



**UNIVERSIDAD DE CIENCIAS Y HUMANIDADES
DIRECCIÓN DE INVESTIGACIÓN**

**DISEÑO Y SIMULACIÓN DE FILTRO IIR
BUTTERWORTH SOBRE FPGA**

Diseño y simulación de filtro IIR Butterworth sobre FPGA

Ricardo Alonso Muñoz Canales

Asesor: Carlos Andrés Mugruza Vassallo

Facultad de Ciencias e Ingeniería, Universidad de Ciencias y Humanidades

ramc2402@gmail.com

Resumen—En el Perú los filtros digitales se emplean mayormente sobre sistemas de cómputo y su aplicación sobre software de simulación de alto nivel y escasamente sobre hardware como son el micro-controlador o FPGA.

Se presenta en este artículo el diseño un filtro rechaza banda entre las frecuencias de 330 y 400 Hz, el cual emplea 20 coeficientes, inviabilizando su implementación en un microcontrolador para las operaciones próximas a tiempo real, por ella se diseñan otras 2 etapas de procesos digitales en FPGA con un módulo de comunicación serial USART basado en el diseño de este último en máquinas de once estados.

Palabras claves: Field programable Gate Array (FPGA), Infinite Impulse Response (IIR), Universal Asynchronous Receiver-Transmitter(UART), PIC, Procesamiento digital de señales(PDS).

I. INTRODUCCIÓN

LA empresa Naval Industrias realiza servicios de mantenimiento preventivo y correctivo a sistemas electrónicos de comunicaciones marítimas, así como el diseño de prototipos de control basado en microcontroladores, para aplicaciones en barcos como son, paneles de control, girocompases analógicos y digitales, búferes de señales de protocolo *National Marine Electronics Association* (NMEA0183), control de motores azimutales, etc.

La presente investigación desarrolla el diseño de un prototipo de Filtro Digital de Señales, sobre un *Field programable Gate Array* (FPGA), para atenuar el ruido proveniente de los girocompases analógicos, balastros, a través de la línea de alimentación, que se conectan a diversos equipos de telecomunicaciones con alta sensibilidad.

El diseño se ha dividido en tres etapas, las cuales son:

- Módulo de Procesos de Filtros Digitales.
- Adquisición y digitalización de señales analógicas
- Módulo de Comunicación Serial.

El Procesamiento Digital de Señales (PDS) es un campo de estudio importante en el desarrollo tecnológico actual, sus aplicaciones abarcan no solo el campo de la ingeniería sino también campos diversos como la educación, medicina, radio definido por software, etc.

El PDS tiene entre sus elementos básicos el desarrollo de los filtros digitales como herramienta para acentuar, discriminar o eliminar ciertas señales que forman parte de la señal de datos original. Podemos encontrarlas de cuatro tipos

pasa bajo, pasa alto, pasa banda y rechaza banda.

Por ejemplo, recientemente en el 2014 en un congreso en Colombia, Edilberto Vivas y colaboradores [1] implementaron y simularon un filtro IIR de cuarto orden empleando el generador de sistemas de Matlab para FPGA.

En cambio en el presente trabajo se tiene como objetivo diseñar un filtro que permita rechazar la banda de 330 a 400 Hz de un girocompás analógico, elemento utilizados en muchos navíos, además de mantener la sensibilidad de un equipo de comunicación, lo cual indica que el orden de filtro es mayor, que para los requisitos del usuario, según se ve en este artículo es de orden 20.

II. DISEÑO DEL SISTEMA

El Desarrollo de la aplicación del proyecto está detallado en las etapas que se muestran en la Figura 1, donde se especifican puntos importantes a desarrollar dentro de ellas.

