



Estimación de Armónicos sobre FPGA Aplicando Estadística de Orden Superior y Convolución

*Miguel Enrique Iglesias Martínez, Bárbaro M. López Portilla, Fidel Ernesto Hernández
Montero*

RESUMEN / ABSTRACT

En este trabajo se propone la implementación sobre FPGA de un modelo conjunto que aplica las características estadísticas (cumulantes) de orden superior, combinadas a un proceso de convolución, lo cual permite obtener las componentes espectrales con su amplitud, frecuencia y fase originales, contaminadas por un proceso de ruido de distribución normal, aún desconociendo la señal de entrada. El aporte fundamental del mismo radica en la implementación del modelo propuesto sobre una arquitectura completamente paralela para su uso en aplicaciones en tiempo real en tareas de reducción de ruido y detección de señales. Los resultados obtenidos, altamente satisfactorios, demuestran la efectividad de la utilización de características estadísticas de orden superior y su combinación con procesamiento de segundo orden (convolución) para la cancelación de ruido y estimación de parámetros, entre otras tareas.

Palabras claves: Cumulantes, FPGA, Estimación, Cancelación de Ruido

This work is about the FPGA implementation of a joint model that applies higher-order statistical characteristics (cumulants), combined with a convolution process, which allows to obtain the spectral components with their amplitude, frequency and phase originals, contaminated by a noise process of normal distribution, even not knowing the input signal. The main contribution of this work is the implementation of the proposed model on a fully parallel architecture, for use in real-time applications in noise cancellation and signal detection. The results obtained highly satisfactory, demonstrate the effectiveness of the use of higher order statistical characteristics and its combination with the second order processing (convolution), for noise reduction and parameters estimations, among other tasks.

Key words: *Cumulants, FPGA, Estimation, Noise Cancellation*

HARMONICS ESTIMATION ON FPGA APPLYING HIGHER-ORDER STATISTICS AND CONVOLUTION

INTRODUCCION

Las fluctuaciones de un proceso de señal completamente aleatorio o la distribución de una clase de señales aleatorias en el espacio, no pueden ser modeladas por una ecuación predictiva, pero pueden describirse en términos de los estadísticos de la señal y modelados por una función de distribución de probabilidad en un espacio de señal multidimensional [1]. Numerosos métodos han sido desarrollados buscando precisamente la extracción de señales perturbadoras, entre los cuales destacan los que se basan en el procesamiento estadístico de la señal.

En la actualidad se han reportado métodos de pro-cesamiento de señal, basados en espectros de orden superior, los cuales aunque conservan la información de fase [2] [3], su complejidad computacional no los hace viables para soluciones en la práctica ya que las características espectrales de orden superior son funciones que se manejan en planos multidimensionales. Además de estos, existen otros métodos como los llamados PEP (Phase Estimation using Polyspectrum Slice) [4], utilizados para la reconstrucción de la fase en sistemas LTI (Linear Time Invariant), lo cual ve limitados estos modelos para señales en banda base, afectadas por ruido, cuya información de amplitud, frecuencia y fase se desconoce.

Hoy en día los sistemas electrónicos modernos demandan cada vez mayor capacidad de cómputo. Actualmente los FPGAs ofrecen gran cantidad de recursos al diseñador, millones de puertas lógicas equivalentes, bloques de memoria, bloques DSP, e incluso las más modernas incluyen uno o varios procesadores dentro de la propia FPGA con arquitectura completamente paralela, lo cual puede ser de gran utilidad en tareas de reducción o cancelación ruido, identificación de sistemas, y detección de señales, a la hora de calcular la función o funciones correspondientes del proceso de señal dado, las cuales pueden consumir excesivamente demasiado tiempo en su ejecución.

El presente trabajo muestra la implementación sobre FPGA de un modelo conjunto de estimación de parámetros basado en estadística de orden superior y un proceso de convolución, como metodología para la aceleración del cómputo de funciones estadísticas de orden superior y su posterior aplicación para sistemas de procesamiento de señales en tiempo real.

El trabajo se ha organizado de la siguiente manera: como parte del desarrollo del trabajo cada una de las secciones, explicarán los aspectos generales sobre la estimación basada en estadísticos de orden superior de la señal, además de las soluciones propuestas y los resultados obtenidos de la investigación, conjuntamente con las plataformas de trabajo utilizadas, para finalmente exponer las conclusiones de la investigación y las referencias consultadas.

ASPECTOS TEÓRICOS GENERALES SOBRE ESTADÍSTICOS DE ORDEN SUPERIOR

Existen varias motivaciones generales detrás del uso de la estadística de orden superior en el procesamiento de señales, dentro de las cuales destaca [5]:

- Eliminar el ruido coloreado aditivo gaussiano de espectro de potencia desconocida.
- Identificar los sistemas de fase no mínima o reconstrucción de señales de fase no mínima.
- Extraer la información debido a las desviaciones de Gaussianidad.
- Detectar y caracterizar las propiedades no lineales de las señales, así como identificar los sistemas no lineales.

