

Implementación de la FFT en hardware aplicada a recepción en OFDM

César Augusto Pedraza Bonilla¹

Ángel Felipe Díaz, Damián Prieto, Oscar Espinosa y Javier Salas²

Resumen

Esta investigación ilustra la forma de implementar la transformada rápida de Fourier usando el algoritmo de Cooley Tukey, aplicado a los sistemas que requieren de recepción en OFDM. La investigación está basada en una arquitectura FPGA para obtener más rendimiento que un PC con un lenguaje de alto nivel; además, se desarrollan las diferentes partes de hardware necesarias para el cálculo de la FFT de 64 y 128 puntos, con la posibilidad de ampliarla a 256 y 512 puntos.

Palabras clave

Transformada de Fourier, dispositivo lógico programable, recepción Ofdm, hardware.

Abstract

This investigation shows the way to implement the rapidly transformed of Fourier using the algorithms of Cooley Tukey, applied to the systems that required the reception in OFDM. The investigation is based on a

1 Ingeniero Electrónico, USTA. Magíster en Ingeniería de Telecomunicaciones y de Computadores de la Universidad de los Andes. Docente de Ingeniería Electrónica y de Telecomunicaciones de la Universidad Santo Tomás.

2 Estudiantes de la Facultad de Ingeniería de Telecomunicaciones. Universidad Santo Tomás. Miembros del Semillero de Investigación INVTEL.

°FPGA architecture to obtain more efficiency than a PC with a language of high level; moreover, there are developed the different parts of hardware necessary for the reckoning of the FFT of 64 and 128 points, with the possibility to be enlarged to 256 and 512 points.

Key words

The transformed of Fourier; logical programmable device; OFDM reception, hardware.

Introducción

El procesamiento digital de señales es una de las áreas de mayor crecimiento en la industria de la computación, debido a la necesidad de sistemas de alto desempeño en aparatos de telecomunicaciones portátiles. Este concepto ha causado un auge de los sistemas en chip (SoC), que permiten que se generen bloques funcionales de hardware, listos para ser sintetizados para dispositivos lógicos programables o para sistemas en VLSI, usados en el desarrollo de sistemas de comunicaciones. Uno de estos bloques usados a menudo es el de la transformada rápida de Fourier (FFT), que permite reducir notablemente el área y el consumo de los sistemas basados en OFDM.

Muchos estándares alámbricos o inalámbricos han adaptado OFDM por su variedad de aplicaciones. Por ejemplo, es la base del estándar para ADSL (asymmetric digital subscriber line) y para DAB (digital audio broadcasting) en el mercado europeo. En el entorno de las redes inalámbricas, OFDM es el corazón del estándar IEEE 802.11a y HiperLAN/2, el cual implementa OFDM de una forma similar.

Uno de los bloques usados a menudo es el de la transformada rápida de Fourier (FTT), que permite reducir notablemente el área y el consumo de los sistemas basados en OFDM. Por esta razón, la línea de Tratamiento Digital de Señales de la Facultad de Ingeniería de Telecomunicaciones ha desarrollado un diseño que cumpla con estas expectativas.

1 Marco teórico

1.1 Multiplexación por división de frecuencia ortogonal

Esta técnica se considera una forma de espectro disperso, debido a que las transmisiones están presentes en múltiples frecuencias al mismo tiempo. Su funcionamiento está basado en la transmisión de datos por portadoras de bajo ancho de banda (100Hz-50kHz). La ortogonalidad consiste en que cada portadora tiene un número entero de ciclos sobre un período de símbolo, lo cual es equivalente a separar las portadoras a una distancia entera de la frecuencia de símbolo; con ello se obtiene la mayor eficiencia de espectro mostrado en la figura 1, ya que la energía máxima que corresponde al componente sinusoidal de cada portadora, se alinea solamente con los componentes cero de la energía de la otra señal.