Sin embargo el orden lógico del desarrollo de diseño del Filtro FIR en VHDL es cómo sigue:

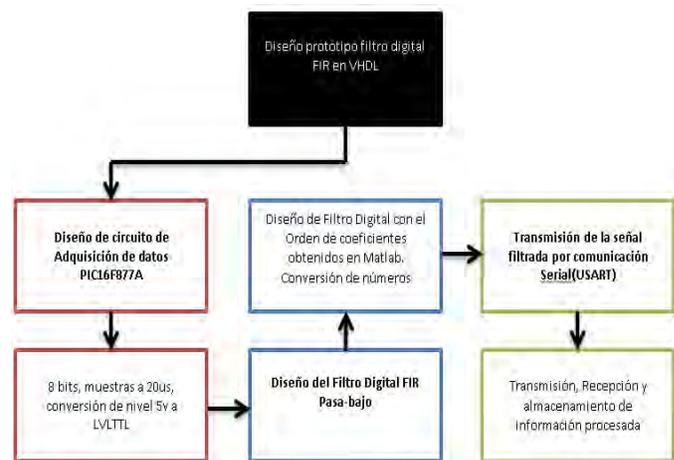


Fig. 1. Flujo de Diseño del prototipo sobre FPGA. Primera etapa (rojo): se diseña el circuito adaptador de voltaje y de conversión analógica digital con el PIC 16F877A. Segunda etapa (Azul), diseño de Filtro Digital con Matlab y posteriormente en VHDL[3]. Tercera Etapa (verde), Comunicación serial USART para el control y envío de datos procesados.

Fuente: Elaboración propia

El trabajo se ha dividido en tres partes:

- 1) Diseño de la adquisición de datos: Compuesto por el uso del microcontrolador PIC16F877A para la adquisición de señales analógicas y su conversión a digital por el puerto

paralelo hacia el FPGA, ya que la tarjeta de desarrollo no cuenta con un módulo de conversión A/D.

2) Diseño de Filtros Digitales:

Una vez seleccionadas las fuentes de ruidos a filtrar, se procede al diseño al análisis y simulación con Matlab de la respuesta de Filtros Digitales Pasa Alto, obteniendo de ello los coeficientes de la función de transferencia que describe al filtro digital

Una vez obtenidos dichos coeficientes se procede a realizar el diseño de la estructura digital del filtro en VHDL, para verificar y comprobar su funcionamiento en el FPGA.

3) Diseño de la etapa de comunicaciones

En esta etapa se realiza el diseño en VHDL del circuito de comunicación USART, el cual enviará los datos originales y procesados a través de un módulo USB-TTL o Bluetooth.

Finalmente se realiza el diseño de la interfaz de comunicación en lenguaje C, entre el PC y el FPGA, para el control y adquisición de datos.

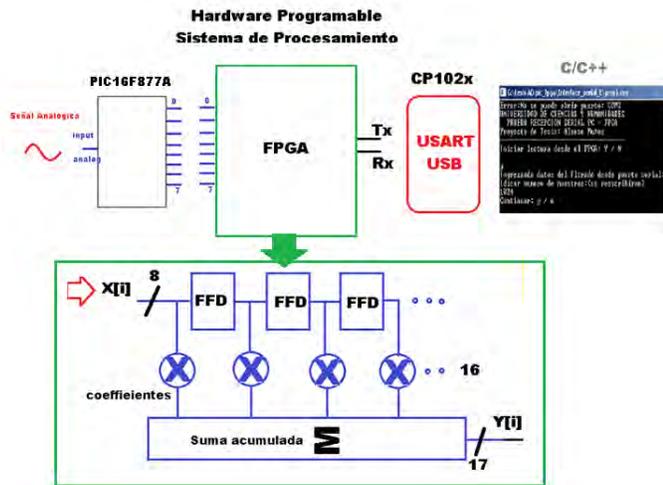


Fig. 2. Esquema de Diseño en bloques del prototipo: Adquisición de señal, filtrado de señal de ingreso, comunicación Serial USART para transmisión de datos y control del dispositivo.

Fuente: Elaboración propia

III. ADQUISICIÓN DE DATOS

En esta etapa se realiza la conversión analógica digital con el módulo de conversión A/D del microcontrolador PIC16F877A, y su posterior envío al FPGA de los datos bit a bit en forma paralela.

