

Síntesis y evaluación de un DSP empotrado en una FPGA

Raidel Herrera Pereda¹, Juan Raúl Rodríguez Suárez²

¹ Universidad de Pinar del Río "Hermanos Saíz Montes de Oca", Cuba, ray@tele.upr.edu.cu, Calle Proyecto # 30 entre Alameda y Gabriel Lache; Código Postal 20100; Teléfono: 776129.

² Universidad de Pinar del Río "Hermanos Saíz Montes de Oca", Cuba, jotar@tele.upr.edu.cu, Martí # 160 Este. Pinar 1. Pinar del Río. 20100.

RESUMEN / ABSTRACT

En el trabajo se utiliza el procesador MicroBlaze, como módulo IP (*Intellectual Property*) empotrado en la tarjeta Spartan-3E de la compañía Xilinx, para realizar aplicaciones de Procesamiento Digital de Señales (PDS) tales como filtros del tipo FIR (*Finite Impulse Response*), IIR (*Infinite Impulse Response*), y de análisis espectral como la FFT (*Fast Fourier Transform*); debido a que esta tarjeta no posee embebido un procesador digital. Para ello se genera una señal a través del *software* Matlab, es enviada hacia la tarjeta Spartan-3E, se procesa con una de las aplicaciones, y los resultados son devueltos a Matlab, donde son comprobados mediante la comparación con los resultados de las funciones existentes en dicho *software* relativas a las aplicaciones dadas. La comunicación entre la PC (*Personal Computer*) y la tarjeta se hace a través del puerto serie. Los resultados de todas las aplicaciones producidas con MicroBlaze coinciden con los de Matlab.

Palabras claves: Sistemas Embebidos, FPGA, DSP, Matlab.

The MicroBlaze processor is used in the work, as a module IP embedded in the Spartan-3E board of the company Xilinx, to conduct applications of Digital Signal Processing, such as filters of the type FIR, IIR, and for spectral analysis as the FFT; due to this board does not possess embedded a digital processor. For this reason a signal is generated via the Matlab software, is sent to the board Spartan-3E, processed with one of the applications, and results are returned to Matlab, where are audited by comparison with the results of existing functions in the software relating to applications given. The communication between the PC and the board is done through the serial port. The results of all applications match those of Matlab.

Key words: Embedded Systems, FPGA, DSP, Matlab.

Synthesis and Evaluation of a DSP Embedded in a FPGA.

INTRODUCCIÓN

El Procesamiento Digital de Señales es una de las más poderosas tecnologías que puede desarrollar la ciencia y la ingeniería en el siglo veintiuno. Cambios revolucionarios han sido hechos en un gran número de campos: comunicaciones, imagen médica, sonares y radares, reproducción de música de alta fidelidad, industria petrolera, por nombrar tan solo algunos. Cada una de estas áreas ha desarrollado una tecnología DSP (*Digital Signal Processing*) profunda, con sus propios algoritmos, matemática, y técnicas especializadas. Los sistemas embebidos y aún más los circuitos de *hardware* reprogramable llamados FPGAs (*Field Programmable Gate Arrays*), han jugado y aún juegan un papel importante en este tipo de desarrollo, sobre todo en aquellas aplicaciones en que se necesite de una elevada velocidad de cómputo.

Las FPGAs de la compañía Xilinx contienen un microprocesador suave empotrado llamado MicroBlaze¹, por

ser un sistema SOC (*System on-Chip*) permite contener varios módulos IP embebidos; posee arquitectura Harvard, lo que implica buses separados para instrucciones y datos; soporta datos de formato *byte*, media palabra, palabra; es un procesador RISC (*Reduced Instruction Set Computer*), el cual aumenta su rendimiento por la segmentación (*Pipelining*); posibilita el uso tanto de interrupciones como de excepciones; presenta una unidad de punto flotante (FPU) de precisión sencilla basada en el estándar IEE 754; puede ser programado en lenguaje ensamblador o de alto nivel como C/C++. Estas características básicas hacen de MicroBlaze una herramienta eficaz para el Procesamiento Digital de Señales. Para realizar el procesamiento en tiempo real de un sistema, la salida o respuesta del mismo tiene que ser procesada en un tiempo menor que la frecuencia de muestreo del sistema, antes de que la siguiente muestra de entrada arribe al procesador.

La FPGA utilizada en el trabajo es la Spartan-3E² de la mencionada firma; la misma posee un circuito XC3S500E-4FG320, este circuito posee: más de 10 000 celdas lógicas; 20

bloques de memoria RAM (*Random Access Memory*), cada bloque contiene 18 432 bits de RAM estática rápida, para un total de bits de 368 640, y un total Kbits de 360; 20 multiplicadores *hardware* de 18x18 bits, un total de 320 terminales de conexión externos; 64 MByte de DDR (*Double Data Rate*) SDRAM; dos puertos RS-232 de nueve pines (tipo DTE y DCE); un ADC (*Analog Digital Converter*) Y DAC (*Digital Analog Converter*); ocho leds discretos; cuatro *switches*; cuatro botones; un reloj de 50 MHz; entre otras características. Sin embargo los conversores que emplea comparten líneas comunes en sus terminales de control lo que imposibilita lograr la operación simultánea de los mismos a fin de conformar un sistema de procesamiento sincrónico en tiempo real, utilizando en este caso como procesador al microprocesador empotrado suave MicroBlaze.

