

**ND6006UNLP**

**UNIVERSIDADE:** Universidad Nacional de La Plata. Facultad de Ingeniería/ CETAD

**AUTORES:** Joaquín Requena (jrequena@ing.unlp.edu.ar); Ing. José Rapallini  
(jsrap@ing.unlp.edu.ar)

## **FILTROS DIGITALES DE USO EN EQUIPOS DE MEDICIÓN PARA ENSAYOS ELÉCTRICOS**

### **RESUMEN**

Se presenta el diseño de un filtro digital versátil, donde se pueda variar el orden y la longitud de la ventana de datos dentro del entorno de los Filtros Coseno, Seno y Fourier. El dispositivo será utilizado en la detección de señales de frecuencias bajas (50 Hz) enmascaradas en ruido, presentes en todo ensayo de máquinas o elementos eléctricos. En este trabajo se realiza el diseño con lógica programable sobre dispositivos comerciales, optimizando las características de cada filtro para los componentes utilizados y realizando pruebas de laboratorio para ver la calidad de filtro obtenido. Con este procedimiento se trata de obtener una Descripción del Hardware (HDL) que contenga la mayor cantidad de casos de utilización práctica posible y a partir del HDL realizar el diseño del circuito de aplicación específica (ASIC).

Palabras clave: Filtro digital; Codiseño hardware/software; Lógica programable; AHDL

Palavras-chave: Filtro digital; Codesenho hardware/software; Lógico programável; AHDL

### **1 INTRODUCCIÓN**

En los ensayos eléctricos que se realizan sobre diferentes tipos de máquinas y/o elementos, se presenta la necesidad de medir diversos parámetros, a través de los cuales se obtienen conclusiones directas o indirectas de las características del objeto ensayado.

Las señales que se obtienen en el ensayo pueden contener distorsiones que dificultan la lectura en forma correcta de los parámetros requeridos, motivo por el cual en la mayoría de los equipos de medida se debe optimizar y mejorar la medición y el análisis a través del procesamiento digital de las señales[1]. Se utilizan diferentes algoritmos de filtrado, sean estos del tipo FIR (Finite Impulse Response) o IIR (Infinite Impulse Response). Para poder establecer los criterios de diseño se analizarán distintos tipos de herramientas digitales, tales como el algoritmo numérico para el cálculo de funciones avanzadas FFT (Fast Fourier Transform) para el cómputo de la DFT (Discrete Fourier Transform).

En la línea de tareas del grupo se desarrollo el diseño de un sistema de medición de laboratorio para la medición de impedancia, desarrollado a través de un sistema de PC- Placa ADQ – Software [3], [5], que fue base para el presente desarrollo. Dado que el mismo fue realizado con la metodología de Codiseño HW/SW [4], se pueden reutilizar algunos de los pasos realizados para el desarrollo del presente dispositivo.

## 2 FILTROS DIGITALES

La razón por la cual deben ser filtradas las señales de entrada, es que estas están contaminadas con ruido. Este dependerá de las condiciones impuestas por la red en la cual el sistema sense tensión y corriente. Por ejemplo, cuando ocurre una falla (cortocircuito) en un sistema eléctrico, el comportamiento resistivo-inductivo dominará la situación y la tensión y corriente se convertirán en sinusoides con una característica exponencialmente decreciente (o creciente) hasta llegar al nuevo estado estacionario. Debido al carácter inductivo de la red eléctrica este comportamiento exponencial afectará fuertemente a la corriente, más rara vez afectará a la tensión.

Las cargas no lineales, los transformadores de potencia y los transformadores de instrumentación, entre otros introducen armónicos de la frecuencia fundamental en la línea, lo cual también es visto como ruido.

Las compensaciones capacitivas en serie ingresan en la línea fenómenos transitorios de baja frecuencia, al igual que los transformadores de tensión de acople capacitivo. La naturaleza sobreamortiguada de estos transitorios, hace que se asemejen a offset de continua. Existen también fenómenos de reflexión de potencia, que generan ruido en tensión.

Todos estos hechos generan ruido que se superpone a las componentes fundamentales de 50Hz de tensión y corriente, las cuales son de nuestro interés, pues como ya se ha mencionado el cálculo de la impedancia se realiza sobre la base del cociente de los fasores de tensión y corriente (a 50 Hz por supuesto). Surge entonces la necesidad de filtrar las señales de entrada para rescatar las componentes deseadas.