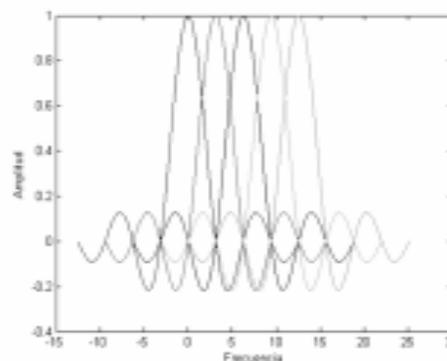


Figura 1. Espectro para OFDM

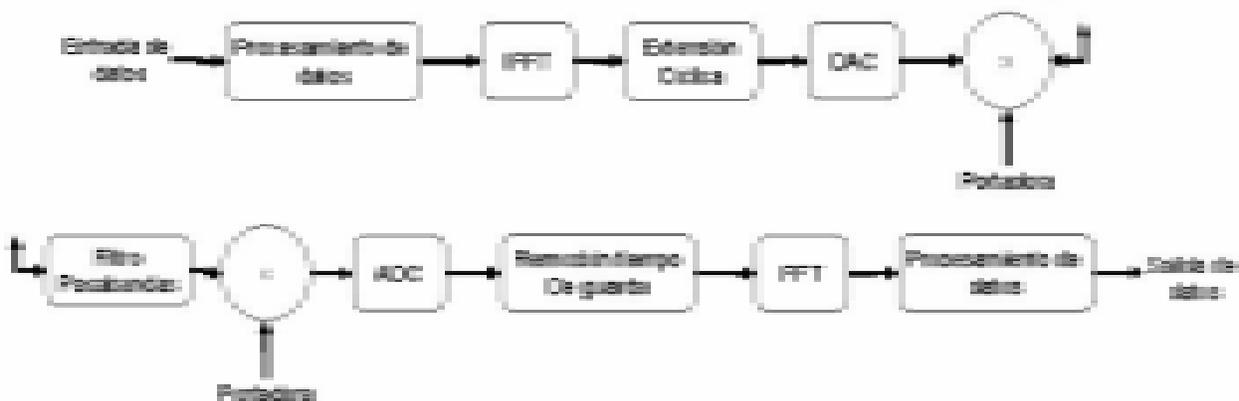


Figura 2. Diagrama de bloques simplificado de un transmisor y un receptor con OFDM.

La tecnología OFDM permite que cada portadora se ajuste independientemente a las condiciones de contorno (señal y ruido) en su zona de espectro. Al emplear un elevado número de tonos y poder ajustarlos independientemente, el sistema ofrece la posibilidad de adaptarse de forma casi óptima a las condiciones de propagación de cada enlace.

Para un alto desempeño y bajo consumo de los sistemas con OFDM, el transmisor y el receptor mostrado en la figura 2 usa la IFFT y la FFT respectivamente, que permiten reducir significativamente el consumo de potencia y el espacio empleado para el sistema.

Un problema presentado en esta técnica de transmisión es la interferencia entre símbolos ISI, la cual se debe a los multicaminos (todas las rutas que toma la señal, para viajar del transmisor al receptor), que causan, no una, sino que muchas copias de la señal lleguen al receptor, debido a las múltiples reflexiones. El problema se resuelve, agregando un intervalo de guarda, lo cual no es más que una extensión de símbolo redundante.

1.2 Algoritmo para la FFT de Cooley-Tukey

La transformada discreta de Fourier es una de las técnicas más populares para convertir señales del do-

minio del tiempo al dominio de la frecuencia. La DFT (transformada discreta de Fourier) puede ser representada mediante la ecuación 1.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jkn} \quad (1)$$

En donde $x(k)$ son las muestras de la señal en el dominio del tiempo, $X(n)$ son las muestras obtenidas en el dominio frecuencial, N es el número de puntos y W_N son los factores de giro.

