

GENERADOR AUTOMÁTICO DE FFT CON ALTO GRADO DE PARALELISMO

Julián F. Acosta Orozco, Mario Vera-Lizcano, Jaime Velasco Medina

Grupo de Bio-Nanoelectrónica, EIEE,
Universidad del Valle A.A. 25360, Cali, Colombia

julifranaco@hotmail.com; mario, jvelasco@univalle.edu.co

ABSTRACT

Este artículo presenta un generador automático del hardware de una arquitectura pipelined paralela base-2 para calcular la Transformada Rápida de Fourier (FFT). Los parámetros del generador son la longitud de la transformada y el número de bits de cada punto, el formato utilizado es de punto fijo para representar datos de tipo complejo. La descripción del generador se realizó con VHDL orientándola para síntesis en FPGAs que incluyan bloques DSP embebidos, particularmente para la familia Stratix II, lo que permite obtener una FFT de high-throughput. A partir del generador se sintetizó automáticamente con herramientas EDA FFTs de 8, 16 y 32 puntos. Obteniéndose un desempeño de frecuencia de operación de 167 MHz y un tiempo de cálculo de 6.0 ns para la FFT de 32 puntos y 18 bits.

1. INTRODUCCION

La FFT aunque es un algoritmo que reduce drásticamente el número de operaciones de la DFT, tiene una carga computacional relativamente alta. La FFT debe ser diseñada detalladamente con el fin de lograr un óptimo desempeño del sistema aplicativo. Las soluciones software y circuitos DSP no son necesariamente las más apropiadas para las aplicaciones que requieren high-throughput. Para los dispositivos FPGAs y ASIC se han desarrollado arquitecturas FFT como soluciones apropiadas para high-throughput [1].

Los fabricantes de FPGAs al tiempo que incluyen en las arquitectura bloques multiplicadores embebidos, también diseñan cores IP que permiten el cálculo de funciones DSP tal como DCT, filtros FIR, Transformada Wavelet, FFT, NCO, codificadores (Reed-Solomon, Viterbi), todos estos con una amplia gama de parámetros.

Este artículo presenta los resultados del diseño de un generador automático genérico que permite obtener una

arquitectura pipelined paralela base-2 para calcular la FFT. Los parámetros del generador son la longitud de la transformada y el número de bits de cada punto, el formato utilizado es de punto fijo para representar datos de tipo complejo. La descripción del generador se realizó con VHDL orientándola para síntesis en FPGAs que incluyan bloques DSP embebidos, particularmente para la familia Stratix II, lo que permite obtener una FFT de high-throughput.

2. ALGORITMO FFT

La FFT es un algoritmo para el cálculo eficiente de la DFT. La expresión (1) es la DFT de la secuencia $x(n)$ para $n = 0, 1, \dots, N-1$, donde N es la longitud de la transformada.

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad k = 0, 1, \dots, N-1 \quad (1)$$

El Algoritmo FFT base-2 realiza el cálculo de la DFT por descomposición de la secuencia de entrada de longitud N , en transformadas de longitud $N/2$, y esta en transformadas de $N/4$, y así sucesivamente hasta obtener transformadas de dos puntos. Estas operaciones básicas son llamadas mariposas o motores de procesamiento.

El factor de giro W_N^{nk} de la ecuación (1) corresponde a la exponencial compleja $e^{-j2\pi k n/N}$. El número de factores de giro y el valor de estos dependen de la longitud de la transformada y de la base utilizada.

Al realizar el diezmado para N durante $\log_2 N$ veces, se reducen las multiplicaciones realizadas por el método normal de la DFT de N^2 , a $(N/2) \cdot \log_2 N$ y las sumas de $N \cdot (N-1)$ a $N \cdot \log_2 N$. [2]