Existen dos tipos de filtros digitales[6], clasificados según su respuesta al impulso: los filtros de respuesta impulsional finita (Finite Impulse Response, FIR) y los de respuesta impulsional infinita (Infinite Impulse Response, IIR). Los filtros FIR sirven en sistemas que requieren una *respuesta con fase lineal exacta*, lo cual, excepto en algunos casos muy específicos, los filtros IIR no pueden alcanzar. En contraste con los filtros IIR, los FIR también se benefician de una *respuesta en frecuencia inherentemente estable*. Sin embargo, es importante considerar el costo potencial de hardware adicional versus un filtro IIR con una respuesta en frecuencia equivalente. Los filtros IIR, los cuales son inherentemente inestables, pueden trabajar en sistemas que toleran su distorsión de fase intrínseca. La salida de un filtro FIR depende sólo de una cantidad finita de muestras pasadas de la entrada, mientras que la de los filtros IIR depende de todas las muestras anteriores de la entrada.

## 2.1 RESPUESTA EN FRECUENCIA DE LOS FILTROS DIGITALES

Así como en los sistemas de tiempo continuo, la respuesta en frecuencia es la transformada de Fourier de la respuesta al impulso del sistema, en los sistemas discretos la respuesta en frecuencia es la **Transformada de Fourier de Tiempo Discreto (TFTD)** de la respuesta al impulso del sistema, quizás la propiedad más destacada y que vale la pena mencionar aquí es la periodicidad de la TFTD. En efecto para cualquier señal discreta su TFTD será periódica en  $s$  con período igual a 1.

Debido al carácter de sumatoria infinita de la TFTD resulta imposible calcular en forma numérica dicha transformada por medios computacionales. Más aún resulta irrazonable almacenar señales discretas de largo infinito en la memoria de la computadora. Por este motivo se definió una nueva transformada capaz de trabajar con señales de largo finito: esta es la **Transformada Discreta de Fourier (TDF)**.

Para realizar el cálculo de la TDF se ha propuesto un algoritmo numérico llamado la **Transformada Rápida de Fourier (FFT)** muy utilizado en el cálculo por computadora.

La TFTD será igual a la transformada de la señal de tiempo continuo dentro del intervalo  $[-f_m, f_m]$  sólo afectado el módulo de la original por el factor  $f_s$ . Fuera del mismo se extienden réplicas de la transformada de tiempo continuo, quedando así el espectro periódico característico de las señales de tiempo discreto. Se observa también que el valor  $s = 1$  se corresponde con la frecuencia  $f_s = 2 \cdot f_m$ .

## 2.2. FILTROS FIR UTILIZADOS EN ENSAYOS DE EQUIPOS Y/O COMPONENTES ELÉCTRICOS

Existen también algunos tipos de filtros particulares, muy utilizados a la hora de la medición en equipos de ensayos de elementos eléctricos. Estos son los llamados filtros *de Fourier*, *Coseno* y *Seno*.

### 2.2.1 Filtro Coseno

Los coeficientes que determinan la respuesta al impulso del filtro coseno vienen dados por la ecuación:

$$h_c[n] = \frac{2}{N} \cos\left(\frac{2\pi(n+1)}{N_c}\right) \quad \text{con } 0 \leq n \leq N-1.$$

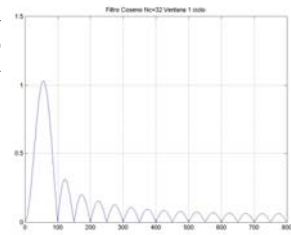
Una respuesta en frecuencia típica de este tipo de filtro se muestra en la figura 2.1.

Esta respuesta en frecuencia es la de un filtro coseno con 32 muestras por ciclo ( $N_c$ ) y ventana de 1 ciclo ( $N=N_c$ ). La gráfica ha sido realizada en el entorno MATLAB, utilizando el

comando **fft** (transformada rápida de Fourier) con 4096 puntos para calcular la transformada de Fourier de tiempo discreto (respuesta en frecuencia) de la respuesta al impulso del filtro.

Para el caso que  $N=k \cdot N_c$  con  $k$  entero, el factor  $2/N$  asegura que la ganancia del filtro a la frecuencia fundamental sea 1.

Figura 2.1 Respuesta en frecuencia del filtro Coseno con  $N_c=32$  y ventana de 1 ciclo



### 2.2.2 Filtro seno

Es análogo al anterior sólo que en este caso los coeficientes se calculan con la fórmula:

$$h_s(n) = \frac{2}{N} \cdot \text{sen}\left(\frac{2 \cdot \pi \cdot (n+1)}{N_c}\right) \quad \text{con } 0 \leq n \leq N-1$$

valen para este filtro todas las consideraciones realizadas con el filtro coseno. Su respuesta en frecuencia típica con 32 muestras por ciclo y ventana de 1 ciclo es:

La gráfica fue realizada con MATLAB y la respuesta en frecuencia se calculó con la **fft** de 4096 puntos. Comparando las características

