

PROCESADOR PARALELO DE FFT IMPLEMENTADO EN FPGA

Pablo Sonna, Elvio Serrano, Leonardo Castillo.

Grupo Tratamiento de la Información, Instituto Universitario Aeronáutico, Córdoba, Argentina

psonna876@alumnos.iaa.edu.ar, eserrano301@alumnos.iaa.edu.ar,
lcastillo591@alumnos.iaa.edu.ar

ABSTRACT

La transformada rápida de Fourier es una aplicación que consume muchos recursos computacionales para su cálculo, razón por la cual es interesante disponer de hardware dedicado a esta tarea exclusivamente.

En este trabajo se exploran las capacidades de una FPGA para implementar un procesador de FFT con entrada de datos en paralelo. El procesamiento en paralelo tiene la ventaja de ser extremadamente rápido, pero tiene una limitación, que es la cantidad de celdas necesarias para lograr un procesador de alta resolución.

1. INTRODUCCIÓN

Existe una gran variedad de algoritmos aplicables al cálculo de la FFT, sin embargo, el más popular de todos es el algoritmo de Cooley-Tukey, en los N puntos de la transformada se pueden expresar como una potencia de la base r , $N=r^p$. Estos algoritmos se conocen como algoritmos radix- r y los más populares son aquellos en que $r=2$ o $r=4$.

El algoritmo de radix-2 diezmado en frecuencia se obtiene diezmado la serie de salida ϕ veces, en frecuencias pares $\{X_{2k}|k=0, \dots, N/2-1\}$ e impares $\{X_{2k+1}|k=0, \dots, N/2-1\}$ [1]. Para un caso de una transformada de 8 puntos, los cálculos que se deben efectuar son los que se muestran en la figura 1.

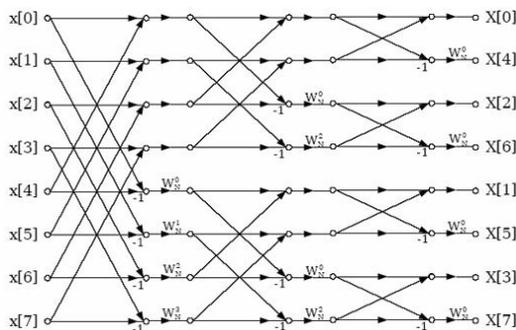


Figura 1 – Algoritmo Radix-2 de 8 Puntos

Los procesadores de FFT comúnmente utilizados, calculan estas sumas y multiplicaciones complejas de manera secuencial. Sin embargo, el procesador que se

implementó en este trabajo realiza los ocho cálculos de manera simultánea, de manera que en tres operaciones se realiza un cálculo completo de una FFT de ocho puntos.

2. ARQUITECTURA DEL PROCESADOR

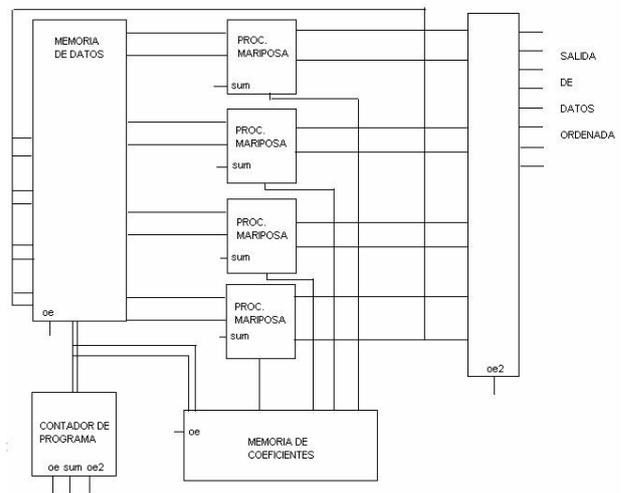


Figura 2 – Procesador de FFT en Paralelo

El procesador de FFT que se implementó en VHDL, dispone de cinco bloques fundamentales, que son: una memoria de 16 salidas en paralelo de ocho bits cada una, que además cumple la función de ordenar los datos para el cálculo, una memoria de tres salidas en paralelo de los coeficientes de multiplicación complejos, un contador de programa, cuatro procesadores de mariposa y un bloque de salida que ordena los datos para ser leídos, ya que el algoritmo los reordena según el orden binario inverso [1].

El contador de programa es el encargado de sincronizar todos estos bloques. El mismo posee una salida de dos bits (sólo se necesita contar hasta tres para una FFT de ocho puntos) para indicar en que estado se encuentra el proceso, con una salida habilita la salida de datos hacia los procesadores de mariposa y con otra salida realiza la operación de la mariposa. Cuando el cálculo ha sido completado, habilita la salida de los 16 datos del resultado, para que puedan ser leídos en paralelo.