La transformada rápida de Fourier (Fast Fourier Transform) es un algoritmo para el cálculo de la DFT desarrollado por Tukey y Cooley en 1965, el cual reduce el número de sumas y multiplicaciones respecto al algoritmo original.

Existen básicamente dos tipos de algoritmos para la FFT, el de diezmado en tiempo y diezmado en frecuencia.

Básicamente el algoritmo FFT toma el de la DFT y lo separa en dos partes, uno con índices pares y otro con impares. Ec 2. [2]

$$X(k) = \sum_{n=0}^{N_1-1} x(n) W_N^{nk} + \sum_{n=N_1}^{2N_1-1} x(n) W_N^{nk} + L + \sum_{n=(N_2-1)W_1}^{N_2N_1-1} x(n) W_N^{nk} \quad (2)$$

Se obtienen dos ecuaciones, cada una de las cuales contiene una FFT de menor cantidad de puntos, más exactamente $N/2$ puntos, ec 3.

$$X(k) = \sum_{n=0}^{N/2-1} x(n)W_N^{kn} - \sum_{n=N/2}^{N-1} x(n)W_N^{kn} \quad (3)$$

Dado que $W_N^2 = W_{N/2}$, entonces la ecuación 3 se puede expresar mediante la expresión 4.

$$X(k) = \sum_{n=0}^{N/2-1} x(n)W_{N/2}^{kn} - W_N^k \sum_{n=0}^{N/2-1} x(n)W_{N/2}^{kn} \quad (4)$$

Por tanto, si continuamos este proceso de división, de cada transformada, esperamos llegar a una de dos puntos, que consiste en una suma y una resta (mariposa), como se observa en la figura 3, y una multiplicación por un factor de giro para pasar a siguiente FFT. Es claro que la condición para realizar dicho proceso es que se esté realizando una FFT de 2^N puntos.

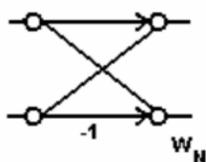


Figura 3. Cálculo de una mariposa base 2

Al desarrollar las ecuaciones mediante la FFT se observa que se deben operar las muestras o los $x(n)$ en orden de bit reverso, por ejemplo, si se realiza una FFT de 8 puntos y la primera muestra es la 0 (000), esta se opera con la 1 en bit reverso (100), es decir la 4. Así mismo, la muestra 2 (001) se opera con la 6 (110) y así sucesivamente. A dicho proceso se le denomina diezmado en tiempo, las muestras ingresan en bit reverso y salen en orden. La figura 4 muestra como se desarrolla dicho algoritmo. Los valores entre paréntesis corresponden a los exponentes de los factores de giro con $N=8$.

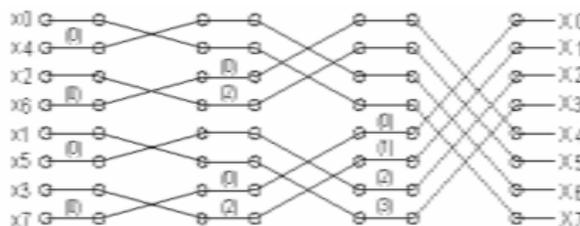


Figura 4. Cálculo de una FFT de 8 puntos con diezmado en tiempo

Ahora bien, si las muestras ingresan en orden, esperaríamos obtener las muestras en orden de bit reverso, a este caso se le denomina diezmado en frecuencia.

2 Diseño del procesador FFT

El procesador de la FFT se ha dividido en bloques de hardware, como se observa en la figura, cada uno de los cuales es descrito a continuación. La figura 5 muestra un diagrama simplificado del procesador al cual ingresan las muestras a la memoria 3, para, posteriormente, ser procesadas usando las memorias 2 y 3 para guardar los datos en los calculados en las etapas 0 al 5.