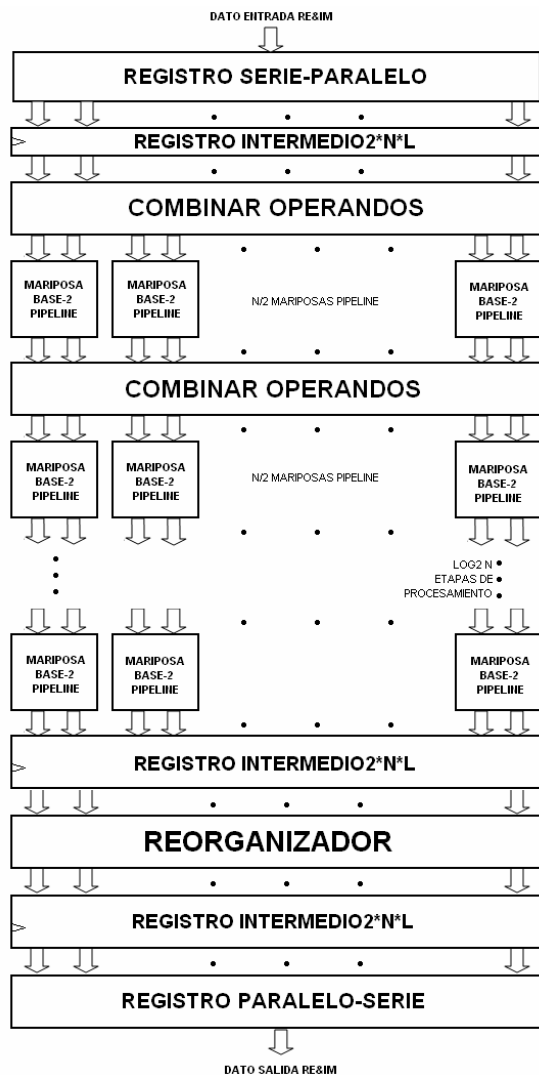


Figura 1: Arquitectura del generador

3. ARQUITECTURA DEL GENERADOR

En la figura 1 se presenta la arquitectura pipelined paralela base-2 del generador automático del hardware para calcular la FFT, donde la longitud de la FFT (N) y el tamaño de la palabra (L) son parametrizables.

Los puntos son ingresados al procesador FFT secuencialmente, pero los cálculos se hacen de forma paralela. Es decir, antes de empezar a procesar los datos de entrada, toda la secuencia debe estar almacenada en un registro paralelo auxiliar llamado REGISTRO INTERMEDIO. La labor de tomar la secuencia de entrada y almacenarla en el Registro Intermedio la ejecuta un bloque llamado REGISTRO SERIE-PARALELO. La operación inversa a la salida del procesador FFT la hace el bloque REGISTRO PARALELO-SERIE. También Existe

un bloque que toma los datos en orden binario invertido y los reordena en su forma natural llamado el REORGANIZADOR.

El procesamiento es ejecutado mediante un arreglo de mariposas base-2 (MARIPOSA BASE-2 PIPELINE). Estos motores de procesamiento reciben los datos de un bloque que combina los operandos adecuadamente (COMBINAR OPERANDOS) antes de cada etapa de procesamiento.

3.1. Registro Serie-paralelo

El procesador FFT genérico de estructura paralela procesa todos los datos de la secuencia de entrada en forma paralela. La longitud N de la transformada que es posible entrar en forma paralela es limitada, por lo que dicha entrada se hace de una forma secuencial.

Los datos de entrada inicialmente pasan por una cadena de registros que los desplaza hasta completar el conjunto de puntos. Posteriormente los datos almacenados en cada registro de la cadena son almacenados en un registro paralelo auxiliar (REGISTRO INTERMEDIO). Es importante indicar que la cantidad de registros es igual a la longitud de la transformada. La figura 2 muestra el diagrama RTL para un registro Serie-Paralelo.