En esta etapa se realiza la conversión analógica digital con el módulo de conversión A/D del microcontrolador PIC16F877A, y su posterior envío al FPGA de los datos bit a bit en forma paralela.

En la figura 3 se muestra el diseño de la etapa de adquisición de señal basado en el uso del micro-controlador 16F877A y el circuito para acondicionar la señal de entrada de alimentación, adicionalmente el diseño realizado en Proteus

Presentado en el XXII Congreso Internacional de Ingeniería Eléctrica, Electrónica, Telecomunicaciones y Computación – INTERCON 2015, Ciudad de Huancayo del 03 al 07 de agosto.

VSM, se muestra solo para fines de prueba, la lectura de datos digitales obtenidos en el display LCD y el Virtual Terminal.

El circuito de acondicionamiento de señal está conformado por un transformador reductor de voltaje y un OPAM TL062 con diodos de protección de voltaje inverso y regulador de tensión a 5v.

Ya que la interconexión con el FPGA se realizará mediante el uso del puerto paralelo PORT D del micro-controlador, el cual está diseñado por el fabricante para dicho fin de transmisión.

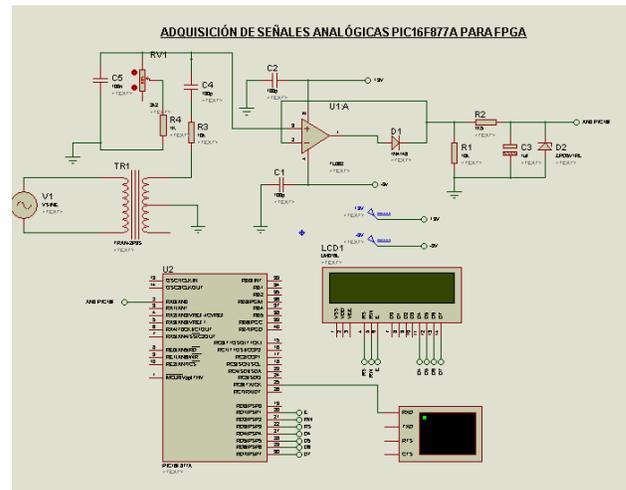


Fig. 3. Adquisición de señales analógicas, compuesta del circuito para adaptar el voltaje de alimentación al convertor AD del PIC 16F877A.

Fuente: Elaboración propia

IV. DISEÑO DE FILTRO DIGITAL

Para el presente proyecto se ha propuesto el diseño de un filtro digital tipo IIR.

El filtro IIR tiene la siguiente expresión matemática:

$$y_n = b_0x_n + b_1x_{n-1} + b_2x_{n-2} + \dots + b_Nx_{n-N} + a_1y_{n-1} - a_2y_{n-2} - \dots - a_My_{n-M}$$

Donde a y b son los coeficientes del filtro y M y N determinan los polos y ceros.

Haciendo uso de la transformada Z, obtenemos la función de transferencia del filtro:

$$H(z) = \frac{\sum_{i=0}^P b_i z^{-i}}{1 + \sum_{j=1}^M a_j z^{-j}}$$

En específico para el presente diseño se realizará el diseño del filtro IIR tipo AR (Autoregresivo), el cual se describe mediante la siguiente ecuación de transferencia:

$$H(z) = \frac{1}{1 + A_1 z^{-1} + A_2 z^{-2} + \dots + A_N z^{-N}}$$

La selección se justifica debido a que contiene solo polos y en la respuesta prácticamente no tiene rizados.

Finalmente se utilizará el método por aproximación de Butterworth para el diseño de filtro IIR debido a que presenta un mínimo rizado en las bandas próximas a la frecuencia de corte.

Para la obtención de la Función de transferencia del Filtro Digital IIR se han utilizado los siguientes parámetros para su diseño y simulación en Matlab:

Con ganancias mínimas y máximas para el rizado y rechazo de banda establecidas por el usuario conforme a la sensibilidad de los quipos afectados.