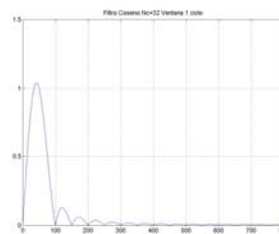


Figura 2.2 Respuesta en frecuencia del filtro Seno con  $N_c=32$  y ventana de 1 ciclo

buscadas para los filtros utilizados en protecciones por medición de impedancia con las respuestas en frecuencia de las Fig. 2.1 y 2.2 se observa que los filtros coseno y seno satisfacen varias de las nombradas características. En primer lugar en la banda de paso la ganancia para la componente de 50 Hz es uno. Luego el filtro posee ceros en los armónicos de la frecuencia fundamental (múltiplos de 50 Hz). Además ambos rechazan la componente de continua. Tanto el filtro *Seno* como el *Coseno* cumplen con las características de filtrado deseadas obteniendo la componente de dicha frecuencia.

### 2.2.3 Filtro de Fourier

Este filtro está compuesto por un filtro *Seno* y uno *Coseno*. Los coeficientes de su respuesta al impulso son números complejos y quedan determinados por la siguiente expresión:

$$h_F(n) = \frac{2}{N} \cdot \left[ \cos\left(\frac{2\pi(n+1)}{N_c}\right) + j\text{sen}\left(\frac{2\pi(n+1)}{N_c}\right) \right]$$

con  $0 \leq n \leq N-1$ .

El filtro de Fourier aprovecha la característica de ortogonalidad de los coeficientes de los filtros *Seno* y *Coseno* para obtener a la salida del mismo el faser de la señal de entrada. En este caso el filtro posee coeficientes complejos, con lo cual luego de la suma de convolución la señal de salida será también compleja representando el faser buscado.

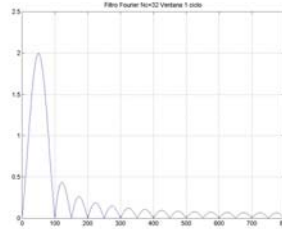


Figura 2.3 Respuesta en frecuencia del filtro de Fourier con  $N_c=32$  y ventana de 1 ciclo.

Repitiendo lo hecho para los filtros *Seno* y *Coseno* se presenta en la figura 2.3 la gráfica de la respuesta en frecuencia calculada en MATLAB con 4096 puntos de la *fft*,  $N_c=32$  y ventana de un ciclo.

Como era de esperarse por estar formado por los filtros antes nombrados, el filtro de Fourier posee características similares respecto a los algoritmos del *Seno* y el *Coseno*, en cuanto al rechazo de las componentes armónicas de la frecuencia fundamental y al rechazo de la continua. Respecto a la ganancia se nota que para  $f=50$  Hz la ganancia deja de ser 1 para convertirse en 2, y que para las demás frecuencias también la ganancia es mas elevada. Esto es innecesario puesto que lo que se busca es filtrar las componentes no deseadas. Esto último se subsana dividiendo los coeficientes del filtro por 2.

### 3 EVALUACIÓN DEL FILTRO DIGITAL

Para saber que filtro se ha de elegir entre los mencionados anteriormente se realizó una evaluación de cada uno a partir de su estado estable y de su desempeño en la respuesta transitoria[4]. El filtro se encuentra en un estado transitorio cuando la ventana de datos del filtro cubre parcialmente datos de pre-falla y parcialmente datos de post-falla, después que su ventana de datos cubre totalmente los datos post-falla, el filtro está en estado estable post-falla. Como deducción se obtuvo que los aspectos más importantes del análisis son:

Al aumentar la frecuencia de muestreo ( $N_c$ ) no se logra una gran mejora en la respuesta en frecuencia del filtro y se encarece el diseño por la necesidad de aumentar la velocidad de cómputo. Eligiendo  $f_s=1600$ Hz se obtiene un compromiso aceptable entre la respuesta en frecuencia y la velocidad para realizar los cálculos.

1. Cuanto mayor sea el tamaño de la ventana de datos, más selectivos será el filtro en la banda pasante, pero a costa de tener que aumentar la cantidad de componentes (sumadores, ganancias y retardos) al aumentar el orden del filtro. En particular con ventanas de datos de 1 ciclo se consigue rechazar totalmente todos los armónicos de la frecuencia fundamental lo cual es una característica deseable.
2. El filtro Seno posee una muy buena característica en frecuencia en la banda de rechazo, pero es el que peor comportamiento transitorio tiene al trabajar con señales exponenciales. Por su parte, el filtro Fourier responde mejor en frecuencia, pero el filtro Coseno lo supera en cuanto a respuesta transitoria para señales de entrada con componentes exponenciales.

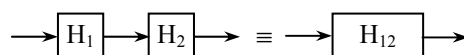
Basado en el análisis de los resultados obtenidos, se decidió hacer la realización e implementación de un **filtro Coseno con 32 muestras por ciclo** ( $f_s=1600\text{Hz}$ ) y ventana de datos de 1 ciclo.