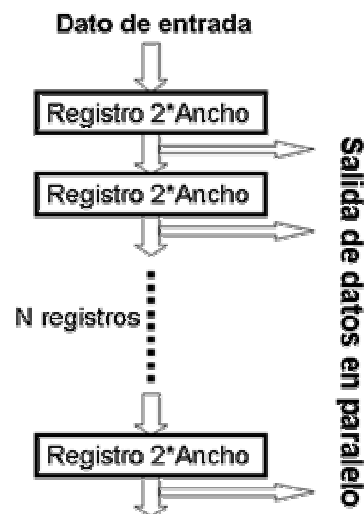


Figura 2: Registro Serie-Paralelo

3.2. Registro Paralelo-Serie

Realiza la función contraria al bloque anterior, es decir, almacena los datos de salida de la transformada en forma paralela y los entrega en forma secuencial. En este bloque la entrada de cada registro debe ser conmutada entre datos de entrada paralela y salida del registro anterior. Es por esta razón que el registro de desplazamiento paralelo-serie tiene una señal de control adicional de carga/desplazamiento llamada DESPCARG, que es la señal de selección de los multiplexores encargados de

conmutar entre los dos estados. El diagrama RTL del registro de desplazamiento se muestra en la figura 3.

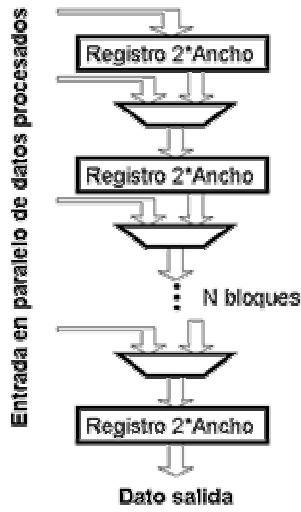


Figura. 3: Registro Paralelo-Serie.

3.3. Mariposa base-2 Pipeline

La mariposa base-2 es la operación que consume la mayor parte del tiempo del procesamiento, debido a que se realiza $(N/2) * \log_2 N$ veces. Al considerarse como el corazón del procesador FFT, esta se diseñó en detalle.

Cada vez que la mariposa opera realiza suma y multiplicación entre tres datos a, b y W, como se observa en la figura 4. Las letras a y b representan datos a procesar en el cálculo de la FFT; W representa los factores de giro del sistema, los cuales son constantes almacenadas como un arreglo de nodos auxiliares. Debido a que W son números complejos, las operaciones en la mariposa deben hacerse para este tipo de datos.

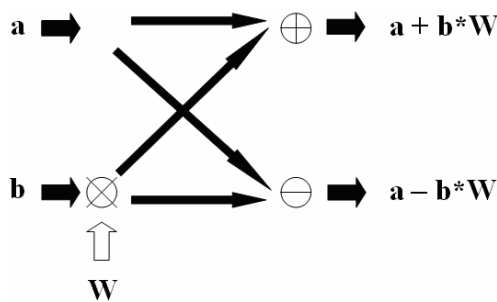


Figura. 4: Mariposa base-2

La multiplicación entre complejos de la mariposa se sintetizó sobre bloques DSP y las suma/resta a la salida, sobre bloques de lógica. Esto permite que haya una mejor distribución en la utilización de recursos del dispositivo, con un poco de sacrificio en el desempeño del procesador,

lo que no sucedería si toda la mariposa se sintetiza sobre bloques DSP. Para la ejecución de la multiplicación compleja entre $b = A + jB$ y $W = C + jD$ se requieren de cuatro multiplicaciones y dos sumas como lo indica la expresión 2 [3][4]. La figura 5 muestra como fue configurado cada bloque DSP.