Específicamente se persiguen los objetivos de diseñar un procesador digital empleando el procesador empotrado MicroBlaze, conformado por la plataforma *hardware* y *software*, logrando la elaboración de códigos de programas para aplicaciones de filtrado digital, y FFT, evaluando los resultados a nivel de simulación y experimental.

METODOLOGÍA

Solución Propuesta

Para comprobar el uso eficaz de la tarjeta Spartan-3E para procesar señales digitales en tiempo real sin utilizar los conversores de dicha tarjeta; se decidió utilizar la comunicación a través del puerto serie entre la PC y el procesador embebido en la Spartan-3E. De esta forma se pueden desarrollar una serie de aplicaciones básicas de PDS, con el procesador empotrado suave MicroBlaze, mediante la programación en este caso en lenguaje de alto nivel C. Las aplicaciones desarrolladas consisten en la implementación de los algoritmos de filtros FIR, IIR, y FFT, son evaluadas en la PC a través del *software* Matlab; teniendo así una herramienta ideal con la cual comparar los resultados obtenidos. Además se pueden conocer los recursos utilizados y los tiempos de ejecución de programa, los cuales comparados con los períodos de muestreo de los conversores nos permite saber la eficacia de MicroBlaze para aplicaciones de PDS.

En la Figura 1 se muestra el diagrama en bloques del sistema constituido por la comunicación mediante el puerto serie entre Matlab en la PC, y MicroBlaze en la Spartan-3E.

Para configurar el procesador MicroBlaze se necesita hacer la selección del *hardware* del sistema, y generar los ficheros de las bibliotecas y controladores de los periféricos, donde se crean las funciones necesarias para ejecutar los mismos. Para el diseño y la evaluación de las aplicaciones de PDS, se realizaron determinadas consideraciones:



Figura 1

Diseño general para la implementación y comprobación de aplicaciones en MicroBlaze.

- Comunicación mediante el puerto serie con interrupción: se puede efectuar la recepción en MicroBlaze de datos enviados desde la PC, para ser procesados en MicroBlaze. El flujo de datos es de byte en byte (cada 8 bits), lo que obliga a concatenarlos para formar la muestra inicialmente transmitida en el formato sencillo de 32 bits, o efectuar la operación inversa a la concatenación para enviarlos hacia la PC. Memoria externa DDR RAM con capacidad de 32 MB: permite almacenar un gran volumen de datos.
- Temporizador interno con interrupción: para considerar el uso de MicroBlaze en aplicaciones de PDS en tiempo real. Para ello se realizan interrupciones cada 125 us, en este tiempo se lee el dato almacenado previamente en la DDR, se ejecuta una aplicación, y se almacena nuevamente en la DDR. De esta forma se comprueba el posible uso de MicroBlaze para aplicaciones de PDS en tiempo real con una frecuencia de muestreo del ADC de 125 us.
- Unidad de punto flotante (FPU), opción presente en el BSB (*Base System Builder*), que mejora considerable el tiempo de ejecución de programa. Se basa en el Estándar IEE-754 que incluye 1 bit de signo, 8 bits de exponente y 23 bits de mantisa.
- Conversión de formatos: los datos guardados en DDR en formato entero se convierten a formato flotante para realizar las operaciones matemáticas relacionadas con las aplicaciones *software* que se realicen, y viceversa.

En resumen se genera una señal en Matlab en la PC, se envía mediante el puerto serie hacia la tarjeta en la cual es guardada en memoria, procesada a través de la programación en lenguaje C/C++ de las aplicaciones de PDS, y finalmente el resultado es analizado correctamente en Matlab.

DESARROLLO DE APLICACIONES

Para emplear el procesador virtual MicroBlaze el usuario no tiene acceso directo al código del procesador en lenguaje de descripción de *hardware* tal como se hace en otro procesador virtual como PicoBlaze. Por ello para poder trabajar con el mismo es necesario hacer uso del paquete de programas

denominado EDK (*Embedded Design Kit*). El sistema EDK está compuesto por las siguientes partes:

- XPS (*Xilinx Platform Studio*): ambiente de desarrollo usado para diseñar la parte correspondiente al *hardware* del procesador embebido y el desarrollo de aplicaciones simples de *software*.
- SDK (*Software Development Kit*): ambiente de desarrollo integrado, complementario a XPS, que es usado para la creación y verificación de aplicaciones complejas de software en lenguaje C/C++.
- SDK es construido en el ambiente de trabajo de fuente abierta Eclipse.