Sea entonces $\{X(k)\}, k = 0, \pm 1, \pm 2, \pm 3, \dots$ una señal real estacionaria discreta en el tiempo donde sus momentos de orden superior existen, luego

$$m_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) = E\{X(k)X(k + \tau_1) \dots X(k + \tau_{n-1})\} \quad (1)$$

representa el momento de orden n de una señal estacionaria, el cual depende solo de los diferentes espacios de tiempo $\tau_1, \tau_2, \dots, \tau_{n-1}, \tau_i = 0, \pm 1, \dots$ para todo i . Claramente se puede apreciar que el momento de segundo orden $m_2^x(\tau_1)$ es la función de autocorrelación de $\{X(k)\}$, del mismo modo que $m_3^x(\tau_1, \tau_2)$ y $m_4^x(\tau_1, \tau_2, \tau_3)$ representan los momentos de tercero y cuarto orden, respectivamente. $E\{\cdot\}$ denota el operador valor esperado de la señal [5]. Luego a partir de esto se puede decir que el cumulante de orden n de una señal aleatoria no gaussiana estacionaria, $\{X(k)\}$, puede escribirse solamente para $n = 3$ o $n = 4$ como:

$$C_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) = m_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) - m_n^G(\tau_1, \tau_2, \dots, \tau_{n-1}) \quad (2)$$

Donde $m_n^G(\tau_1, \tau_2, \dots, \tau_{n-1})$ es el momento de orden n de una señal gaussiana equivalente que presente el mismo valor medio y función de autocorrelación que $\{X(k)\}$. Si $\{X(k)\}$ es Gaussiana, $m_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) = m_n^G(\tau_1, \tau_2, \dots, \tau_{n-1})$, por tanto

$C_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) = 0$. Aunque la ecuación (1) es solo verdadera para $n = 3$ y $n = 4$, $C_n^x(\tau_1, \tau_2, \dots, \tau_{n-1}) = 0$ para todo n si $\{X(k)\}$ es gaussiano.

- Cumulante de Cuarto Orden

$$\begin{aligned} c_4^x(\tau_1, \tau_2, \tau_3) = & m_4^x(\tau_1, \tau_2, \tau_3) \\ & - m_2^x(\tau_1) \cdot m_2^x(\tau_3 - \tau_2) - m_2^x(\tau_2) \cdot m_2^x(\tau_3 - \tau_1) \\ & - m_2^x(\tau_3) \cdot m_2^x(\tau_2 - \tau_1) - m_1^x[m_3^x(\tau_2 - \tau_1, \tau_3 - \tau_1) \\ & + m_3^x(\tau_2, \tau_3) + m_3^x(\tau_2, \tau_4) + m_3^x(\tau_1, \tau_2)] \\ & + (m_2^x)^2[m_1^x(\tau_1) + m_2^x(\tau_2) + m_2^x(\tau_3) + m_2^x(\tau_3 - \tau_1) \\ & + m_2^x(\tau_3 - \tau_2) + m_2^x(\tau_2 - \tau_1)] - 6(m_1^x)^4 \end{aligned} \quad (3)$$

Si la señal $\{X(k)\}$ tiene media cero ($m_1^x = 0$), los cumulantes de segundo y tercer orden son idénticamente iguales a los momentos de segundo y tercer orden, respectivamente. Sin embargo, para generar el cumulante de cuarto orden, se necesita contar con los momentos de segundo y cuarto orden en la ecuación (3).

$$\begin{aligned} c_4^x(\tau_1, \tau_2, \tau_3) = & m_4^x(\tau_1, \tau_2, \tau_3) \\ & - m_2^x(\tau_1) \cdot m_2^x(\tau_3 - \tau_2) - m_2^x(\tau_2) \cdot m_2^x(\tau_3 - \tau_1) \\ & - m_2^x(\tau_3) \cdot m_2^x(\tau_2 - \tau_1) \end{aligned} \quad (4)$$

ESTIMACIÓN DE SEÑALES BASADA EN LA APLICACIÓN DE ESTADÍSTICA DE ORDEN SUPERIOR

Para el caso real en un problema de reducción del ruido para una señal dada (componente armónica) los datos observados pueden ser descritos de la forma:

$$y(t) = A \cos(2\pi f t + \phi) + w(t) = x(t) + w(t) \quad (5)$$

Donde $x(t)$ es la señal útil, para este caso, $A \cos(2\pi f t + \phi)$ y $w(t)$ es el ruido aditivo. Además, A , f y ϕ representan la amplitud, frecuencia y fase de la señal correspondiente.

De lo tratado en [5], el cumulante de segundo orden de un proceso de ruido es distinto de cero. Por esta razón, para cancelar ruido, se pudiera estimar el cumulante de tercer orden para el proceso descrito en (5), sin embargo, existe la problemática de que para una señal cosinusoidal su cumulante de orden 3 es cero [6]. Además se puede comprobar que para funciones cuya densidad probabilística sea simétrica el resultado del cumulante de tercer orden es siempre cero [7]. Por esta razón, en esta investigación se trabajará con la estimación del cumulante de cuarto orden.

ESTIMACIÓN DEL CUMULANTE DE CUARTO ORDEN

El cumulante de cuarto orden del proceso descrito en (5) se puede calcular de la siguiente manera:

$$\begin{aligned} C_{4y}(\tau_1, \tau_2, \tau_3) &= C_{4x}(\tau_1, \tau_2, \tau_3) + C_{4w}(\tau_1, \tau_2, \tau_3) \\ &= C_{4x}(\tau_1, \tau_2, \tau_3) \end{aligned} \quad (6)$$

$$\begin{aligned} C_{4x}(\tau_1, \tau_2, \tau_3) &= E\{x(t) \cdot x(t + \tau_1) \cdot x(t + \tau_2) \cdot x(t + \tau_3)\} \\ &\quad - C_{2x}(\tau_1) \cdot C_{2x}(\tau_2 - \tau_3) \\ &\quad - C_{2x}(\tau_2) \cdot C_{2x}(\tau_3 - \tau_1) \\ &\quad - C_{2x}(\tau_3) \cdot C_{2x}(\tau_1 - \tau_2) \end{aligned} \quad (7)$$