$$b * W = (A * C - B * D) + j(A * D + B * C) \quad (2)$$

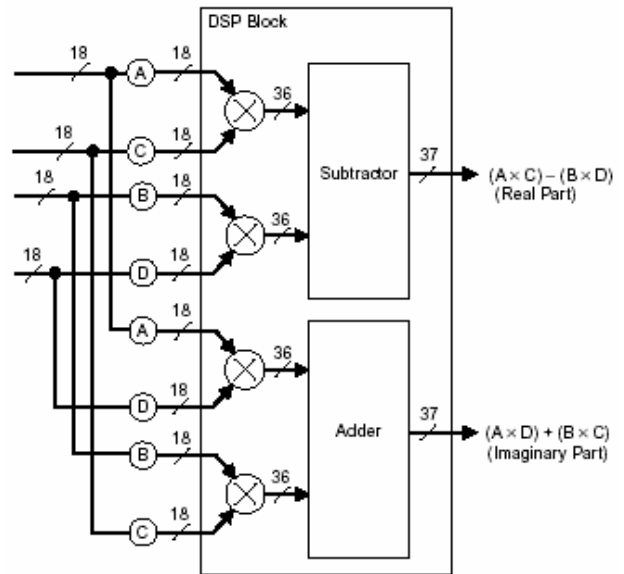


Figura 5: Multiplicación compleja en un bloque DSP

3.4. Bloque para combinación de datos

El cálculo de la FFT necesita de $\log_2 N$ etapas de procesamiento, cada una posee un bloque a la entrada que combina los datos de acuerdo a la etapa de procesamiento en la que se encuentra. Una de las principales ventajas del algoritmo FFT base-2 es la regularidad en la forma en la que se presentan las operaciones. Lo cual permite su descripción en VHDL mediante una anidación de estructuras repetitivas.

En la figura 1 se presenta el flujo de información necesario para el cálculo de una FFT, se observa que los cálculos se hacen en paralelo con un flujo en pipelined. Se debe considerar que la secuencia de entrada puede ser almacenada en orden normal (caso considerado) o en orden binario invertido. En ambos casos los recursos de hardware son similares.

En la tabla I se indica la secuencia de las operaciones para una FFT de longitud 8, esta secuencia depende de la posición de cada dato en el dispositivo de almacenamiento (memoria o registros) y del factor de giro correspondiente [5].

ETAPA 0	ETAPA 1	ETAPA 2
---------	---------	---------

$D_i \pm D_j * W$	$D_i \pm D_j * W$	$D_i \pm D_j * W$
D0±D4*W0	D0±D2*W0	D0±D1*W0
D1±D5*W0	D1±D3*W0	D2±D3*W2
D2±D6*W0	D4±D6*W2	D4±D5*W1
D3±D7*W0	D5±D7*W2	D6±D7*W3

TABLA I: Secuencia de operaciones para la FFT con N = 8

En las tablas I y II se ha resaltado en escala de grises las diferentes operaciones para destacar que hay un cambio de secuencia. Se observa que el factor de giro W_N^{nk} está representado abreviadamente como por W_{nk} . Para una transformada de 32 puntos se tiene la secuencia de la tabla II.

ETAPA 0 $D_i \pm D_j * W$	ETAPA 1 $D_i \pm D_j * W$	ETAPA 2 $D_i \pm D_j * W$	ETAPA 3 $D_i \pm D_j * W$	ETAPA 4 $D_i \pm D_j * W$
D0±D16*W0	D0±D8*W0	D0±D4*W0	D0±D2*W0	D0±D1*W0
D1±D17*W0	D1±D9*W0	D1±D5*W0	D1±D3*W0	D2±D3*W8
D2±D18*W0	D2±D10*W0	D2±D6*W0	D4±D6*W8	D4±D5*W4
D3±D19*W0	D3±D11*W0	D3±D7*W0	D5±D7*W8	D6±D7*W12
D4±D20*W0	D4±D12*W0	D8±D12*W8	D8±D10*W4	D8±D9*W2
D5±D21*W0	D5±D13*W0	D9±D13*W8	D9±D11*W4	D10±D11*W10
D6±D22*W0	D6±D14*W0	D10±D14*W8	D12±D14*W12	D12±D13*W6
D7±D23*W0	D7±D15*W0	D11±D15*W8	D13±D15*W12	D14±D15*W14
D8±D24*W0	D16±D24*W8	D16±D20*W4	D16±D18*W2	D16±D17*W1
D9±D25*W0	D17±D25*W8	D17±D21*W4	D17±D19*W2	D18±D19*W9
D10±D26*W0	D18±D26*W8	D18±D22*W4	D20±D22*W10	D20±D21*W5
D11±D27*W0	D19±D27*W8	D19±D23*W4	D21±D23*W10	D22±D23*W13
D12±D28*W0	D20±D28*W8	D24±D28*W12	D24±D26*W6	D24±D25*W3
D13±D29*W0	D21±D29*W8	D25±D29*W12	D25±D27*W6	D26±D27*W11
D14±D30*W0	D22±D30*W8	D26±D30*W12	D28±D30*W14	D28±D29*W7
D15±D31*W0	D23±D31*W8	D27±D31*W12	D29±D31*W14	D30±D31*W15