Esta herramienta permite ejecutar las siguientes operaciones:

- Configurar las características del *hardware* que se necesiten del procesador.
- Generar librerías y *drivers* necesarios para el control de periféricos.
- Desarrollar aplicaciones de *software* para el procesador en lenguaje C/C++.
- Simular la ejecución de las aplicaciones, así como depurar los errores y evaluar el desempeño en el tiempo de las mismas.
- Descargar la síntesis completa del procesador (*hardware* y *software*) hacia el circuito FPGA para su ejecución.

Todas las aplicaciones de *software* de este trabajo han sido desarrolladas sobre XPS. Debe señalarse que no se puede trabajar con el sistema EDK si no está previamente instalado el entorno de desarrollo ISE (*Integrated Software Environment*) de Xilinx, puesto que la herramienta XPS hace llamadas de las funciones provistas por el *software* ISE. En el XPS tiene lugar la especificación del microprocesador, periféricos, y la interconexión de estas componentes, acompañado de sus respectivas propiedades, y la creación del fichero final para configurar el circuito programable en la FPGA.

El EDK posee una herramienta guía llamada BSB, mediante la cual se van seleccionando los distintos periféricos que se necesitan para desarrollar las aplicaciones. Los elementos seleccionados para la configuración del procesador MicroBlaze son los siguientes:

- *On-chip H/W debug module*, permite habilitar el depurado tanto en la plataforma virtual como en *hardware*. Se selecciona porque se usa el XMD (*Xilinx Microprocessor Debugger*) en conjunción con el módulo de depurado en *hardware* (opb_mdm) implementado en la compañía Xilinx.
- Memoria Local BRAM (*Block RAM*) de 32 KB, para guardar las variables de acceso rápido y permanente.
- Habilitación de Unidad de Punto Flotante, para mejorar el tiempo de ejecución de programa en los cálculos con números en ese formato.
- RS232_DCE, con el periférico OPB UARTLITE, a una velocidad de 115 200 bits por segundos, con 8 bits de

datos, ninguna paridad, y habilitada la interrupción, para efectuar la comunicación con la PC.

- LEDs_8Bit, con el periférico OPB GPIO. Para comprobar un proceso determinado del programa.
- DDR_SDRAM_32Mx16, con el periférico OPB DDR. Para el almacenamiento de grandes cantidades de datos.
- opb_timer_1, con el periférico OPB TIMER, ancho del contador de 32 bits, con un solo temporizador. Para efectuar las operaciones en el tiempo de muestreo deseado.
- Dispositivo usado como entrada y salida estándar RS232_DCE.

Una vez seleccionado el *hardware* se generan las librerías y *drivers* necesarios para el control de los periféricos, y se desarrollan las aplicaciones *software*, las cuales pueden ser simuladas en una plataforma virtual y depuradas a través de una herramienta llamada XMD del XPS.

XMD es una utilidad de diseño de *software* que facilita depurar programas que haya creado el usuario. También ayuda a verificar sistemas que usan los microprocesadores ofrecidos por Xilinx. Puede ser usada para depurar programas que corren en una tarjeta de *hardware* o que usan la Plataforma Virtual. La información del *hardware* del sistema se encuentra en un archivo llamado system.mhs, y la información del software en el archivo system.mss.

RESULTADOS Y DISCUSIÓN

Se desarrollaron tres aplicaciones básicas de PDS consistentes en filtros (del tipo FIR e IIR) y la Transformada Rápida de Fourier (FFT), programadas todas en alto nivel a través de lenguaje C/C++, con el paquete de software EDK que provee la firma Xilinx para la programación de MicroBlaze.

Filtros FIR

Los filtros FIR³ son sistemas en los cuales la salida depende solamente de cierto número de entradas retardadas. La ecuación en diferencias para un filtro FIR de orden M y forma directa está dada por la ecuación (1)

$$y(n) = h_0x(n) + h_1x(n-1) + \dots + h_Mx(n-M) \quad (1)$$

con respuesta impulsiva. $h = h_0, h_1, \dots, h_M$

Para evaluar el comportamiento de un filtro FIR se ha desarrollado una aplicación con coeficientes calculados a partir de la herramienta de diseño FDATAOOL de Matlab. En la Figura 2 se presenta la respuesta de frecuencia calculada para este filtro con dicha herramienta, que resulta en un filtro de orden 4 con coeficientes definidos en la Tabla 1.