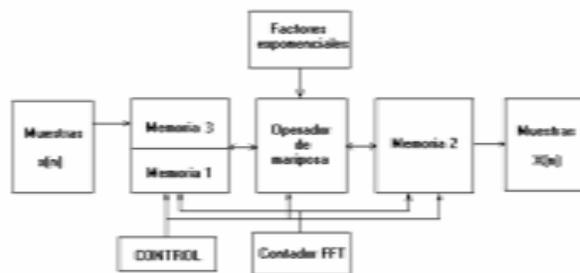


Figura 5. Diagrama de bloques del procesador FFT

2.1 Representación de punto fijo

El procesador FFT requiere la manipulación de números reales desde el momento en que se opera una muestra con un factor de giro, debido a que

este último consiste en un número complejo cuyas partes son menores o iguales a uno. Para representar los factores de giro se usan 8 bits para la parte real y ocho para la parte imaginaria (figura 6).

Bit	7	6	5	4	3	2	1	0	
Peso	Signo		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
	0	0							

Figura 6. Representación en punto fijo de los factores de giro

De igual forma se usó una precisión de 16 bits con punto fijo para representar los datos procesados. Los números negativos se trabajaron usando el complemento a dos.

2.2 Operador mariposa

Cada una de las mariposas de la transformada requiere de una multiplicación compleja, por lo que es necesario determinar las partes de un dispositivo que realice esta operación. La figura 7 muestra que se deben realizar cuatro multiplicaciones binarias y dos sumas.

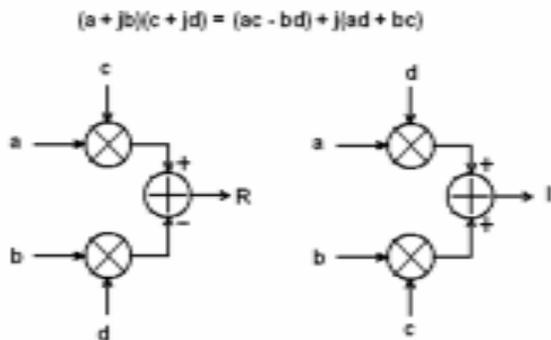


Figura 7. Partes del multiplicador complejo

Este dispositivo es uno de los que requiere más recursos de hardware para ser implementado, debido a que exige 256 operaciones lógicas and y 16 sumas

aritméticas de 16 bits por cada multiplicador, como se observa en la figura 8, por lo tanto, es conveniente que el sistema comparta un solo multiplicador que deberá ser usado para calcular todas las mariposas de la FFT [1].

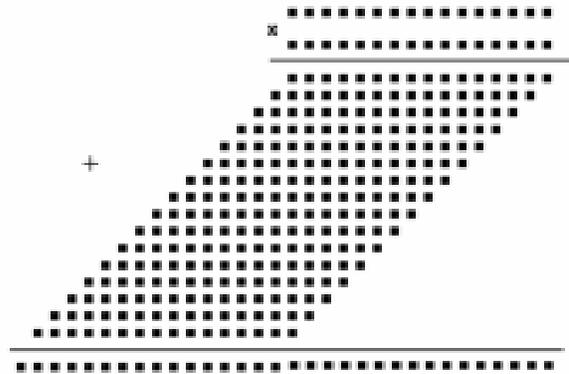


Figura 8. Operaciones de una multiplicación binaria de dos números de 16 bits

El operador mariposa consiste en un bloque de hardware encargado de realizar un cálculo de una de estas, por lo que requiere de un multiplicador complejo y dos sumadores. La figura 9 muestra los bloques requeridos para realizar este proceso.

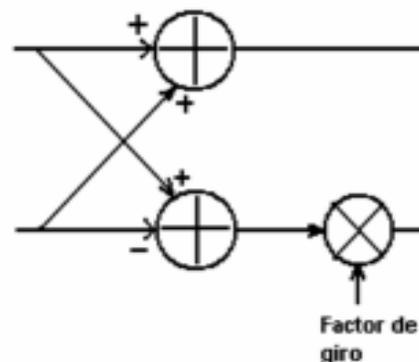


Figura 9. Operador mariposa

2.3 Bloques de memoria

Para almacenar los datos procesados de forma temporal se usan los bloques de memoria dispuestos en el FPGA. Dichos bloques pueden ser configurados de distintas formas, variando la longitud de la palabra y su cantidad de posiciones. La figura 10 muestra una de estas configuraciones.