Si se trabaja solamente con la componente unidimensional del cumulante de cuarto orden de la señal, $C_{4x}(\tau_1, 0, 0)$, se puede notar que se llega al mismo resultado expuesto en [5] haciendo coincidir $\tau_2 = \tau_3 = 0$, lo cual con tan solo graficar esta componente se pueden recuperar los parámetros fundamentales (amplitud y frecuencia) de la muestra útil de señal, aunque no se recupera la fase. Esto puede ser aprovechado a la hora de aplicar este método en plataformas hardware tales como FPGAs, microcontroladores, DSP, entre otros. Luego, si se hace $\tau_2 = \tau_3 = 0$, se puede obtener lo siguiente:

$$\begin{aligned} C_{4x}(\tau_1, 0, 0) &= E\{x(t)^3 \cdot x(t + \tau_1)\} - 3 \cdot C_{2x}(\tau_1) \cdot C_{2x}(0) \\ &= -\frac{3A^4}{8} \cos(w_k \tau_1) \end{aligned} \quad (8)$$

El término $C_{2x}(\tau_1)$ para un proceso de valor medio igual a cero corresponde a la función de autocorrelación de la señal lo cual para el caso de armónicos reales es $\frac{A^2}{2} \cos(w_k \tau_1)$ y el término $C_{2x}(0)$ corresponde a la varianza de la señal que para una muestra cosinusoidal corresponde a, $\frac{A^2}{2}$, lo cual representa la potencia de la señal. Luego, desarrollando la ecuación (8) y tomando como señal $x(t)$ declarada en (5) se tiene:

$$\begin{aligned}
 E\{x(t)^3 \cdot x(t + \tau_1)\} &= \int_{-\pi}^{\pi} g(x) \cdot f_{\phi}(\phi) d\phi \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} A \cos(w_k t + \phi)^3 \cdot A \cos(w_k t + w_k \tau_1 + \phi) d\phi \\
 &= \frac{3A^4}{8} \cos(w_k \tau_1)
 \end{aligned} \tag{9}$$

Sustituyendo en la ecuación (10) el resultado obtenido en la expresión (9) se obtiene:

$$C_{4_x}(\tau_1, 0, 0) = \frac{3A^4}{8} \cos(w_k \tau_1) - \frac{3A^4}{4} \cos(w_k \tau_1) \tag{10}$$

Lo que da como resultado, de manera similar a lo obtenido en [5]:

$$C_{4_x}(\tau_1, 0, 0) = -\frac{3A^4}{8} \cos(w_k \tau_1) \tag{11}$$

Como $w(t)$ se considera una señal aleatoria con distribución normal, $C_{4_w}(\tau_1, \tau_2, \tau_3) = 0$.

Considerando el resultado obtenido en la ecuación (11), se puede concluir que se obtienen la amplitud y frecuencia de la componente espectral original aunque se presenta la problemática de la pérdida de la fase de la misma, lo cual será tratado a continuación.

MÉTODO DE RECUPERACIÓN DE FASE

De lo tratado en la sección anterior se observa la pérdida de la fase original de la componente espectral dando como resultado la ecuación (11). En este trabajo se propone, como solución para recuperar la fase, realizar la convolución de la señal contaminada inicial con la salida obtenida tras evaluar la componente unidimensional del cumulante de cuarto orden (ver figura 1).

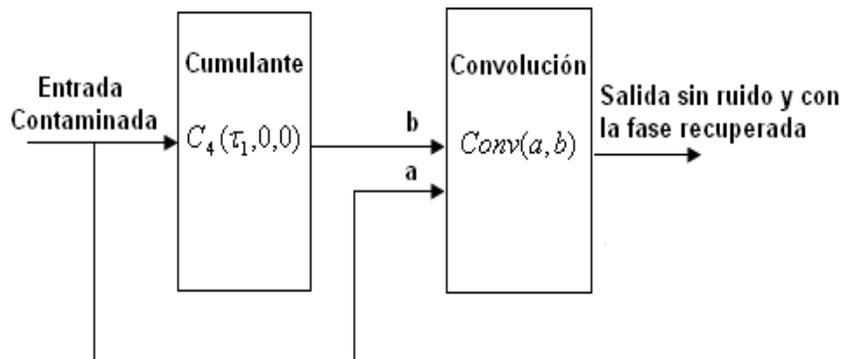


Figura. 1 Esquema general del modelo de estimación de parámetros propuesto

Haciendo un acercamiento teórico para demostrar la efectividad del modelo propuesto se tiene: $a(t) = A \cos(w_k t + \phi) + w(t)$ como la muestra inicial de señal contaminada y $b(\tau_1) = -\frac{3A^4}{8} \cos(w_k \tau_1)$ como la salida obtenida de evaluar $C_4(\tau_1, 0, 0)$. De acuerdo con el teorema de la convolución de dos señales definido por:

$$\begin{aligned}
 Conv_{ab} &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} a(t) \cdot b(-t + \tau) dt & (12) \\
 &= \int_{-T/2}^{T/2} [A \cos(w_k t + \phi) + w(t)] \cdot \\
 &\quad \cdot \left[-\frac{3A^4}{8} \cos(-w_k t + w_k \tau) \right] dt \\
 &= \int_{-T/2}^{T/2} -\frac{3A^5}{8} \cos(w_k t + \phi) \cdot \cos(-w_k t + w_k \tau) dt + \\
 &\quad + \int_{-T/2}^{T/2} -\frac{3A^4}{8} \cos(-w_k t + w_k \tau) \cdot w(t) dt \\
 Conv_{ab} &= -\frac{3A^5}{16} \cos(w_k \tau + \phi) & (13)
 \end{aligned}$$

Como se puede apreciar en la ecuación (13), se ha logrado obtener la señal original con su amplitud, frecuencia y fase respectivamente. Para comprobar experimentalmente y en la práctica el resultado obtenido, se lleva a cabo la implementación de este modelo sobre una arquitectura FPGA.