El filtro FIR programado en MicroBlaze en lenguaje C estará definido por estos coeficientes, y estará caracterizado por una ecuación en diferencias como la (1). Además se usan las funciones de biblioteca para la habilitación e interrupción del

temporizador. Se muestra en la Figura 3 la comparación entre el valor absoluto de la respuesta de frecuencia teórica de este filtro simulada con Matlab, y la respuesta de frecuencia práctica obtenida a partir de aplicar al filtro una señal tipo seno de amplitud uno, a diferentes frecuencias normalizadas. La respuesta en magnitud de los resultados del filtrado se encuentra en la banda de paso del filtro. Es importante aclarar que las magnitudes mostradas con los marcadores de datos son el resultado de la división entre la amplitud teórica del filtro y la amplitud obtenida con el filtrado en MicroBlaze. La adecuada concordancia de ambas respuestas demuestra el correcto funcionamiento del filtro FIR elaborado en el procesador MicroBlaze. En la Figura 4 (a) y (b) se muestran las señales en el tiempo a la salida del filtro digital descrito, calculadas por la función *filter* de Matlab y las calculadas por el microprocesador MicroBlaze, en ellas se aprecia que los resultados son satisfactorios pues las señales son similares.

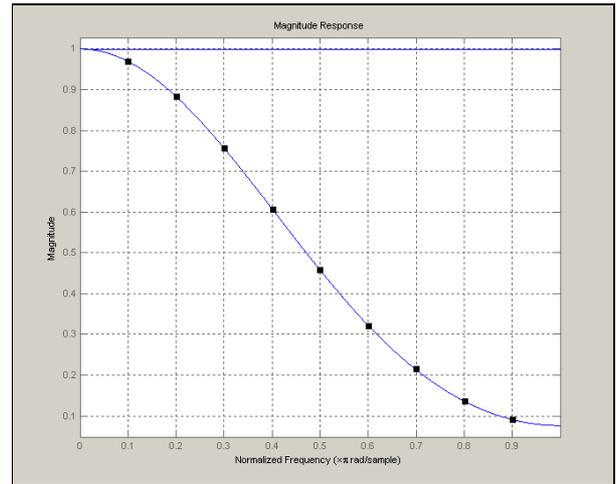


Figura 3

Respuesta en magnitud del filtro FIR.

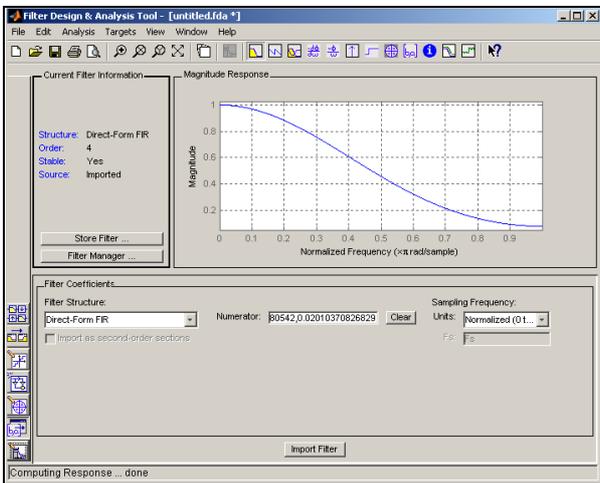


Figura 2

Filtro de forma directa I FIR, en la herramienta fdatool de Matlab.

Tabla 1: Coeficientes del filtro FIR

	Numerador
Coeficientes	0.020103708268289999 0.23086668180542 0.49805921985259 0.23086668180542 0.020103708268289999

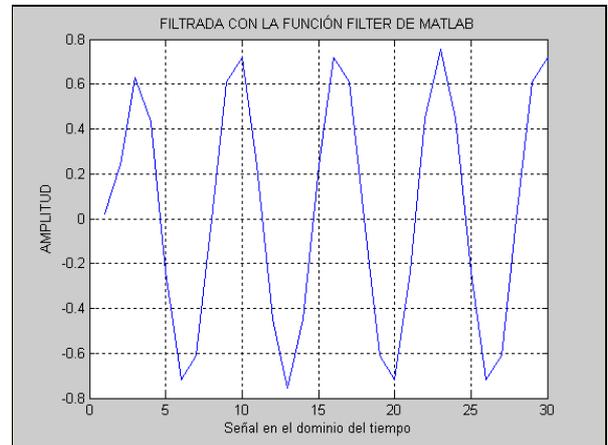


Figura 4 (a)

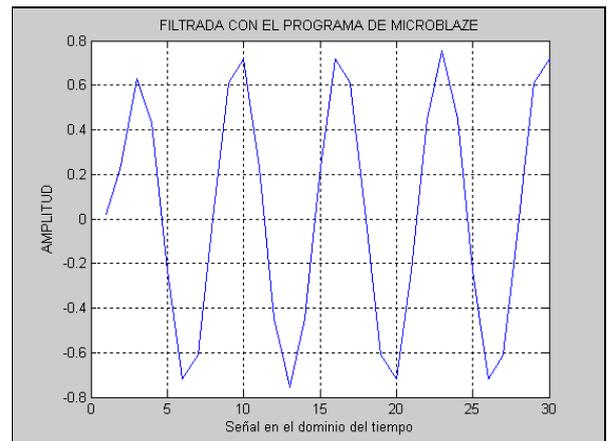


Figura 4 (b)

Figura 4 (a) y (b): Resultado de filtrar con la función *Filter* de Matlab y con el programa de MicroBlaze, respectivamente.