Como se observa en la figura esta memoria es síncrona y de puerto dual, es decir, es posible leer datos y escribirlos en un mismo pulso de reloj. La configuración S16_S16 tiene la capacidad de manejar dos bloques independientes cada uno con una palabra de 16 bits y 256 posiciones [6].

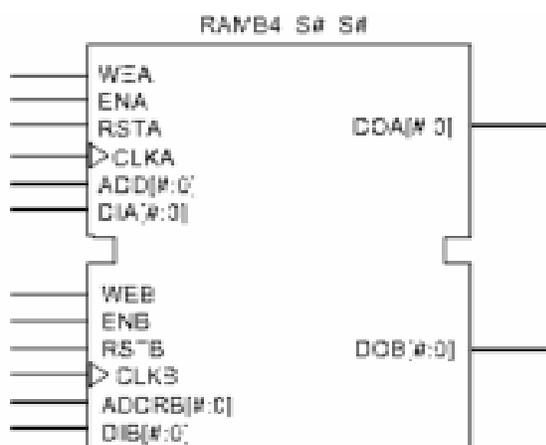


Figura 10. Bloque de memoria RAM configurado como S16_S16 [7]

Para efectos del procesador FFT se ha determinado usar dos bloques S16_S16 identificados como me-

moria 1 y 2, con el fin de guardar la parte real y compleja resultantes de cada etapa de procesamiento. Una tercera memoria de ocho bits de palabra es la que se encargará de almacenar las muestras que ingresan al procesador y que proceden de un conversor analógico digital.

2.4 Sistema de control

Habiendo determinado la forma en que se implementa los cálculos para una mariposa, se procede a diseñar el sistema que realizará los cálculos completos de la FFT.

La figura 11 muestra el diagrama de mariposa para la transformada de interés, donde se identifica claramente la necesidad de realizar seis etapas de cálculos para completar el algoritmo. La primera toma las muestras con direccionamiento de bit reverso y luego de procesarlas almacena el resultado en la memoria 2. La segunda toma los resultados de la primera etapa y calcula las mariposas una a una guardándolas en una primera memoria. Las siguientes etapas realizan la misma operación leyendo los resultados de la etapa anterior, pero teniendo en cuenta que los factores de giro son distintos.

El proceso anterior es realizado mediante un sistema de control basado en una máquina de estados, que direcciona las memorias, genera los índices de escritura y lectura de las mismas, especifica el factor de giro a usar en determinado instante de cálculo y controla la salida de las muestras de la FFT.