IMPLEMENTACIÓN SOBRE FPGA

A partir de lo mencionado anteriormente se inicia la implementación de la arquitectura para el modelo propuesto, según los parámetros de señal para los cuales el sistema deberá responder de acuerdo a las características de ruido presente en la señal útil a detectar. Todo el sistema fue sintetizado mediante la arquitectura FPGA de Xilinx SPARTAN-3AN, la cual posee 700 000 elementos lógicos, 20 multiplicadores embebidos y un reloj externo de 50 MHz, frecuencia base con la cual se trabaja el diseño. Cabe resaltar que toda la implementación fue descrita mediante código VHDL, haciendo de este proyecto reutilizable en cualquier otra arquitectura FPGA disponible.

Se decidió emplear como arquitectura hardware un FPGA y no un sistema secuencial, para disminuir el tiempo de procesamiento del sistema en general y paralelizar cada una de las funciones implementadas, ya que estas obedecen a una lógica secuencial desde el punto de vista matemático, lo cual su implementación mediante HDL, acelera considerablemente su compilación. Este aspecto comparativo se mostrará mas adelante

El entorno de desarrollo ISE se utilizó para generar el algoritmo correspondiente a la componente unidimensional del cumulante de cuarto orden de la señal así como para el cálculo de la convolución, además de la arquitectura hardware de control que compone el sistema, para la estimación de parámetros. Sobre esta plataforma conjuntamente se ejecutó y verificó todo el diseño, incluyendo tanto las señales simuladas, como las reales utilizadas, generadas estas últimas mediante un dispositivo externo.

ARQUITECTURA DE DISEÑO

Para el diseño del sistema se utilizó una arquitectura de segmentación paralela para el procesamiento de los datos, digitalizados mediante un sistema constituido por un módulo de conversión analógico digital (ADC) *PMOD AD1* de *Digilent. Inc* [8]. De este módulo se utiliza uno de los dos ADCs AD7476A que este contiene internamente. Mediante el método de muestreo uniforme en tiempo real se obtienen las muestras digitales correspondientes a la señal de entrada del sistema, para ser pasadas a la próxima etapa de procesamiento.

El PMOD AD1 tiene internamente dos filtros paso bajo (uno por cada canal de conversión) de 500 kHz de ancho de banda, de manera que la señal antes de entrar al conversor que se va a utilizar pasa por este filtro con el fin de que no aparezca el denominado efecto *aliasing*. La señal analógica a convertir está en el rango de 0 V a 3.3 V, siendo este último el voltaje de referencia del conversor tratado, cuya resolución es de 12 bits y su máxima frecuencia de muestreo de 1 MHz, siendo esta última la frecuencia a la cual se realizó el experimento.

Todo el sistema está condicionado por la señal “Inicio”(figura 2), la cual da comienzo al proceso. Esta señal posee un ciclo de repetición siempre inferior a la frecuencia de muestreo utilizada y se activa durante pulso estrecho positivo equivalente a la frecuencia base de trabajo del sistema, para este caso de 50 MHz.

Una vez digitalizadas las 1024 muestras en la memoria RAM de entrada, este módulo emite una señal de completamiento o “Full” (figura 2), la cual es controlada mediante el bloque “*FSM de Control 1*”, (figura 2). Este módulo es el encargado de gestionar todo el proceso de extracción de datos desde la memoria RAM de entrada y hacia el módulo “*Bloque de Cálculo*”, que efectúa la operación , para la obtención de la función de autocorrelación de la señal $C_{2y}(\tau)$ y del término izquierdo de la componente unidimensional del cumulante de cuarto orden $C_{4y-TI}(\tau)$.

Las muestras obtenidas a la salida de este módulo son enviadas hacia dos memorias RAM de salida hasta obtener el número máximo de muestras permisibles por cada una, para este caso 1024. El control de estas memorias se realiza mediante el bloque “*FSM de Cálculo y Control*”, el cual garantiza la funcionalidad de las operaciones pertinentes así como, el completamiento del cálculo de la componente $C_{4y}(\tau_1, 0, 0)$ del cumulante de cuarto orden. Efectuando la resta entre los dos términos de la ecuación (8), correspondiente uno, al resultado de $C_{4y-TI}(\tau)$ y el otro, al resultado de evaluar $C_{2y}(\tau)$ y multiplicar cada muestra obtenida por el término $3 \cdot C_{2x}(0)$, correspondiente a la potencia de la señal de entrada.

Una vez concluida esta etapa, se obtienen a la salida del módulo “*FSM de Cálculo y Control*”, las muestras correspondientes, con las componentes de amplitud y frecuencia originales, no así la componente de fase , la cual se obtendrá posteriormente mediante el proceso de convolución. Los datos obtenidos a la salida del bloque “*FSM de Cálculo y Control*”, son almacenados en una memoria RAM hasta tener su completamiento.

El proceso de llenado de la memoria RAM se controla mediante el módulo “*Bloque FSM de Control 2*”, a través de una señal de tipo “Full”, la cual emite un pulso positivo cuando se han almacenado las 1024 muestras de dato correspondiente. Al activarse la señal “Full”, comienza el proceso de convolución entre la salida correspondiente a la componente unidimensional del cumulante de cuarto orden y la señal de entrada, el cual es efectuado por el módulo “*Bloque de Cálculo Conv_{ab}(a,b)*” .

Cada muestra obtenida a la salida de este módulo es normalizada por el “*Bloque de Normalización*”, el cual divide cada muestra a procesar por el índice (τ) correspondiente, para lograr la uniformidad de la señal a la salida para posteriormente ser enviada hacia el controlador del conversor digital analógico (DAC), con una componente de DC implícita ya que el módulo DAC utilizado es unipolar. En la figura 2 se puede observar el diagrama en bloque del sistema.