La banda de frecuencia a filtrar está determinada por la frecuencia de oscilación de las bobinas que conforman el cuerpo del girocompás, que genera el campo magnético para la ubicación del Norte, el rango de frecuencia se encuentra entre 300 a 400 MHz.

Para la prueba se han establecidos los siguientes parámetros:

Amin=40dB; (Ganancia banda de rechazo)
 Rs=0.1dB; (Ganancia rizado stop)
 Rp=0.1dB; (Ganancia rizado paso)
 Wp(1)=2*pi*300; (Frecuencia de corte inferior)
 Ws(1)=2*pi*330; (Frecuencia de paso inferior)
 Ws(2)=2*pi*400; (Frecuencia de corte superior)
 Wp(2)=2*pi*440; (Frecuencia de paso superior)

El resultado de simulación arroja un filtro de Butterworth de 10 polos en una función de transferencia de orden 20.

Se debe tener en cuenta que según la aplicación que se requiera implementar, es posible configurar la respuesta del filtro en base a los parámetros de respuesta como ganancia de rizo, ganancia en corte, tipo de filtro, frecuencias, número de bits para la representación de señal y coeficientes del filtro, etc. Así podremos obtener de diversas formas la cantidad de coeficientes o configurar manualmente los coeficientes deseados del filtro, donde en este último caso se verá cómo se afecta la respuesta diferente a la configurada en base a los parámetros.

Una vez obtenido los coeficientes se procede a su conversión en números binarios, para luego ser usados en el diseño en VHDL del circuito digital del filtro obtenido en Matlab.

La base del diseño del circuito está sustentada en el diagrama de bloques de Filtros Digitales Tipo FIR, en su forma Transpuesta, ya que esta configuración a diferencia de la Directa es más óptima de ser implementada computacionalmente.

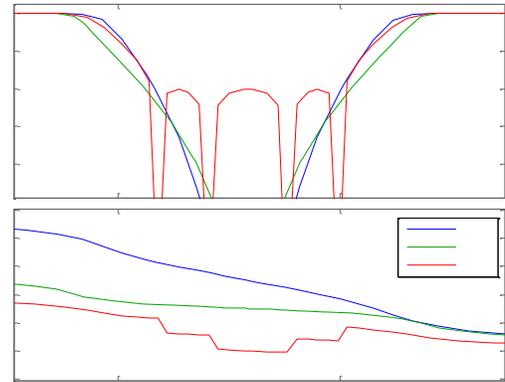


Fig. 4. Respuesta en Magnitud y Fase (respectivamente) obtenidos mediante la simulación en Matlab del diseño del filtro rechaza banda IIR para efectos de comparación; el Butterworth es la señal roja, Chebyshev señal verde y el Eliptico la señal azul.

Fuente: Elaboración propia

La etapa de diseño de filtro digital conlleva tres consideraciones importantes:

- Resolución (bits) de la señal muestreada.
- Número de bits para la representación de coeficientes.
- Circuito multiplicador-acumulador encargado de las operaciones que conforman la estructura del circuito digital.

La primera etapa está determinada por la resolución del convertor analógico digital utilizado, por tanto en este artículo se propone trabajar con 8 bits de resolución ya que corresponde a la del ADC del PIC16F877A, sin embargo esta limitación en número de bits reducirá la respuesta sin perder precisión al representar los números en la conversión analógica digital, ni los del punto flotante que nos proporciona Matlab como coeficientes.

Sin embargo es factible de utilizar otro convertor de mejores prestaciones como el AD7991 o el AD7476A, sin embargo la capacidad de datos a ser transferidos requieren el uso de un protocolo de comunicación serial de mayor velocidad al USART, como son el SPI y el I2C, esto provoca mayor tiempo en el diseño y desarrollo en VHDL de dichos circuitos encargados para la comunicación.