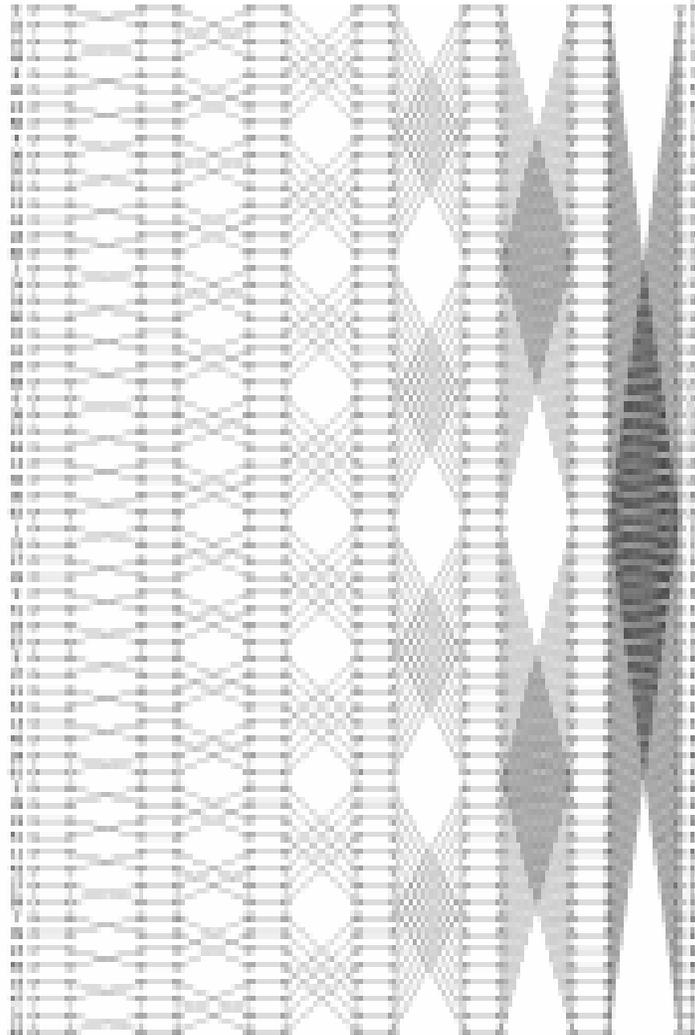


Figura 11. Diagrama de mariposa para una FFT de base 2 de 64 puntos con diezmado en tiempo

Este sistema de control tiene la característica de leer información de una memoria para procesarla, al tiempo que guarda los datos de un proceso anterior.

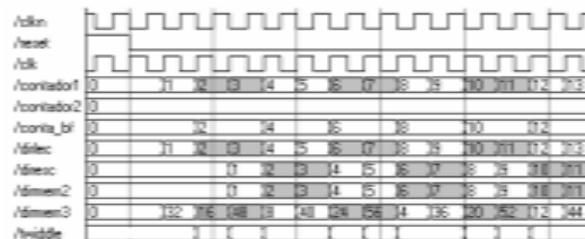


Figura 12. Simulación de parte del sistema de control

La figura 13 muestra algunas de las señales del sistema de control durante los primeros ciclos de reloj, durante los cuales se inician los cálculos de la etapa 1, y se aprecia que las direcciones de las memorias 2 y 3 se encuentran desfasadas dos ciclos de reloj, debido a que se está leyendo la memoria 3 para tomar las muestras, y luego de procesarlas mediante el operador de mariposa se almacena en la memoria 2. El bus de dirección de la memoria 3 se encuentra con los bits en reverso, razón por la cual se observan las direcciones con los valores observados en la figura 8. La señal contador2 define la etapa de la transformada que se está realizando y la señal clkn corresponde a la señal de escritura de la memoria 2.

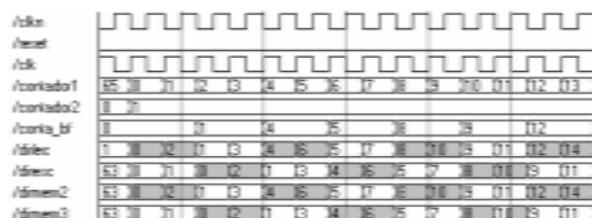


Figura 13. Simulación de parte del sistema de control

Una vez finalizada la primera etapa, la máquina de estados inicia el proceso de cálculo de la segunda. La figura 13 muestra dicho cambio, en donde la señal contador2 indica que se inicia la segunda fase de la FFT, además que el direccionamiento se realiza de dos en dos como se observa en la figura 11. De igual forma, las señales de control de escritura y de lectura se encuentran desfasadas dos ciclos de reloj para el pipelining.