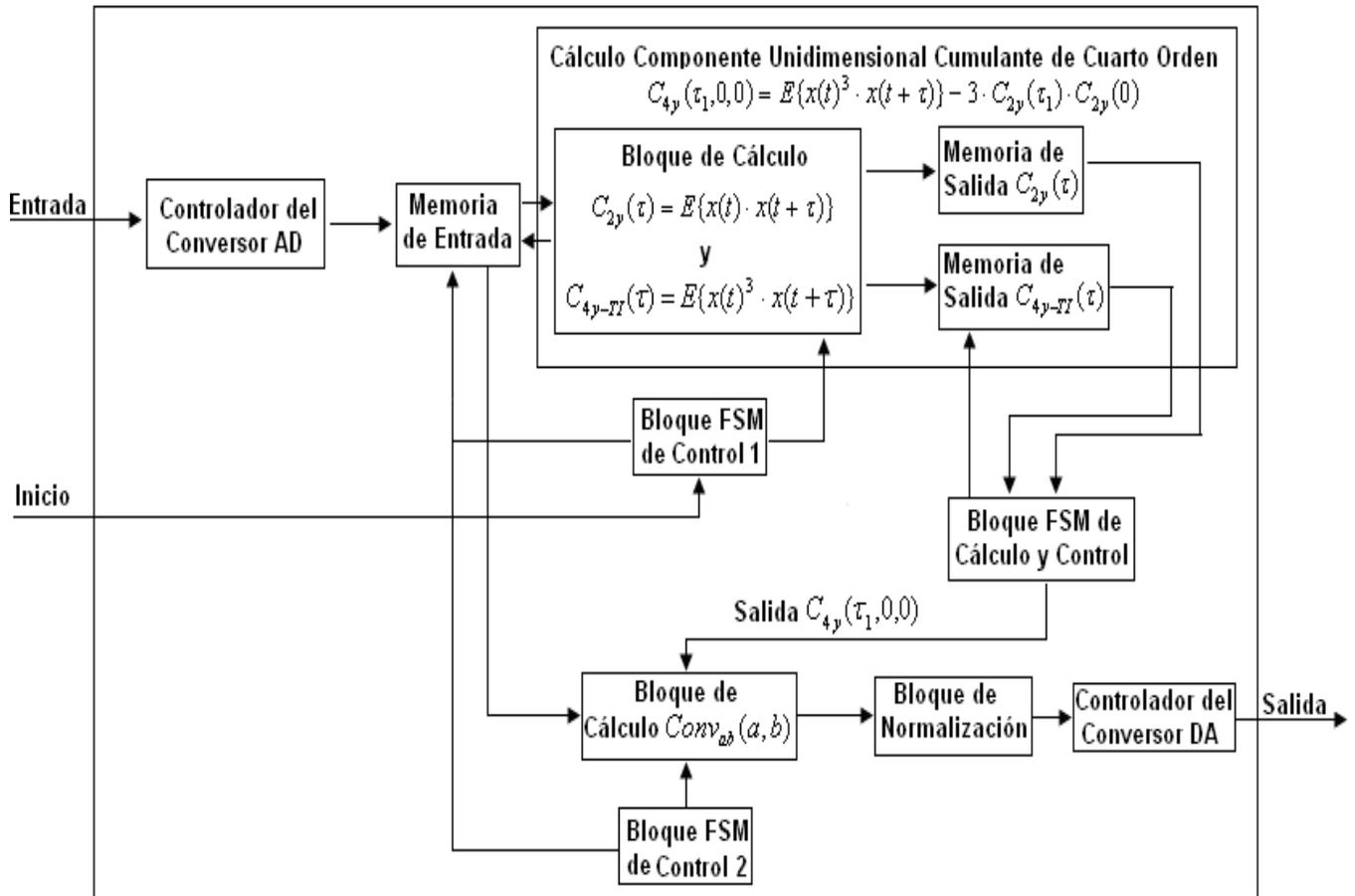


Figura. 2 Diagrama en Bloques del Sistema

Como ejemplo ilustrativo se muestra en la figura 3 el diagrama de tiempo del módulo diseñado en VHDL para el cálculo de la función de autocorrelación de la señal. Como se puede observar, todo el proceso comienza con el flanco de subida de la señal de “Inicio” y termina listo para extraer los datos de la memoria RAM de salida, con el pulso de la señal “FIN”, el cual habilita la señal de “LECTURA”, para comenzar el proceso de extracción de datos a la frecuencia base de trabajo para este caso 50 MHz.

ANÁLISIS DE RENDIMIENTO

Para hacer un análisis del rendimiento del sistema, se realiza una comparación con la implementación en VHDL del estadístico de orden superior (componente unidimensional del cumulante de cuarto orden), mediante un sistema basado en una arquitectura de dos procesadores Microblaze en paralelo utilizando la herramienta XPS de Xilinx, y otro modelo utilizando AccelDSP, la cual es una herramienta de síntesis proporcionada por Xilinx, que permite transformar un diseño en punto flotante desarrollado en Matlab, en un módulo hardware que puede ser implementado en un FPGA. Posee una interfaz de usuario fácil de usar que controla un ambiente integrado con otras herramientas de diseño, tales como: Matlab, Xilinx ISE, System Generator y otras como simuladores de código HDL y sintetizadores lógicos [9].

Como primer análisis se muestra en la figura 4 la gráfica del comportamiento de los recursos críticos consumidos por las implementaciones correspondientes, lo cual muestra un excesivo consumo de recursos en cuanto a los multiplicadores embebidos en el diseño basado en AccelDSP. Por otro lado, los demás recursos consumidos por las implementaciones son bastante similares destacando un menor consumo, la implementación tratada en este trabajo, dando además validez a la arquitectura propuesta.