#### 4 REALIZACIÓN DEL FILTRO DIGITAL

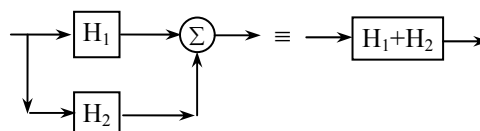
En la práctica hay dos formas de realizar filtros digitales, una es mediante software y la otra por medio de hardware. En los dos casos se debe hacer un diagrama con las operaciones a realizar. En software se habla de un diagrama de flujo, mientras que en hardware es un diagrama de bloques que especifica los elementos del circuito y sus interconexiones.

Una correcta elección del diagrama de bloques puede optimizar significativamente las prestaciones de la realización (tiempo de computación, memoria necesaria, minimizar los efectos de cuantización, etc). Algunas propiedades de los diagramas de bloques son:

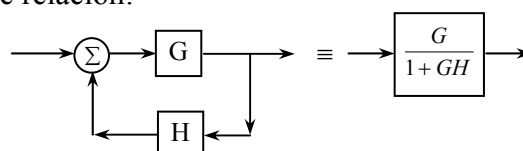
- *Conexiones en cascada:* La función de transferencia total de una conexión en cascada es el producto de las funciones de transferencia individuales.



- *Conexiones en paralelo:* La función de transferencia total de una conexión en paralelo es la suma de las funciones de transferencia individuales.



- *Conexión en realimentación:* La salida se realimenta en la entrada directamente o a través de otros subsistemas. La función de transferencia total viene dada por la siguiente relación:

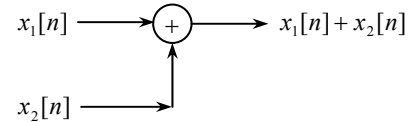


Ahora, estos bloques pueden ser realizados usando tres tipos básicos de elementos:

- *Unidad de retardo*: el propósito de este elemento es mantener su entrada por unidad de tiempo (físicamente igual al intervalo de muestreo  $T$ ) antes de que este sea llevado a la salida.

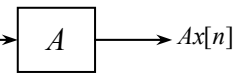
La unidad de retardo puede ser implementada en hardware por un registro de datos, el cual mueve su entrada a la salida cuando recibe la señal de reloj. En software, es implementado por una variable de almacenamiento, la cual cambia su valor cuando es instruida por el programa.

- *Sumador*: el propósito de este elemento es sumar dos o más señales presentes en la entrada en un tiempo específico.



- *Multiplicador*: el propósito de este elemento es multiplicar una señal (una cantidad diversa) por un número constante.

A partir de la función de transferencia o respuesta impulsional del filtro Coseno se tiene:



n	hc[n]	n	hc[n]	n	hc[n]	n	hc[n]
0	0.0613	8	-0.0122	16	-0.0613	24	0.0122
1	0.0577	9	-0.0239	17	-0.0577	25	0.0239
2	0.0520	10	-0.0347	18	-0.0520	26	0.0347
3	0.0442	11	-0.0442	19	-0.0442	27	0.0442
4	0.0347	12	-0.0520	20	-0.0347	28	0.0520
5	0.0239	13	-0.0577	21	-0.0239	29	0.0577
6	0.0122	14	-0.0613	22	-0.0122	30	0.0613
7	0	15	-0.0625	23	0	31	0.0625

$$h_c[n] = \frac{1}{16} \cos\left(\frac{2\pi(n+1)}{32}\right)$$

$$n = 0, \dots, 31.$$

Por lo tanto los valores de los coeficientes son los de la tabla 1.

Observando, se puede hacer:

$C_0 = a_0 = a_{30} = -a_{16} = -a_{14}$ ,  $C_1 = a_1 = a_{29} = -a_{17} = -a_{13}$ ,  $C_2 = a_2 = a_{28} = -a_{18} = -a_{12}$ ,  $C_3 = a_3 = a_{27} = -a_{19} = -a_{11}$ ,  $C_4 = a_4 = a_{26} = -a_{20} = -a_{10}$ ,  $C_5 = a_5 = a_{25} = -a_{21} = -a_9$ ,  $C_6 = a_6 = a_{24} = -a_{22} = -a_8$  y  $C_7 = a_{15} = -a_{31}$ .

De tal manera que la función de transferencia queda representada como:

$$H_c(z) = [C_0(1 - z^{-14}) + C_1z^{-1}(1 - z^{-12}) + C_2z^{-2}(1 - z^{-10}) + C_3z^{-3}(1 - z^{-8}) + C_4z^{-4}(1 - z^{-6}) + C_5z^{-5}(1 - z^{-4}) + C_6z^{-6}(1 - z^{-2}) - C_7z^{-15}](1 - z^{-16})$$

A partir de esta función y de los elementos y las propiedades mencionadas de los diagramas de bloque, se obtiene la realización tal como se observa en la figura 4.1.