TABLA II: Secuencia de operaciones para la FFT con N = 32

En la tabla II se observa que el número de etapas es $\log_2 N$, cada etapa tiene un número de saltos (diferenciados en escala de grises), esta cantidad de saltos está gobernada por la expresión 2^{ETAPA} . Cada salto se contabiliza dependiendo tanto del salto en el que se encuentre como de la etapa. El factor de giro es el número del salto representado en orden binario invertido. Cada celda de la tabla II se calculó mediante un algoritmo desarrollado en esta investigación.

4. ANÁLISIS DE RESULTADOS

La arquitectura FFT genérica se sintetizó sobre el dispositivo EP2S180F1020C4 de la familia STRATIX II.

4.1. Resultados de Compilación

Los resultados de la compilación en QUARTUS II de la FFT para N = 8, 16, 32 con un formato de datos (L) de 18 bits, se presentan en la tabla III. Cuando el generador de FFT es parametrizado con N igual a 8 puntos, se obtiene el hardware de la FFT operando a una frecuencia de 197 MHz; Cuando se parametriza con N igual a 32, se reduce la frecuencia de operación a 167 MHz y se utiliza el 83% del total de bloques DSP (multiplicador embebido de 9 bits) de la FPGA.

Longitud de la FFT (N)	8	16	32
Fmax. (MHz)	197	184	167

Bloque DSP	96	256	640
ALUTs	2915	6850	15738
Tiempo de Cálculo (ns)	5.07	5.43	5.98

TABLA III: Compilación FFT Genérica con N = 8,16,32

Para N = 64 y L = 18, no se dispone en la FPGA de los recursos DSP necesarios. Por lo tanto para la arquitectura propuesta y con L = 18, una FFT de 32 puntos es la longitud máxima que se puede generar sobre las FPGA de esta familia. Para poder obtener una FFT con N = 64, es necesario reducir L a un valor menor de 9.

Para la arquitectura desarrollada el tiempo de cálculo de la FFT, está relacionado directamente con el inverso de la Fmax ($T_{calc} = 1/F_{max}$). Se observa en la figura 6 que el tiempo de cálculo aumenta directamente proporcional con N.

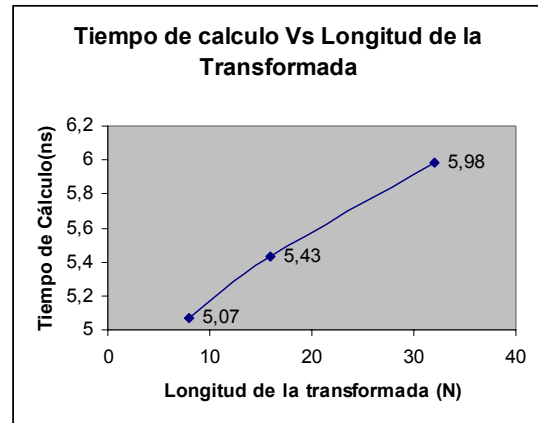


Figura 6: Tiempo de cálculo de la FFT en función de N

Como se indicó en el numeral 3, el procesador FFT posee una estructura pipeline, donde el número de etapas es $\log_2 N$. En la ruta pipeline interviene en cada una de las $\log_2 N$ etapas una mariposa, y cada operación de mariposa se implementó con 3 etapas de pipeline para obtener un mejor throughput. Por lo tanto la latencia de la FFT está determinada por la expresión (3).

$$Latencia = 3 * \log_2 N \quad (3)$$

Otro de los aspectos de análisis es el número de bloques DSP (multiplicador de 9 bits), el cual depende de la longitud de la transformada (N) y del tamaño de los operandos (L). El número de bloques DSP para N puntos con L = 18, corresponde a la expresión 4.

$$Bloques DSP = 4 * N * \log_2 N \quad (4)$$

El porcentaje de LUTs que se requieren con respecto al total que posee el dispositivo es bajo, por lo cual no se considera relevante.