2.5 Sistema de adquisición

Luego de procesar la información es necesario extraerla del FPGA con el objeto de mostrarla. Para tal efecto, se ha determinado leer la memoria donde

quedaron almacenadas la parte real e imaginaria de la FFT, representadas en 16 bits cada una, mediante el puerto paralelo de un PC. Finalmente en este se calcula la magnitud del vector complejo de cada muestra que corresponde a la magnitud de la componente frecuencial de la señal que ingresó en el dominio del tiempo.

3 Resultados

Luego de realizar simulaciones y pruebas de laboratorio resulta prometedor el desempeño del procesador FFT que se ha diseñado. Inicialmente se cuenta con un oscilador de 50MHz, lo que lleva al sistema a realizar un cálculo completo de la FFT de 64 puntos en 7.92 μ s, para un total de 126262 transformadas por segundo.

Adicionalmente, se encuentra abierta la posibilidad de aumentar la frecuencia de la señal de reloj y adicionar un nuevo pipelining en el cual se puedan calcular etapas de la FFT en paralelo, lo que requeriría más memoria y recursos del FPGA, que actualmente se encuentran disponibles.

Se realizaron pruebas en matlab que confirman la efectividad del algoritmo y los procesos usados en el FPGA. Dichas pruebas se usaron con el fin de determinar la resolución del análisis espectral obtenido. La figura 14 muestra la forma en que se realizaron unas pruebas para modular un carácter de 8 bits. Los resultados obtenidos se muestran en la figura 15, se observa la señal en el tiempo y la salida del procesador FFT de 64 bits, que revela el dato enviado (01100100) dado que se moduló la señal usando frecuencias de 1KHz hasta 8KHz para cada uno de los bits del dato. Se observa también un error en el cálculo de la FFT a causa de la precisión usada en el sistema.

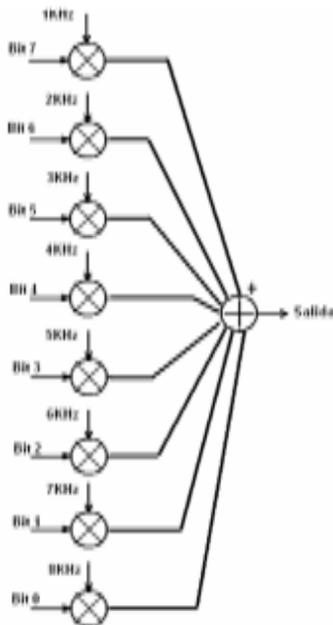


Figura 14. Esquema del modulador usado para las pruebas

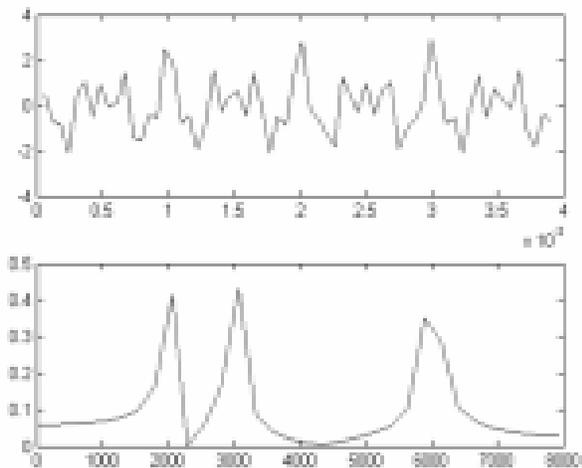


Figura 15. Simulación en tiempo y frecuencia del algoritmo implementado para la FFT de 64 puntos para el carácter 'd'.