Se recomienda expresar la estructura del filtro digital en una forma transpuesta, siendo que esta es más óptima de ser realizada computacionalmente como se observa en la figura 6.

Otra etapa importante a considerar es el formato de numeración a utilizar para la representación de los coeficientes y datos almacenados para el proceso de filtrado, se recomienda utilizar el standard IEEE 754[6], el cual define el formato de representación de números en punto flotante, el cual está implementado dentro de la estructura digital de la mayoría de microprocesadores con capacidades de procesamiento de datos, sin embargo el presente diseño utiliza una conversión de los coeficientes a complemento a dos.

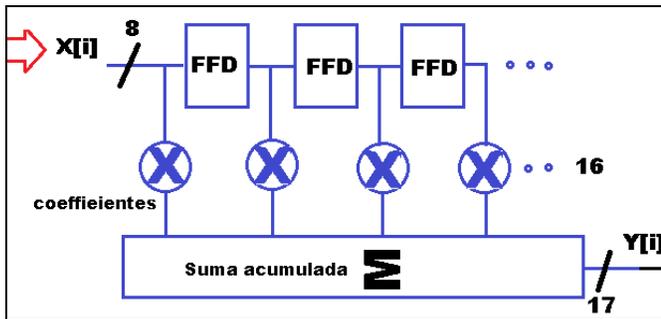


Fig. 5. Estructura del Filtro Digital a diseñarse en VHDL, donde "x[i]" representa la señales de entrada, FFD(Flip Flops D) representan los retardo de z,"X" los productos de coeficientes, la sumatoria indica la suma acumulada de los resultados, una cantidad de veces equivalente al orden del filtro.

Fuente: Elaboración propia

V. COMUNICACIÓN SERIAL

La comunicación serial USART se diseña en base al diagrama de flujo de la figura 7, para el envío de datos seriales a través del USB TTL y del módulo de comunicación Bluetooth.

En un inicio tenemos el 'clear' como señal asíncrona que reinicia el estado de recepción o envío de datos, a continuación la verificación del flag 'ready', el cual indica si la línea de comunicación está libre y lista para recibir o enviar datos, en base al envío de datos seriales empezando con el bit menos significativo de un registro de datos de 8 bits se procede con la secuencia de envío bit a bit, en intervalos de tiempo determinados en base a la velocidad de datos establecida en un inicio, para la simulación se utilizaron por defecto los siguientes parámetros:

- Velocidad: 9600 baudios
- Bits: 8 bits de datos
- Paridad: ningún bit de paridad
- Stop: 1 bit de stop

Este diagrama permite identificar la etapas necesarias para el diseño del módulo de comunicación serial USART en VHDL, es por tanto que se utiliza el diseño en base a máquinas de estado en este caso Mealy.

En la figura 6 también la variable 'Tick' hace referencia al tiempo de envío por cada bit de datos, siendo este tiempo un parámetro obtenido de obtener el tiempo de envío de cada bit en relación a la velocidad en baudios de la comunicación

$$Vel(\text{baudios}) = 9600\text{baudios}$$

$$tick = \frac{1\text{segundo}}{2 * Vel(\text{baudios})}$$

$$tick \approx 52.08\mu s$$

Donde la constante dos es una referencia para utilizar los semiperiodos de los tick's y así asegurar el tiempo de inicio y final de envío de cada bit de la trama de comunicación.