4.2. Simulación

Para verificar el funcionamiento de la FFT sintetizada, se realizó la simulación con la herramienta QUARTUS II de FFTs con $N = 8$ y $L = 9$ y 18 .

El elemento patrón utilizado son los resultados del cálculo de la FFT con MATLAB para el vector de prueba $X = [-1, 1, 2, -0.5, -3, -1, 2, 0]$ cuyo resultados es $Y = [-0.5000, 3.7678-1.0607i, -8.0000-0.5000i, 0.2322-0.0607i, 0.5000, 0.2322+1.0607i, -8.0000+0.5000i, 3.7678+1.0607i]$

Los resultados de la simulación del procesador para datos de entrada de 9 bits, con representación en punto fijo y complemento a dos se muestran en la figura 7. Se observa que los 4 bits menos significativos de cada dato a la salida del procesador FFT representan la parte decimal, los siguientes cuatro la parte entera y el bit más significativo el signo.

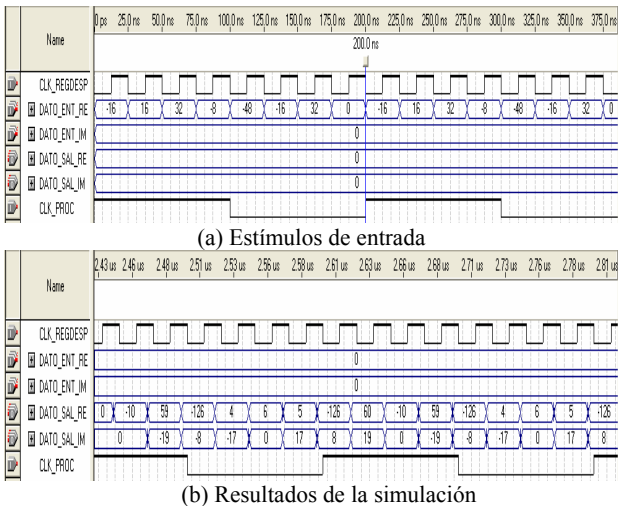


Figura 7: Simulación para datos de 9 bits

El error promedio tomando como base los valores arrojados por MATLAB es de 12.14%, este valor disminuye a medida que se toma mayor cantidad de bits para representar la parte decimal, al coste de tener un rango más reducido para representar datos.

La simulación para datos de 18 bits se presenta en la figura 8. El error promedio para datos representados en 18 bits es del 0.3%. En esta representación los datos toman los 9 bits menos significativos para representar la parte decimal, los siguientes ocho para la parte entera y el bit más significativo el signo.

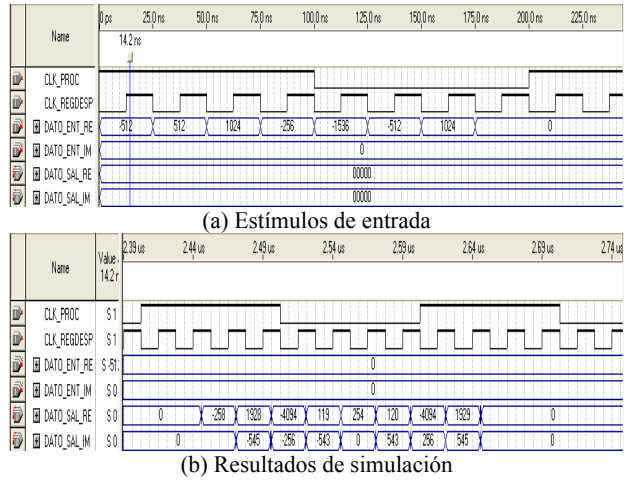


Figura 8: Simulación para datos de 18 bits

La mejora en la precisión es considerable con respecto a la representación de datos en 9 bits. Es importante anotar que después de cada operación aritmética se hace truncamiento independiente de L