4 Conclusiones

- Los FPGA son dispositivos que ofrecen soluciones de alto rendimiento en el diseño de sistemas electrónicos de comunicaciones, gracias al procesamiento en paralelo que estos pueden realizar. En los sistemas de comunicaciones es importante el procesamiento a gran velocidad debido a las restricciones temporales que estos exigen hoy en día en su desempeño.
- En el desarrollo de sistemas con procesamiento digital, los multiplicadores son los bloques que más recursos abarcan. Es posible reducir el hardware sacrificando velocidad de procesamiento, lo cual no es deseable para sistemas con restricciones de temporales y de espacio.
- Es evidente que el procesamiento de señales necesita de hardware muy especializado (DSP), un hardware genérico como lo es un FPGA puede reemplazar con creces uno de estos dispositivos, y en la mayoría de casos a un costo más módico, sin sacrificar velocidad de procesamiento ni complejidad de los problemas a ser resueltos.
- En el diseño de sistemas con FPGA el tiempo de desarrollo y depuración son más extensos que los demás sistemas, dado que estos dispositivos son elementos de hardware genéricos. Pero, por otro lado, estos dispositivos ofrecen la posibilidad de diseñar SoC, tendencia actual que permite integrar bloques funcionales primarios de un sistema en un solo circuito integrado, ganando espacio, velocidad y, por ende, reduce los costos de diseño.
- Como gran ventaja del desarrollo de procesadores digitales de señales en FPGA, encontramos la posibilidad de implementar siste-

mas concurrentes aumentando el horizonte de velocidad de procesamiento en tiempo real.

- Uno de los puntos críticos en el diseño de procesadores digitales de señales son las memorias. Procesar grandes cantidades de información requiere de capacidades de memorias algo considerables. Actualmente se cuenta con algoritmos

de aritmética distribuida, que minimizan el uso de los recursos combinatoriales, pero requieren así mismo de cantidades de memoria superiores. Es importante avanzar en la comprensión e investigación de algoritmos nuevos que permitan optimizar más algunos de los procesos requeridos en el procesamiento digital de señales.

Bibliografía

- BAAS, B.M. (1999). "Low-power high-performance 1024-point FFT processor". En: *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 3, March.
- Xilinx programmable databook. DS077-2 . November 15 2001.
- BAAS, B.M. (1999). "Low-power high-performance 1024-point FFT processor". En: *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 3, March.
- Xilinx programmable databook. DS077-2 . November 15 2001.
- BEKERMAN, A., OWAL, V. I, y TORKELESON, M. (2000). "A Low Depth Complex Multiplier Using Distributed Arithmetic". En: *IEEE Journal of solid state circuits*, vol. 35, pp. 656-659, Apr.
- GEIE, Jim. www.see.ed.ac.uk/~acmc/OFDMTut.html?http://oled.se.e.d.ac.uk/~acmc/OFDMTut.html.
- HASAN, M., ARSLAN T, THOMPSON J. A Novel
- KAMAI, B. S. *A hardware efficient architecture for fast fourier transform*. University of Maryland Baltimore Country. Sin publicar.
- PROAKIS, J., MANOLAKIS D. *Tratamiento digital de señales principios, algoritmos y aplicaciones*. Segunda edición. Prentice may. pp. 457-507.
- SHOUSHENG, He, TORKELESON M. (1996). "A new approach to pipeline FFT processor". En: *Proc. 10th International Parallel Processing Symposium*. IPPS April.
- SHOUSHENG, He, TORKELESON M. (1998). "Designing pipeline FFT processor for FDM (de)modulation". En: *Proc. International Symposium on Signals, systems and Electronics*. ISSSE Oct.
- SPARTAN-IIIE (2001). *1.8V FPGA Family: Functional Description*. DS077-2 (v1.0), November 15.
- TANENBAUM, A. (2003). *Redes de computadoras*, 4ta ed, Pearson, pp. 294-295.

Los autores

- César Pedraza. Ingeniero Electrónico de la Universidad Santo Tomás y magíster de la Universidad de Los Andes. Docente de la Facultad de Ingeniería de Telecomunicaciones de la Universidad Santo Tomás.
- Ángel Díaz, Óscar Espinosa, Damián Prieto y Javier Salas. Estudiantes de Ingeniería de Telecomunicaciones de la Universidad Santo Tomás. Miembros del grupo de digitales aplicado a las comunicaciones.