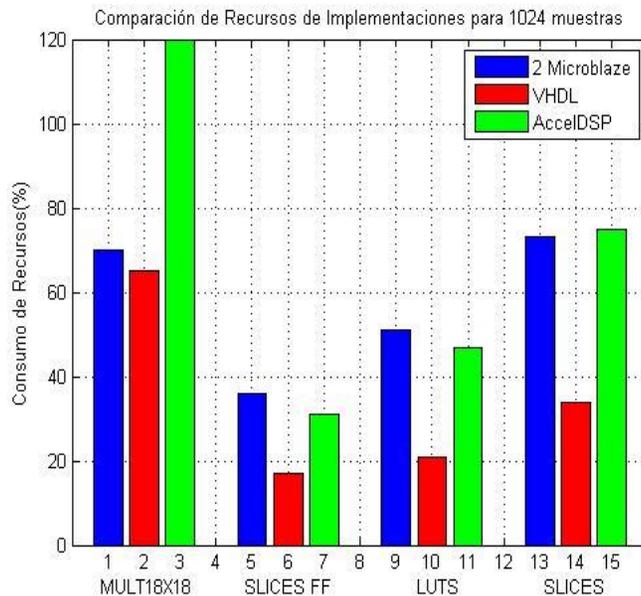


Figura. 4 Consumo de recursos de implementaciones en AccelDSP, XPS y VHDL

Una de las tareas indispensables a desarrollar en aplicaciones orientadas al procesamiento de datos en tiempo real, es la medición del tiempo necesario para que un determinado algoritmo se ejecute en el programa, en aras de lograr el acondicionamiento, y óptimo funcionamiento del sistema. Como última comparación en el diseño propuesto se establece un análisis del tiempo de ejecución y las frecuencias máximas de trabajo del diseño en VHDL, con respecto al diseño implementado tanto en AccelDSP como en XPS. El perfil de tiempo utilizado en XPS para la comparación se obtuvo utilizando la herramienta XMD (*Xilinx Module Debugger*), la cual, además, permite depurar el código de la aplicación para detectar y corregir errores presentes en la misma [10]. y cuyo resultado se obtuvo de un anterior trabajo, así mismo el perfil de tiempo obtenido en AccelDSP [10]. En la figura 5 se puede observar el análisis de tiempo para todas las implementaciones mencionadas.

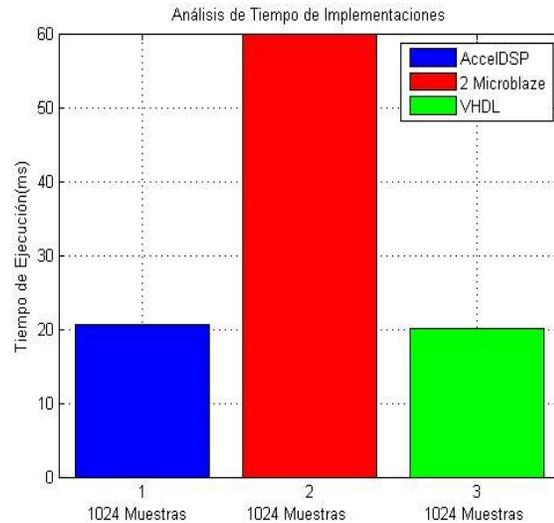


Figura. 5 Análisis de Tiempo de implementaciones en AcceIDSP, XPS y VHDL

En la figura anterior se observa el desbalance temporal de la implementación en XPS aún utilizando sistemas multinúcleo con respecto a la implementación en AcceIDSP y VHDL cuyo tiempo de ejecución es bastante similar, comprobando la afectividad del diseño propuesto.

RESULTADOS OBTENIDOS

Para la comprobación de los resultados se experimentó utilizando una señal coseno de 2 kHz de frecuencia, muestreada a 1 MHz, contaminada con ruido de naturaleza gaussiana generado a partir de la función *randn* de Matlab, cuyos datos se insertaron al sistema a través de un fichero de estímulos tipo texto, mediante el archivo de simulación “TestBench” del proyecto, utilizando para esto, la herramienta “*ISIM*” que proporciona Xilinx. La salida emitida por el sistema se envía a través de un fichero tipo texto hacia Matlab, donde se comprueban los resultados con el modelo descrito en esta herramienta.

En la figura 6 se muestran los resultados obtenidos tras evaluar el modelo propuesto descrito en VHDL y su comparación con el modelo simulado en Matlab. En la figura 6a se puede observar la muestra de señal contaminada utilizada primeramente, así mismo en la figura 6b se ilustra la salida obtenida tras evaluar el modelo propuesto descrito en VHDL (rojo) y su comparación con la señal de entrada (azul). Por otra parte en la figura 7a se observa la comparación entre el espectro de la señal, obtenido tras evaluar el diseño del sistema en VHDL y la señal de entrada, observándose la componente de frecuencia original de la señal a detectar (2 kHz), además en la figura 7b se muestra la señal de salida obtenida en VHDL y su comparación con el modelo simulado en Matlab, comprobándose la obtención de la fase correspondiente. Cabe resaltar que la diferencia de amplitud entre la señal de salida en VHDL y el modelo descrito en Matlab, se debe a que en el modelo generado en VHDL se trabaja con aritmética de tipo entero y se elimina el resto de la división durante el proceso de normalización, lo cual introduce cierto error a la salida.