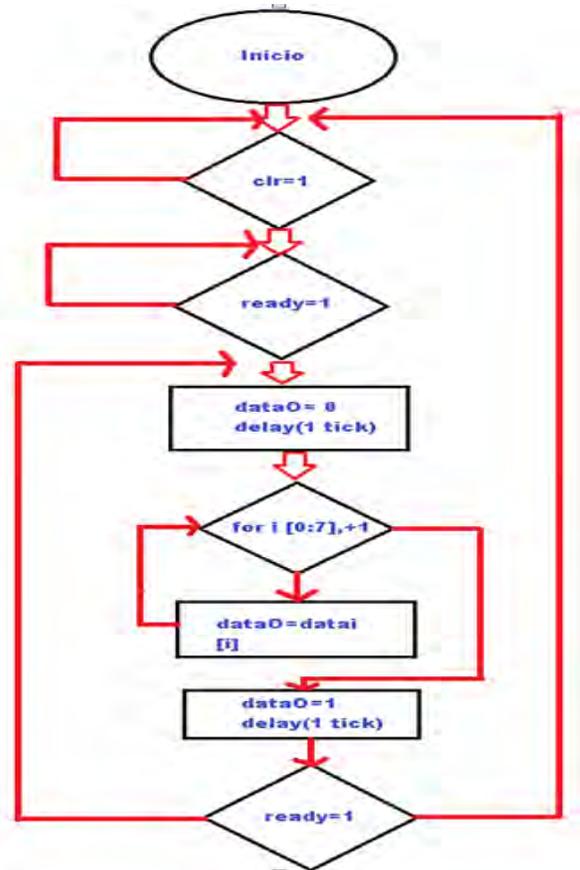


Fig. 6. Diagrama de flujo para el envío de datos por comunicación serial USART, para 8 bits de datos a una velocidad de 9600baudios
Fuente: Elaboración propia

En la figura 6, podemos observar el desarrollo de la máquina de estados tipo Mealy, para el diseño del módulo de comunicación serial USART sobre el FPGA en lenguaje VHDL, para ello se utilizó el diseño en base a dos procesos [4].

Se observa que la máquina empieza en un estado de espera verificando si el flag de 'ready' y la señal de 'clr' permiten el acceso al siguiente estado llamado inicio, el cual está conformado por análisis al estado lógico de la línea de comunicación el cual debe estar en alto '1', enseguida se realiza el envío bit a bit del dato a ser transmitido realizando cada envío de bit en un estado, finalmente llegado al estado de stop se reinicia el envío de datos seriales desde el estado de espera.

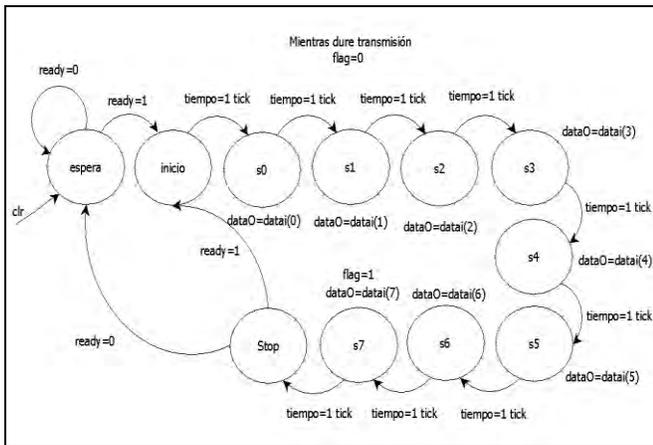


Fig. 7. Máquina de estados para el diseño en VHDL del módulo de comunicación Serial USART, basado en una máquina de estado tipo Mealy, con señal de entrada 'data', tiempo entre estados 'tick' (9600bauds) y una señal asíncrona 'clear'.
Fuente: Elaboración propia

VI. INTERFAZ DE CONTROL

En esta etapa se realiza el diseño del software para la computadora, que hará la interfaz de control y recepción de datos desde el FPGA por intermedio de la virtualización de un puerto serial COM a través del puerto USB.

VII. RESULTADOS

En la respuesta de trenes de onda podemos observar en la figura 9, el envío de bits de la secuencia Datai = "01010101" entre los flancos del Flag de aviso de señal de envío y cada bit enviado conformado por dos Tick's con la señal en Ready='1', produciendo el envío de la data en serie a través de Datao.