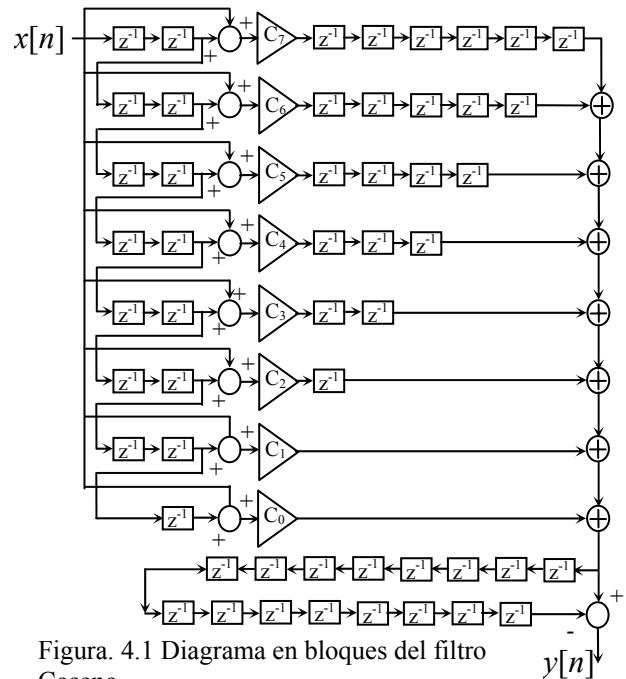


Figura. 4.1 Diagrama en bloques del filtro Coseno.

## 5 IMPLEMENTACIÓN

Para el diseño se utilizó componentes de ALTERA, que ofrece familias de dispositivos de lógica programable (PLD) ajustables a las necesidades de cada uno, incluyendo las familias Stratix, APEX II, APEX, Hardcopy, Mercury, ACEX, FLEX y MAX, y las soluciones Excalibur. Estos dispositivos son de muy altas velocidades, además, ofrecen características avanzadas a nivel sistema, tal como programables sobre la misma placa que esta montado (ISP), soportado por el Test Standard Jam y lenguaje de programación (STAPL), JEDEC Standard JESD-71, y bloques de sistemas embebidos (ESBs), los cuales pueden ser configurados como lógica o como diferentes tipos de memoria. En el laboratorio se cuenta con EPLD MAX7128 y Flex 10K.

Para completar la solución de diseño, ALTERA también tiene poderosas herramientas de desarrollo. Una de ellas es el software Quartus que ofrece edición a nivel bloques, integración con software de control de fuente standard y expandido soporte a las megafunciones para acortar los ciclos de diseño. Otra herramienta, es el software MAX+PLUS II que provee un desarrollo integrado fácil de usar que soporta la familia de dispositivos FLEX, MAX y ACEX 1K. Es compatible con las herramientas de la industria standard EDA y ofrece interfase directa VHDL, Verilog HDL y EDIF, así como también su propio lenguaje AHDL, esta última es la utilizada en la implementación del proyecto.

Para realizar la primera implementación del filtro digital se utilizaron las características de un filtro Coseno, limitando el número de muestras y el número de coeficientes que caracterizan al filtro, con el objetivo de implementarlo en dos componentes MAX7128. De esta manera se obtiene que:  $C_0=a_0=a_{14}=-a_8=-a_6$ ,  $C_1=a_1=a_{13}=-a_5=-a_9$  y  $C_2=-a_0=a_{15}$ , la función de transferencia estará dada por:

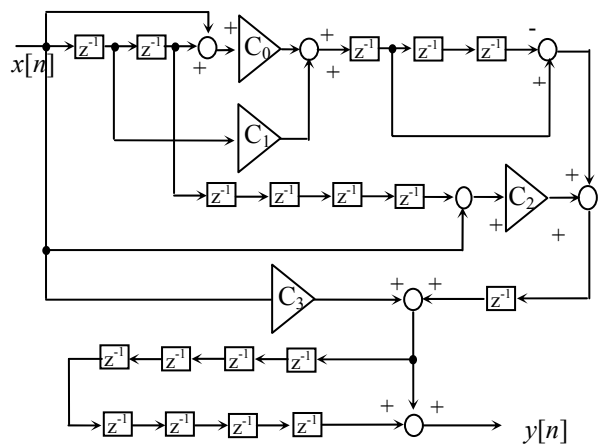
$$H_c(z) = [C_0(1 - z^{-6}) + C_1z^{-1}(1 - z^{-4}) - C_2z^{-7}](1 - z^{-8})$$

Mediante la configuración en cascada se obtiene el siguiente esquema de bloques de la figura 5.1.