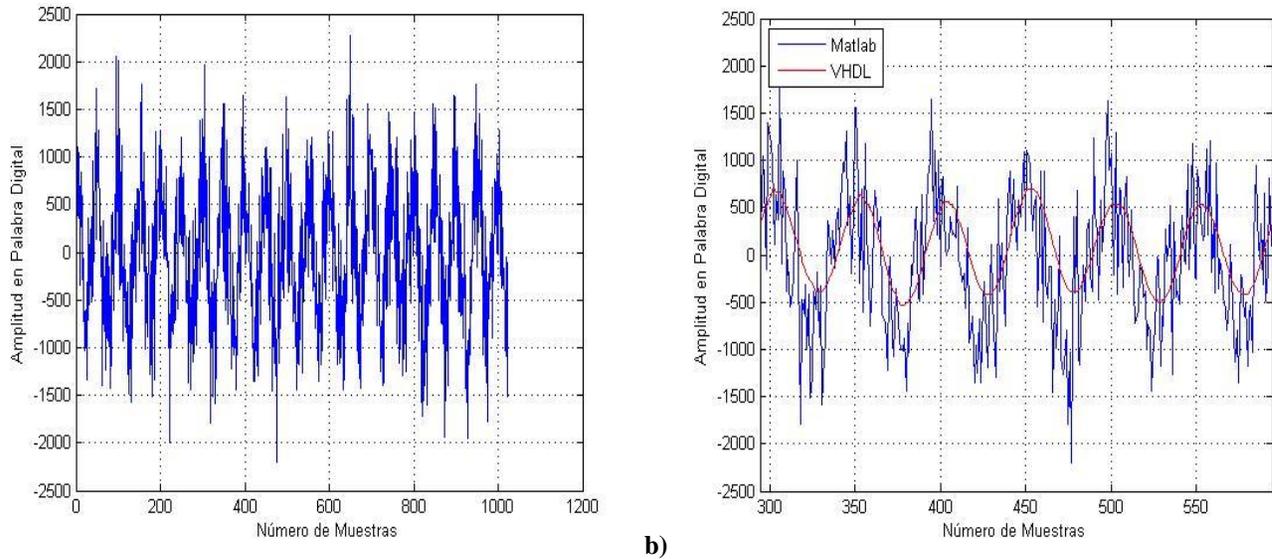


Figura. 6 Ilustración de a) Señal de Entrada b) Comparación entre la señal de entrada y la salida en VHDL

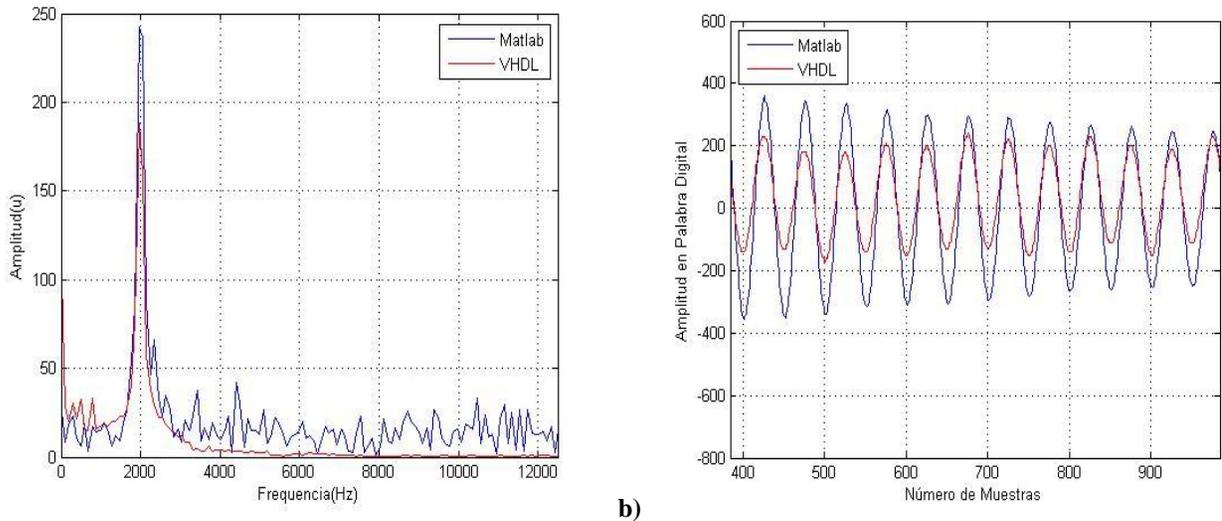


Figura. 7 Ilustración de a) Comparación del Espectro de la señal de entrada y la salida en VHDL b) Comparación del Modelo en Matlab y el Modelo en VHDL

Como último experimento se comprobó el sistema mediante un tono de 100 kHz de frecuencia obtenido a través de un generador de señales, al cual se le suma ruido generado a través de un módulo descrito en VHDL con una ganancia variable para medir el nivel de respuesta del sistema cuando la relación señal/ruido disminuye. Para la obtención del módulo generador de ruido se utiliza el kit de desarrollo SPARTAN-3E XC3S500E de Xilinx [11], cuya salida se asigna a uno de los pines de propósito general del kit.

Tanto la señal proveniente del generador como el ruido son sumadas mediante una configuración sumador no inversor compuesta por el amplificador operacional LM324 [12], la salida de este módulo es la señal a procesar por el sistema. Una vista de la aplicación se muestra en la figura 8.



Figura. 8 Vista General de la Aplicación

Los resultados obtenidos se pueden observar en la figura 9, mostrándose la señal de entrada (figura 9a) y la salida (figura 9b), obtenidas en el osciloscopio tras evaluar el modelo propuesto. Así mismo, en las figuras 10 se observa en detalle el espectro de la señal obtenido a la salida, el cual fue generado mediante la herramienta de Análisis del Osciloscopio en la PC, *NI LabVIEW SignalExpress Tektronix Edition*.