El tiempo de procesamiento empleado por el procesador MicroBlaze en el cálculo de la ecuación en diferencias en este sistema discreto determinado por los coeficientes del filtro es 1.95 us, menor al período de muestreo del ADC de 125 us, por lo que sería efectiva esta aplicación en tiempo real.

Filtros IIR

Los filtros digitales de respuesta al impulso infinita (IIR)⁴, o también llamados filtros recursivos procesan cada muestra de salida dependiendo de determinado número de muestras de entradas y además de salidas previas⁵ como se ve en la ecuación (2).

$$y[n] = \sum_{l=0}^L b_l x[n-l] - \sum_{m=1}^M a_m y[n-m] \quad (2)$$

Para evaluar el comportamiento de un algoritmo de filtrado IIR se ha desarrollado una aplicación a partir de un filtro IIR con coeficientes calculados con la herramienta de diseño FDATOOL de Matlab, con las siguientes características:

- Filtro IIR Discreto en el tiempo, (real) del tipo NOTCH.
- Frecuencia normalizada de corte de 0.2.
- Ancho de banda de la banda de rechazo de 0.033.
- Estructura del filtro: forma directa II.
- Longitud del numerador: 3.
- Longitud del denominador: 3.
- Estable: si.
- Fase lineal: no.

En la Figura 5 se presenta la respuesta de frecuencia calculada para dicho filtro y los coeficientes respectivos dados por la Tabla 2.

Tabla 2: Coeficientes del filtro IIR.

	Numerador	Denominador
Coefficientes	0.974278661051 -1.57641598809 0.974278661051	1 -1.57641598809 0.948557322103

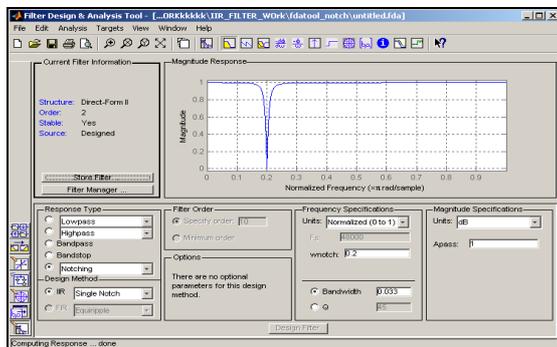


Figura 5

Filtro IIR elaborado con la herramienta fdatool del Matlab.

En la Figura 6 se muestra la comparación entre el valor absoluto de la respuesta de frecuencia de este filtro calculada teóricamente en Matlab, y la respuesta de frecuencia calculada a partir de diversas corridas del programa en MicroBlaze a partir de variar la frecuencia discreta de una secuencia del tipo seno que se puede ver en la Figura 7, aplicada a la entrada del filtro en cuestión.

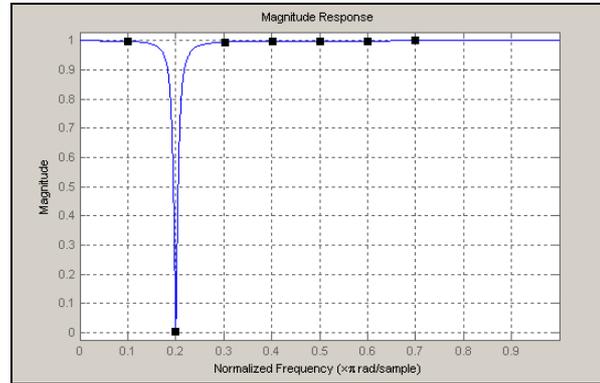


Figura 6

Respuesta de frecuencia del filtro IIR en la herramienta fvtool.

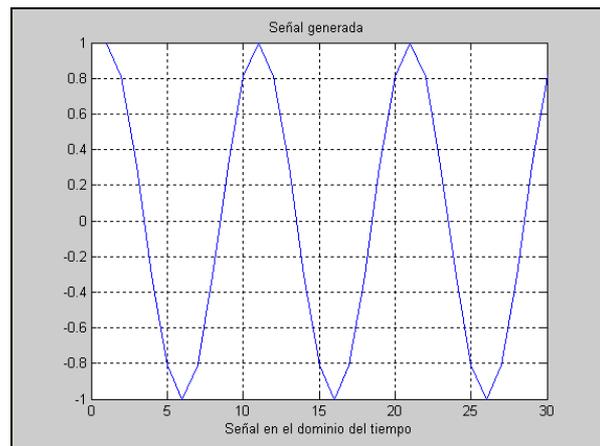


Figura 7

Señal generada en Matlab de frecuencia normalizada de 0.2 y amplitud 1.

En la misma se observa la correspondencia entre la curva teórica y la calculada por MicroBlaze; se atenúa la señal en la frecuencia central de 0.2π y dejando pasar selectivamente las demás frecuencias.

