following devices: LX75, LX75T, LX100, LX100T, LX150, and LX150T.
2. When programming eFUSE, V_{FS} must be less than or equal to V_{CCAUX}. When not programming or when eFUSE is not used, Xilinx recommends connecting V_{FS} to GND. However, V_{FS} can be between GND and 3.45 V.
3. An R_{FUSE} resistor is required when programming the eFUSE AES key. When not programming or when eFUSE is not used, Xilinx recommends connecting the R_{FUSE} pin to V_{CCAUX} or GND. However, R_{FUSE} can be unconnected.