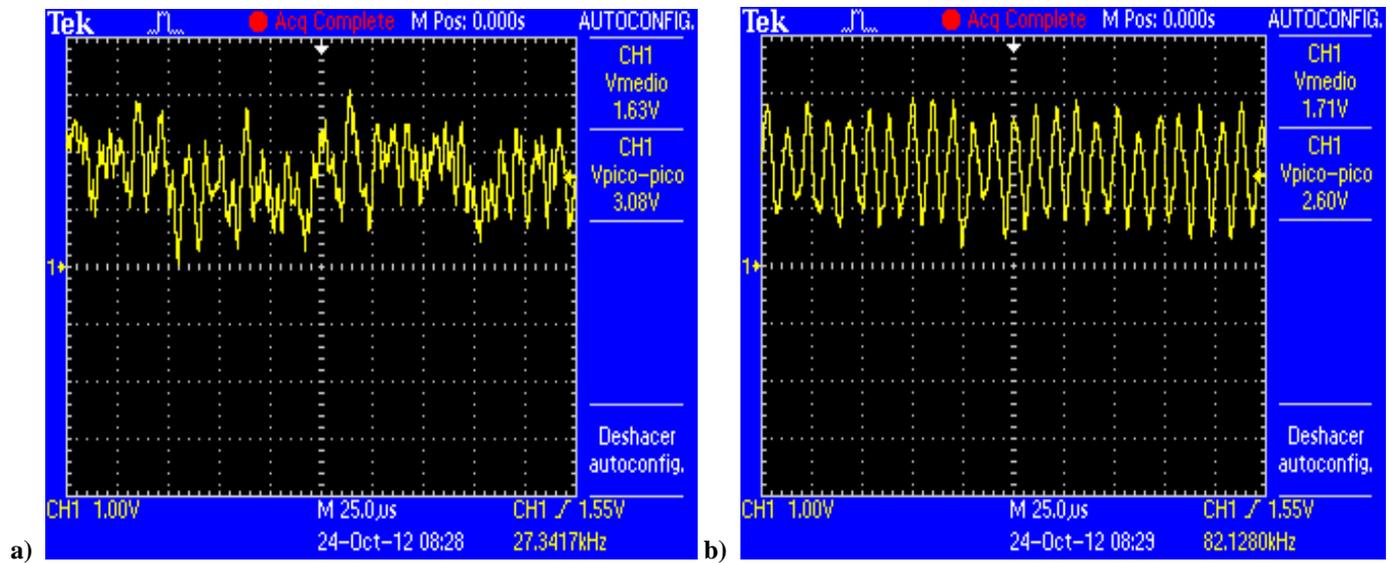


Figura. 9 Ilustración de a) Señal Contaminada b) Salida Obtenida

REFERENCIAS

1. **Roy M. Howard:** Principles of Random Signal Analysis and Low Noise Design, Wiley & Sons Ltd. 2009.
2. **S.V. Narasimhan, Nandini B. Mallick, Ratana Chaitanya:** Improved Phase Estimation based on Complete Bispectrum and modified group delay, Signal, Image and Video Processing, February 2008.
3. **Min Geng, Huaqing Liang, Junwu Wang:** Research on Methods of Higher-order Statistics for Phase Difference Detection and Frequency Estimation, 4th International Congress on Image and Signal Processing, 2011.
4. **Amar Kachenoura, Laurent Albera, Jean-Jacques Bellanger, Lotfi Senhadji:** Non-minimum phase identification based on higher order spectrum slices, HAL author manuscript IEEE Transactions on Signal Processing, 2008, 9 pages
5. **Jerry M. Mendel:** Signal Processing with Higher-Order Spectra, IEEE Signal Processing Magazine Julio 1993.
6. **Josep M[§] Salavedra Molí:** Técnicas de Speech Enhancement considerando Estadísticas de Orden Superior, Tesis Doctoral, Barcelona, Junio 1995.
7. **Izzet Ozcelik:** Blind Deconvolution of Music Signals using Higher Order Statistics, 17th European Signal, Processing, Conference (EUSIPCO 2009), Glasgow, Scotland, August 24- 28, 2009.
8. **Digilent PmodAD1:** obtenido de: http://www.digilentinc.com/Data/Products/PMOD-AD1/Pmod%20AD1_rm.pdf, Julio 2011
9. **AccelDSP Synthesis Tool User Guide:** obtenido de: http://www.xilinx.com/acceldsp_user.pdf, Noviembre 2010
10. **Miguel Enrique Iglesias Martínez, Fidel Ernesto Hernández Montero:** Cancelación de Ruido Aplicando Estadística de Orden Superior y Sistemas Multiprocesador sobre FPGA, Libro de Trabajos Congreso Argentino de Sistemas Embebidos, SASE 2012, Buenos Aires, Argentina Agosto 2012, ISBN: 978-987-9374-82-5, <http://www.sase.com.ar>
11. **Spartan 3E Starter Kit Reference Manual:** obtenido de: <http://www.xilinx.com>, Septiembre 2012
12. **Bárbaro M. López Portilla:** Analizador de Redes Eléctricas Basado en FPGA, Tesis de Maestría, La Habana, Cuba, 2012.

AUTORES

Miguel Enrique Iglesias Martínez: Ingeniero en Telecomunicaciones y Electrónica. Máster en Sistemas Digitales, Profesor de la Universidad de Pinar del Río, Cuba. migueliglesias2010@gmail.com

Bárbaro Maykel López-Portilla Vigil: Ingeniero en Telecomunicaciones y Electrónica. Máster en Sistemas Digitales. Profesor de la Universidad de Pinar del Río, Cuba. barbaro@tele.upr.edu.cu

Fidel Ernesto Hernández Montero: Ingeniero en Telecomunicaciones y Electrónica. Doctor en Ciencias Técnicas. Profesor de la Universidad de Pinar del Río, Cuba. fidel@tele.upr.edu.cu