Las Figura 8 (a) y (b) muestran el resultado de haber utilizado la función *Filter* de Matlab para efectuar el filtrado de la señal sinusoidal generada en Matlab a la frecuencia normalizada de 0.2 y amplitud 1 de la Figura 7, y el resultado de haber empleado la función para el filtro IIR elaborada en MicroBlaze respectivamente. El resultado es satisfactorio,

pues las señales son similares, y se puede observar la atenuación de la señal a la frecuencia normalizada de 0.2.

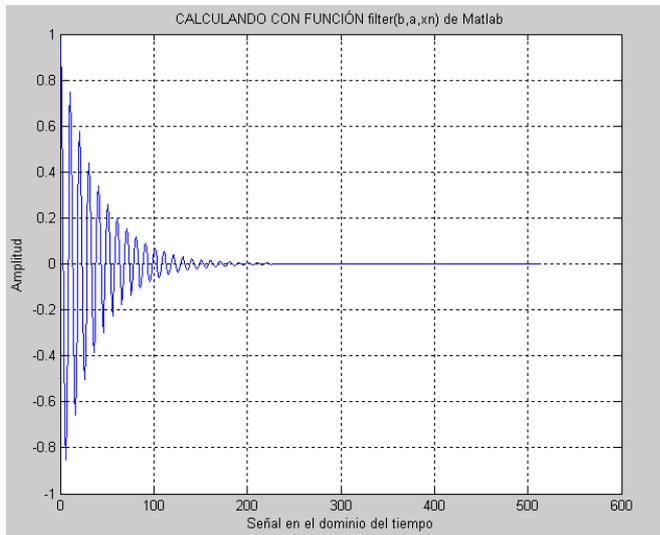


Figura 8 (a)

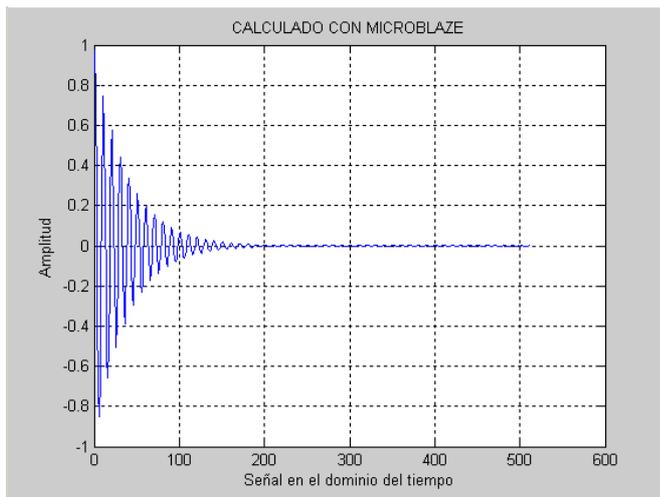


Figura 8 (b)

Figura 8 (a) y (b)

Resultados de filtrar una señal con la función *filter* y con MicroBlaze respectivamente.

La función implementada en MicroBlaze se ejecuta muestra a muestra, y su tiempo de ejecución es constante e igual a 1.96 microsegundos para cada vez que se llame, siendo posible su implementación en tiempo real utilizando un ADC con un período de muestreo de 125 us.

Análisis Espectral o en Frecuencias

Para evaluar el contenido espectral⁵ de una secuencia $x(n)$ necesitamos calcular la transformada de Fourier de dicha secuencia. En la práctica para realizar el análisis en frecuencia de una cierta secuencia digital $x(n)$ finita, con N puntos, se emplea la Transformada Discreta de Fourier (DFT), pero para grandes series esto puede tomar un tiempo considerablemente grande de cómputo, este tiempo es proporcional al cuadrado del número de puntos en las series. Un algoritmo mucho más rápido fue desarrollado por Coley y Tukey llamado la FFT o Transformada Rápida de Fourier. La versión de la FFT usada en este proyecto es llamada FFT por diezmo en tiempo, debido a que el algoritmo es basado en la máxima “divide y vencerás” que es aplicada en el dominio en el tiempo. La DFT de N puntos es primero dividida en dos DFTs de $N/2$ -puntos. Estas DFTs son entonces divididas en cuatro DFTs de $N/4$ -puntos, y así sucesivamente.

En este trabajo se parte del reporte de Bourke⁶. Este programa consiste en dos partes principales, una relacionada con el diezmo en el tiempo, y otra en la operación mariposa. Se creó una aplicación con MicroBlaze, la cual recibe muestras mediante RS232, provenientes de una señal sinusoidal generada en Matlab, de frecuencia normalizada 0.3 y amplitud 1. Las muestras se guardan en la DDR, posteriormente se ejecutan las operaciones de reordenamiento y operación mariposa, los resultados son guardados en la DDR, y luego enviados hacia Matlab para ser analizados.