En el diseño funcional y estructural del filtro digital, se eligió una herramienta de descripción de hardware que es utilizada para circuitos electrónicos de alta complejidad, que es más poderosa que los tradicionales esquemas eléctricos. Estos lenguajes de descripción de hardware (HDLs) han surgido como solución para el diseño de circuitos, dado que facilitan el diseño parametrizado de sistemas, el reuso de diseños y el mantenimiento de un proyecto durante su vida útil. Además, al realizar diseños utilizando lógica programable y HDLs está la opción de utilizar un lenguaje orientado a una tecnología en particular, o utilizar un lenguaje estándar, como VHDL o Verilog.



Cualquiera sea la decisión que se tome, se tienen ventajas y desventajas. Si se opta por un HDL estándar, es posible contar con una amplia y poderosa oferta de herramientas de software y poder realizar diseños que puedan ser sintetizados directamente en cualquier otra familia de lógica programable o incluso transformados en ASICs; si se opta por un HDL específico es posible realizar diseños



**Fig. 5.1.** Diagrama en bloques del filtro Coseno con limitación de muestras

más eficientes con un uso mucho más controlado de las particularidades que ofrece cada tecnología, aunque la oferta de compiladores, simuladores, etc., es mucho más limitada y, en general, acotada al fabricante.

El HDL específico de ALTERA, llamado AHDL, permite la descripción del hardware a través de su comportamiento descrito en modo textual, en lugar de hacerlo a través de un circuito esquemático. Además, se pueden utilizar ecuaciones lógicas, variables, constantes, funciones, operadores aritméticos, tablas de verdad y expresiones condicionales, en forma similar a un lenguaje de software de alto nivel.

Para realizar los bloques con los que se construye el filtro digital se utilizan las librerías de módulos parametrizados, LPM (Library of Parameterized Modules), que se encuentran en el mismo software MAX+PLUS II, de esta manera se tiene:

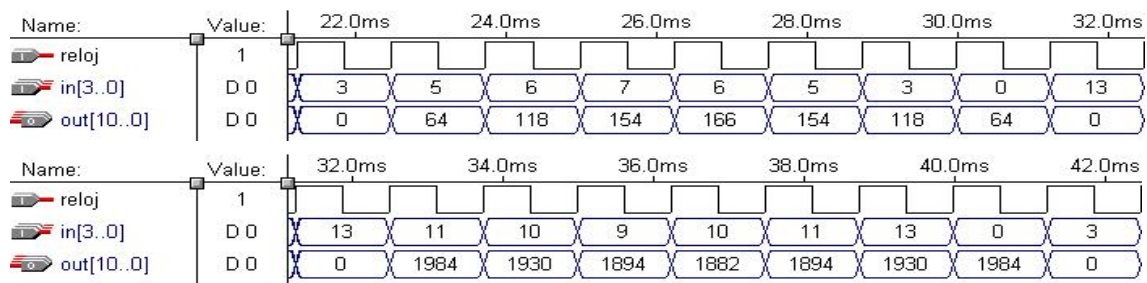
1. **Retardo unitario: lpm\_dff**, es una determinada cantidad de flaps-flops tipo D que responden a una entrada de determinada longitud en bits. Estos son todos los retardos *delay* y *dlay*.
2. **Multiplicador: lpm\_mult**, realiza la operación de multiplicación entre dos señales de determinada longitud en bits. Se utilizan tres multiplicadores, que realizan el producto entre la señal y los coeficientes - más adelante se explica por qué son tres y no cuatro como corresponde por la cantidad de coeficientes -.
3. **Sumador o restador: lpm\_add\_sub**, realiza la operación de suma o resta, según se establezca, de dos señales de determinada longitud en bits.

De esta forma se tiene una entrada que es la señal **in[3..0]**, la salida **out[10..0]**, una señal de clock denominada **reloj**, para mantener las señales sincronizadas, y de los cuatro coeficientes se utiliza tres, porque un coeficiente se considera igual al uno digital, entonces no es necesario multiplicarlo, estos puedan programarse para desarrollar diferentes tipos de

filtros, logrando utilizar solo dos EPLDs como se observa del reporte que entrega la compilación:

	In	Out	Bidir	Shareable			
Device	Pins	Pins	Pins	LCs	Expanders	% Utilized	
EPM7128	19	12	0	124	55	96 %	
EPM7128	9	17	0	120	62	93 %	
<b>TOTAL:</b>	<b>28</b>	<b>29</b>	<b>0</b>	<b>244</b>	<b>117</b>	<b>95 %</b>	
<b>User Pins:</b>	<b>5</b>	<b>11</b>	<b>0</b>				

Los valores de la señal de entrada “que se muestrean” con que se simula en MATLAB, se aplica como entrada al entorno MAX+PLUS II. Obteniéndose:



Cuya representación en el tiempo se muestra en la figura 5.3. Donde se observa que la señal de entrada posee ruido (línea de color azul) y a la salida se obtiene la respuesta sinusoidal de 50 Hz, excepto el 1º período que es el primer análisis donde las muestras pasadas son zeros (línea de color rojo).