El procesador mariposa debe efectuar la operación que se muestra en la figura 3, para lo cual se necesita un multiplicador complejo, un sumador complejo y un restador complejo. El procesador implementado, sigue el mismo concepto que el que se describe en [2] para reducir la cantidad de multiplicadores utilizados, ya que el multiplicador es el bloque que más celdas consume en una FPGA.

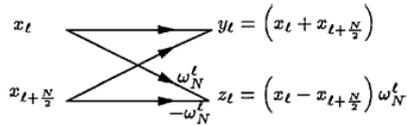


Figura 3 – Procesador Mariposa

3. IMPLEMENTACIÓN Y ENSAYOS

El procesador descrito en el inciso anterior, se programó en VHDL, en el programa de simulación e implementación ISE 8.2 de Xilinx. Para corroborar su funcionamiento, en primera instancia se realizaron simulaciones en software, obteniendo así los resultados esperados. Luego, se lo programó en una FPGA Spartan-3 de Xilinx, la cual se encuentra en la placa de prueba Starter Kit Spartan 3. En la tabla 1 se muestra un resumen de la utilización del dispositivo.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note[s]
Number of Slice Flip Flops	552	3,840	14%	
Number of 4 input LUTs	834	3,840	21%	
Logic Distribution				
Number of occupied Slices	565	1,920	29%	
Number of Slices containing only related logic	565	565	100%	
Number of Slices containing unrelated logic	0	565	0%	
Total Number 4 input LUTs	865	3,840	22%	
Number used as logic	834			
Number used as a route-thru	31			
Number of bonded IOBs	12	173	6%	
IOB Flip Flops	1			
Number of MULT18x18s	12	12	100%	
Number of GCLKs	4	8	50%	
Total equivalent gate count for design	59,447			
Additional JTAG gate count for IOBs	576			

Tabla 1 – Resumen de Utilización de la FPGA Spartan 3

El tiempo de propagación máximo que entrega el programa de síntesis es de 8.526 [ns] para seis niveles lógicos. Esto permitiría operar el dispositivo a 117.29 MHz. Sin embargo, la placa de prueba está limitada a un clock de 50 MHz, por lo que esa es la máxima frecuencia a la que se probó el procesador.

La entrada de datos a la memoria se realizó a través de un puerto de expansión de la placa, directamente hacia la FPGA. Para poder visualizar los resultados obtenidos, se programó además un bloque de transmisión de datos vía RS-232, para tomarlos con una PC y graficar los resultados. Para la visualización de los datos se utilizó el programa de cálculo Matlab.

En la figura 3 se muestra un diagrama de bloques del sistema completo. Para introducir los datos a la

FPGA, se utilizó un procesador PIC, que sincronizado con la FPGA transmite los ocho puntos para efectuarles la FFT.

La rutina de Matlab que se utiliza para mostrar los resultados, toma los 16 bytes provenientes del procesador de FFT a través del puerto serial de la computadora, a 9600 baudios, y los muestra en un gráfico en pantalla. En la figura 4 se muestra un gráfico, resultante de aplicarle la FFT a una senoide pura de frecuencia digital (F/Fs) igual a 0.1, que previamente fue grabada en el procesador PIC.

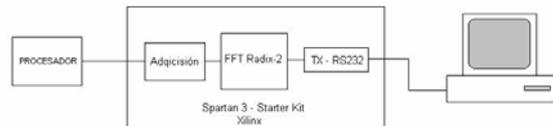


Figura 3 – Sistema Completo

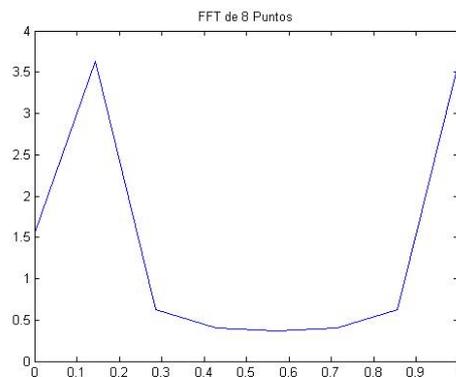


Figura 4 – Resultado de la FFT de una Sinusoide

4. REFERENCIAS

- [1] E. Chu and A. George, *Inside The FFT Blackbox*, CRC Press, Florida, Enero de 2000.
- [2] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Florida.

5. AGRADECIMIENTOS

Agradecemos al Grupo de tratamiento de la información y sistemas irradiantes del Instituto Unversitario Aeronáutico de Córdoba, Argentina. En especial al Ingeniero Ezequiel Brac y el Ingeniero Pablo Ferreyra, que en el contexto del proyecto “Control del Haz de un Arreglo de Antenas” del programa de apoyo a grupo de reciente formación de la Agencia Córdoba Ciencia, nos han brindado todo su apoyo y conocimientos para la elaboración de este trabajo.