4.3. Comparación FFT genérica

En la Tabla IV se compara los resultados de síntesis del generador genérico desarrollado (FFT_UVII), para FFT con $N = 8, 16, 32$ y un formato de datos (L) de 18 bits. Estos resultados son comparados contra algunos diseños comerciales. Los valores de la tabla corresponden al tiempo necesario para realizar una transformada y se presentan en nano-segundos

Nombre \ Longitud	8	16	32
FFT_UVII	5.0	5.4	6.0
DILLON [6]	8.0	8.0	8.0
XILINX[7]	16	32	64
ALTERA [8]	648	1037	1729

TABLA IV: Comparación del generador de FFT

Como se puede observar en la tabla, el desempeño de la FFT genérica está por encima de las otras arquitecturas. Es de notar que todas las arquitecturas comparadas trabajan con formato de punto fijo, excepto ALTERA que utiliza representación IEEE 754 de precisión sencilla. Dillon posee una arquitectura en la cual el número de mariposas es parametrizable, esto la hace que se pueda lograr con esta un muy buen desempeño cuando se ajusta este valor al mismo de la FFT_UVII. XILINX presenta una arquitectura con datos representados en punto fijo de 16 bits, los cuales pueden ser escalados después de las operaciones de mariposa; esto hace que tenga una mejor precisión y un mayor rango dinámico.

5. CONCLUSIONES

Se diseñó un generador automático del hardware de una arquitectura pipelined paralela base-2 para calcular la Transformada Rápida de Fourier (FFT) con high-throughput. Los parámetros del generador son la longitud de la transformada (N) y el número de bits de cada punto (L), los cuales presentan un formato de punto fijo para representar datos de tipo complejo.

A pesar de haber desarrollado el código VHDL para $N = 2^m$ puntos, donde $2 < m < 11$, solo se logró realizar pruebas de simulación para $m = 3, 4$ y 5 , debido a que para valores superiores de m no existen FPGAs con los recursos DSP necesarios para esta arquitectura.

El diseño se orientó para dispositivos que poseen bloques dedicados para funciones DSP, logrando frecuencias de operación que oscilan entre 197 y 167 MHz, con tiempos de cálculo entre 5 y 6 ns.

Se corroboró que las FPGAs proporcionan una alternativa para el diseño de FFT, las FPGAs hace posible la utilización de paralelismo en la ejecución de las diferentes operaciones aritméticas, y de esta forma mejorar la velocidad de procesamiento para las diferentes aplicaciones. Además la posibilidad de diseñar con lenguajes HDL brinda la oportunidad de realizar arquitecturas parametrizables.

6. AGRADECIMIENTO

Algunas de las herramientas utilizadas en este trabajo han sido suministradas por Altera Corporation a través del Programa Universitario.

7. REFERENCES

- [1] L. Fanucci, M. Forliti, P. Terreni, FAST: FFT ASIC automated synthesis. INTEGRATION, The VLSI Journal 33, 2002, pag 23-37
- [2] J. Proakis, D. Manolakis, Tratamiento digital de señales. 3a edición, Prentice Hall, Madrid, 1998, pp 465-473.
- [3] Chapter 6. DSP Blocks in Stratix II Devices, Stratix II Device Handbook, Volume 2, Altera Corporation, July 2004. Disponible: www.altera.com
- [4] Stratix II architecture, Stratix II Device Handbook, Volume 1, Altera Corporation, July 2005. Disponible: www.altera.com
- [5] Chu E., George A., Inside the FFT black box Serial and Parallel Fast Fourier Transform Algorithms, CRC, Ontario, Canada, 2000.
- [6] Ultra High-Performance FFT/IFFT IP Core, Dillon Engineering, Inc, 2003. Disponible: www.dilloneng.com
- [7] Fast Fourier Transform, Volume 3.1, XILINX Corp., april 2005. Disponible: www.xilinx.com/ipcenter/catalog/logicore/docs/xfft.pdf
- [8] Floating Floating-Point FFT Processor (IEEE 754 Single Precision) Radix 2 Core, Altera Corporation, 101 Innovation Drive, San Jose, California 95134, USA, May 2003.