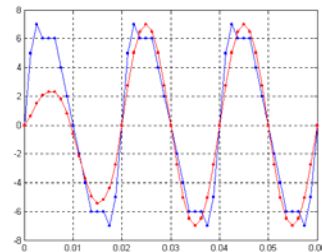


Figura 5.3  
Respuesta en el tiempo.

De esta manera, comparando las señales obtenidas en el cálculo de MATLAB con las del simulador, se observan los mismos resultados. Se puede apreciar que debido al espacio que nos limita las EPLD la resolución de los números no es la mas acorde ya que solo se usan 4 bits para representar las señales de entradas. Es por eso que se plantea realizar la misma realización del filtro digital sobre un dispositivo FLEX.

Por otra parte, si se dispone de dispositivos FLEX, la compilación nos arroja el siguiente reporte:

Device	In Pins	Out Pins	Bidir Pins	Memory Bits	Memory %Utilized	LCs	LCs %Utilized
EPF10K10LC84-3	18	24	0	128	2 %	519	90 %
<b>User Pins:</b>	<b>18</b>	<b>24</b>	<b>0</b>				

Se observa que todo el algoritmo que desarrolla al filtro entra en un solo dispositivo. Pero para aprovechar aun más las prestaciones que ofrece la FLEX, se puede desarrollar una implementación diferente que nos dará un mejor desarrollo del filtro a utilizar. Teniendo en cuenta los siguientes bloques:

1. *Bloque de memoria*, que guarda los coeficientes correspondientes al tipo de filtro, estos valores pueden ser modificados escribiendo sobre la memoria con la habilitación de la entrada prog\_coef.
2. *Bloques de almacenamientos* de la señal de entrada.
3. *Bloque multiplexor*, se encarga de seleccionar el bloque de almacenamiento de la señal de entrada correspondiente para la multiplicación.
4. *Bloque multiplica*, se encarga de multiplicar la señal de entrada con el respectivo coeficiente.
5. *Bloque suma*, se encarga de sumar los términos del producto.
6. *Bloque acumula*, se encarga de almacenar los términos de la suma para obtener el resultado final.
7. *Bloque de control*, que se encarga de sincronizar las señales de cada bloque.

El diagrama de bloques que interconecta a los bloques mencionados es:

En la prueba se realiza el análisis con una señal de entrada de 8 bits por muestras, con una cantidad de 16 muestras por ciclo de la señal a analizar, es decir, una frecuencia de muestreo de  $f_s=800\text{Hz}$ , y el formato a utilizar es complemento a dos.

Para realizar el cálculo de la salida, se realizan 16 multiplicaciones de los pares  $h(k)$ ,  $x(n-k)$ , en el bloque Multiplica, la  $k$ -ésima multiplicación se realiza direccionando la posición  $k$  del bloque memoria y la posición  $k$  del Multiplexor. Los bloques Suma y Acumula corresponden a un circuito acumulador, el cual almacena los 16 resultados del bloque Multiplica. A la salida se obtiene una señal de 16 bits en complemento a dos, permitiendo realizar un nuevo análisis en el próximo ciclo.

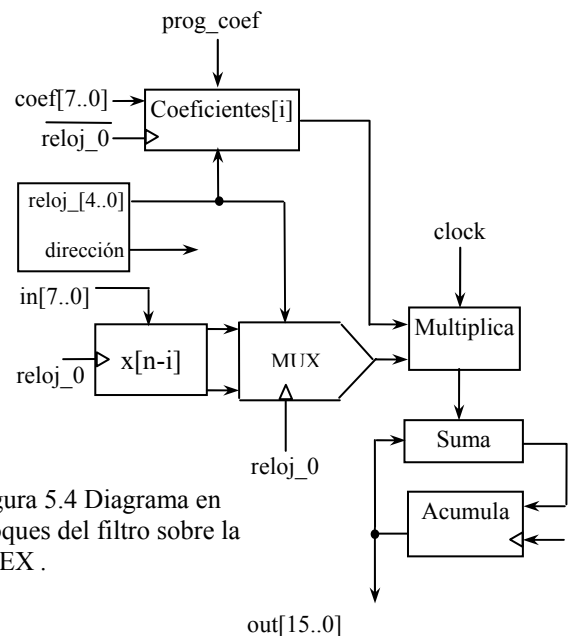
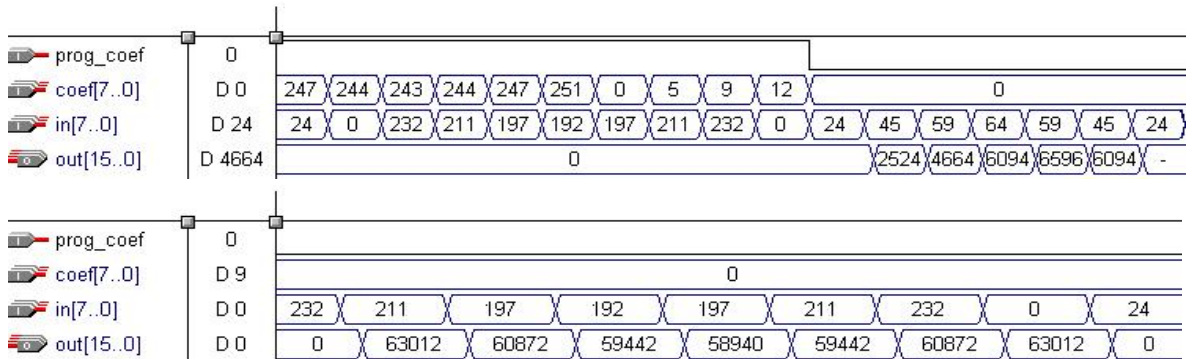