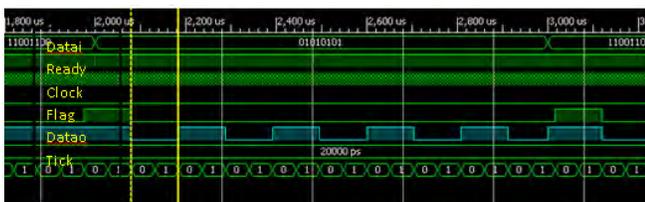


Fig. 8. Simulación de la señal enviada por medio del módulo de comunicación serial USART a 9600 baudios. En amarillo se indica los nombres de los distintos trenes de onda.
Fuente: Elaboración propia

Para la adquisición de datos se realizó la prueba de comunicación entre el FPGA y el PC enviando números constantes para verificar el envío y recepción de datos, así como el control de operación del FPGA. Se utilizó el IDE DevC++, y la librería para comunicación serial LnxCom,

En una primera etapa este interfaz realiza el control de recepción de datos indicando el número de datos que se desean leer.

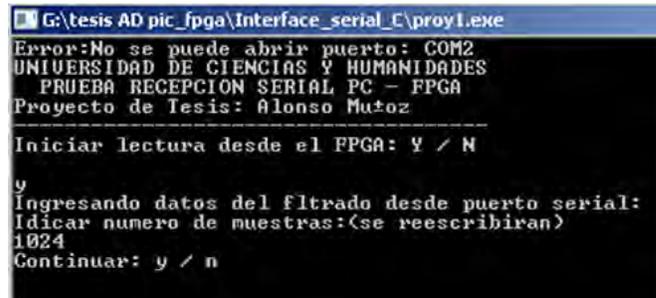


Fig. 9. Interfaz de control y adquisición de datos, realizado en Dev C++, para la comunicación serial USART con el dispositivo FPGA en consola de comando. Fuente: Elaboración propia

VIII. CONCLUSIONES

En este diseño se ha propuesto la adquisición de datos con un microcontrolador PIC16F877A, del cual se obtiene la señal digitalizada para ser enviada al FPGA por intermedio del puerto paralelo del PIC, así mismo por medio del uso de Matlab y sus funciones y scripts se ha realizado el diseño y simulación del filtro IIR Butterworth Rechaza Banda, obteniéndose así los coeficientes a ser utilizados en el diseño en VHDL para el circuito digital en el FPGA que realizará el proceso de filtrado, a continuación se diseñó el módulo de comunicación serial USART en VHDL para el envío de datos procesados hacia el computador, y finalmente se realizó la interfaz de control y adquisición en el computador en lenguaje C.

El diseño y simulación del filtro IIR Butterworth produce para las especificaciones de respuesta requerida nos proporcionan un filtro IIR de orden 20, donde se aprovecha el uso del Hardware programable FPGA.

Una desventaja de la tarjeta de desarrollo usada para el diseño, es que no incluye en su circuito un convertor analógico digital, utilizándose por ello uno externo como es el caso del que tiene internamente el PIC16F877A el cual es solo para la adquisición de señal.

REFERENCIAS

- [1] VIVAS GONZALEZ, Edilberto Carlos; RIVERA PINZON, Diego Mauricio; GOMEZ, Edwar Jacinto. Implementation and simulation of IIR digital filters in FPGA using MatLab system generator. En Circuits and Systems (CWCAS), 2014 IEEE 5th Colombian Workshop on. IEEE, 2014. p. 1-5.
- [2] V.A. Pedroni, *Circuit Design with VHDL*, Cambridge, 2004, pp. 275-301.
- [3] IEEE Standard Multivalued Logic System for VHDL Model Interoperability, IEEE Standard 1164, 1993.
- [4] Pardo Carpio, *VHDL: Lenguaje para descripción y modelado de circuito*, 1994, Valencia: Universidad de Valencia.
- [5] Peter R. Wilson, *Design Recipes of FPGAs*, Oxford: Elsevier, 2007, P.98-109.
IEEE Standard for Floating-Point Arithmetic, IEEE Standard 754, 2008.