Las Figura 9 (a) y (b) muestran el resultado de haber utilizado la función FFT de Matlab, y el resultado de haber utilizado el procesador suave MicroBlaze para efectuar el cálculo de la FFT a la misma señal. Como se puede apreciar las respuestas espectrales son similares, correspondientes a una señal sinusoidal.

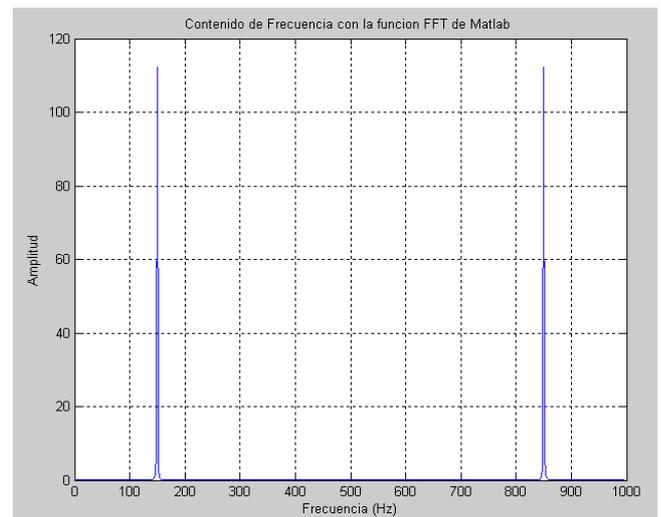


Figura 9 (a)

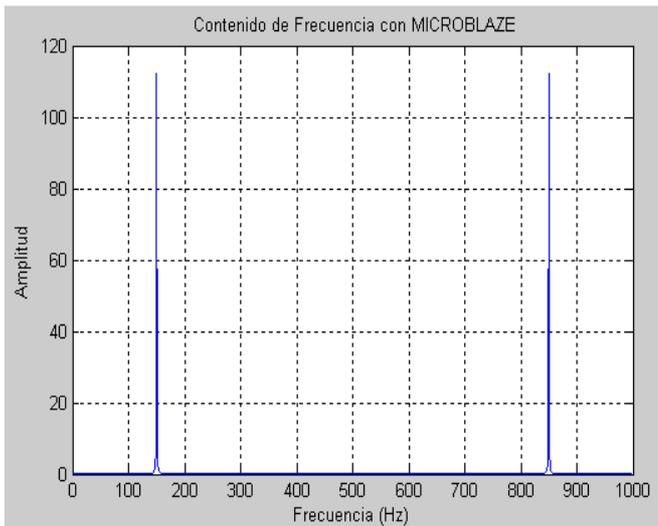


Figura 9 (b)

Figura 9 (a) y (b)

Resultado de la función FFT de Matlab ante la señal generada en Matlab. Resultado del cálculo de la FFT con el programa de MicroBlaze, respectivamente.

Al igual que los casos anteriormente explicados es importante evaluar el consumo de tiempo de la función para calcular la FFT. Para ello se desarrollaron varios experimentos cada uno con diferentes cantidades de muestras de la señal a filtrar, como son: 8 a 1 024. La tabla 3 muestra el tiempo total de ejecución del programa completo en XPS para cada experimento, este tiempo corresponde a todas las funciones del programa, incluyendo las del EDK; y el tiempo consumido por la función nombrada FFT encargada de ejecutar el cálculo de la FFT, el tiempo gastado en esta función es debido a dos bloques fundamentales que son el reordenamiento de los datos, así como la Operación Mariposa, es decir que este tiempo representa el tiempo necesario para obtener el resultado del cálculo de la FFT específicamente, a partir de que estén guardados en la memoria los datos a procesar. El tiempo se va incrementando en la misma medida en que se incrementa el número de muestras a procesar en bloque, este incremento se hace de una manera casi lineal, siendo el mayor el que se consume en el experimento de 1 024 puntos, que es de 5.39425 ms, muy inferior sin embargo a los 128 ms necesarios para formar un bloque de 1024 muestras tomadas a 125us de muestreo.

Tabla 3: Resultados de tiempo de varias pruebas del cálculo de la FFT

	Cantidades de muestras de la señal de cada experimento			
	8	16	32	64
Tiempo Total (us)	745,8	1 438,5	2 746,4	5 278,2
Tiempo Función FFT (us)	28,490	69,91	163,41	378,45

Tabla 3: (continuación)

	Cantidades de muestras de cada Experimento			
	128	256	512	1024
Tiempo Total (us)	10 280,1	28 620,1	85 471,4	161 504,5
Tiempo Función FFT (us)	856,33	1 920,41	4 247,93	5 394,25

Recursos de la tarjeta

Los recursos empleados para implementar el procesador “suave” de 32 bit MicroBlaze, del tipo IP *Core*, de Versión 4.00.a, con *Driver* API, dada la configuración utilizada, se listan en la Tabla 4. Como se aprecia se usa no más de 35 % del total de determinados recursos, lo cual implica que hay reserva para implementar más recursos de lógica híbrida y conformar un sistema SoC.