Figura 5.4 Diagrama en bloques del filtro sobre la FLEX.

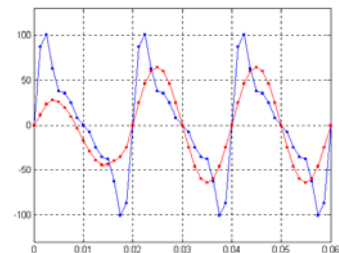
Del análisis en MATLAB y comparándolos con los valores de la salida obtenidos de la simulación sobre MAXPLUS, se obtiene:



El gráfico de la figura 5.5 representa la programación de los coeficientes (prog\_coef=1).

En la figura 5.6 se dibuja la respuesta en tiempo, donde nuevamente se observa que la señal de entrada posee ruido (línea de color azul) y a la salida se obtiene la respuesta sinusoidal de 50 Hz, excepto el 1º periodo que es el primer análisis donde.

Figura 5.6  
Respuesta en el



En este caso, se obtiene una mejor apreciación de los valores, entre los valores calculados por MATLAB y la simulación, debido a la mejor resolución de los valores de la entrada (8 bits).

Cuando se comparan los resultados entre la simulación con MATLAB y los valores obtenidos se observa total coincidencia, por ultimo se presenta el reporte final de MAX+PLUS II.

Device	In Pins	Out Pins	Bidir Pins	Memory Bits	%Utilized	LCs	%Utilized
EPF10K10LC84-3	18	16	0	128	2 %	526	91 %
User Pins:	18	16	0				

## CONCLUSIONES

Del estudio de las señales provenientes de sistemas eléctricos a través de técnicas digitales modernas como el filtrado, nos permite poner de manifiesto las alteraciones que ellas presentan, y a través del análisis cuantitativo correspondiente obtener conclusiones que

posibiliten determinar características del sistema y/o elemento objeto de ensayo, con muy buen grado de detalle.

En nuestro caso se han podido verificar dichas características, a través del desarrollo de distintos filtros en un dispositivo de lógica programable, de esta manera se obtiene la información experimental y la correspondiente descripción del hardware (HDL), necesarias para la implementación final de un dispositivo ASIC que cumpla con los requerimientos propuestos.

## **REFERENCIAS**

PASCUAL, H.O.; RAPALLINI, J.A.; QUIJANO, A.A. Implementación de un sistema de medida de impedancia para redes eléctricas en tiempo real. In: **VII Workshop IBERCHIP IWS'2001**, Montevideo, Uruguay, Marzo de 2001.

**Transductores de corriente, su interacción con las protecciones en sistemas eléctricos de potencia.** UNLP. Fac. Ing. , 2002. Tesis doctoral

GASTALDI, G.G. **Diseño e implementación de un equipo para la medición de impedancia en tiempo real.** Trabajo Final Electrónica. 15/03/02.

GASTALDI, G.G.; RAPALLINI, J.A.; PASCUAL, H.O. Evaluación de programas de cálculo en ingeniería como herramienta de desarrollo para Codiseño Hardware/Software, IWS2002 In: **VIII Workshop IBERCHIP**, Guadalajara, México, Abril 2002.

GASTALDI, Guillermo G.; RAPALLINI, José A.; QUIJANO, Antonio A. Diseño de un Sistema Digital de Medición de Impedancia a Tiempo Real Utilizando la Metodología de Codiseño Hardware/Software, IWS2003 In: **IX Workshop IBERCHIP**, La Habana, Cuba, Abril 2003.

PROAKIS, J.G. ; MANOLAKIS, D.G. **Tratamiento Digital de Señales:** Principios, algoritmos y aplicaciones. 3. Ed. Madrid: Prentice Hall, 1998.