Tabla 4: Recursos utilizados de la tarjeta

	Slices Uso (%)	Slice Flip Flop Uso (%)	4 input LUTs Uso (%)
Total	34	14	27
Debuggers	1	<1	1
<i>Control. Inter.</i>	1	1	<1
<i>LMB (buses)</i>	0	0	0
<i>mb_opb (buses)</i>	2	0	2
<i>DDR_SDRAM_16Mx16</i>	11	6	6
<i>RS232_DCE</i>	1	0	1
<i>Opb_timer</i>	4	2	2

Tabla 4: (continuación)

	MULT18X18SIOs Uso (%)	GCLKs Uso (%)
Total	35	-
Debuggers	-	8
<i>Control. Inter.</i>	-	-
<i>LMB (buses)</i>	-	-
<i>mb_opb (buses)</i>		
<i>DDR_SDRAM_16Mx16</i>	-	-
<i>RS232_DCE</i>		
<i>Opb_timer</i>		

CONCLUSIONES

A lo largo del desarrollo del trabajo se ha podido constatar que si se emplea el módulo IP MicroBlaze de Xilinx, empotrado en la FPGA Spartan-3E, pueden sintetizarse aplicaciones de Procesamiento Digital de Señales eficientemente. Por lo tanto podemos arribar a las siguientes conclusiones:

- Se produjeron aplicaciones básicas del Procesamiento Digital de Señales como son filtros FIR, filtros IIR y análisis espectral a través de la FFT; en lenguaje de programación de alto nivel C/C++. Se insertaron dichas aplicaciones en un programa en el EDK a través del XPS, capaz de controlar el puerto serie, y el temporizador, ambos con interrupción. Las mismas fueron objeto de verificación para comprobar su correcto funcionamiento antes de ser sintetizadas en la FPGA, a través de la Plataforma Virtual del EDK, del XMD y el GDB; donde además fue depurado su diseño y se pudo obtener los

resultados de tiempo de ejecución tanto del programa general, como de cada función envuelta en las mismas.

- Las aplicaciones se ejecutaron antes de cada interrupción del temporizador, siendo estas de 125 us; procesando las muestras eficientemente, demostrando esto el posible uso del conversor ADC presente en la mencionada tarjeta, en el procesamiento digital de señales en tiempo real.
- Los resultados fueron comprobados en la PC, mediante la comunicación por el puerto serie, a través de las funciones para el procesamiento digital de señales del *software* Matlab; mediante comparaciones entre los resultados de los cálculos de dichas funciones, y los resultados de los cálculos en MicroBlaze que arrojaron una excelente correspondencia entre ambos.
- Se pudo verificar que el consumo de recursos en la tarjeta fue del 35% lo que garantiza poder usar otros módulos IP, ya sean propios del sistema o creados por el usuario.

REFERENCIAS

1. **CABRERA SARMIENTO, ALEJANDRO J.**, *Sistemas empotrados y módulos IP*, Dpto. de Automática y Computación, Instituto Superior Politécnico “José Echeverría” CUJAE.
2. Xilinx, Spartan-3E Starter Kit board User Guide, UG230 (v1.0), March 9, 2006.
3. **ORFANIDIS, SOPHOCLES J.** *Introduction to Signal Processing*.
4. **G. LYONS, RICHARD.** *Understanding Digital Signal Processing*, Eighth Printing, April 2001.
5. **KUO, SEN M.** Northern Illinois University, Dekalb, Illinois, USA; **BOB H. LEE**, Texas Instruments, Inc., Schumburg Illinois, USA; *Real-Time Digital Signal Processing, Implementations, Applications, and Experiments with the TMS320C55X*.
6. **BOURKE, PAUL.** DFT (Discrete Fourier Transform) FFT (Fast Fourier Transform), Junio de 1993, Disponible en: <http://local.wasp.uwa.edu.au/~pbourke/other/dft/>.
7. Xilinx, EDK Concepts, Tools, and Techniques, A Hands-On Guide to Effective Embedded System Design, EDK 9.1i, P/N XTP013 (Version 9.1i).

AUTORES

Raidel Herrera Pereda, graduado de Ingeniería en Telecomunicaciones y Electrónica, Universidad de Pinar del Río “Hermanos Saíz Montes de Oca”, ray@tele.upr.edu.cu, es colaborador del grupo de Investigación para el Diagnóstico Avanzado de Maquinarias (GIDAM).

Juan Raúl Rodríguez Suárez, Licenciado en Física, Doctor en Ciencias Técnicas en 1990, Investigador Auxiliar 1994, Profesor Auxiliar 2005, Universidad de Pinar del Río “Hermanos Saíz Montes de Oca”, jotar@tele.upr.edu.cu, es uno de los fundadores y dirige investigaciones en el Grupo de Investigaciones de Diagnóstico Avanzado de Maquinarias (GIDAM).