RIELAC, Vol.XXXIV 2/2013 p.63-75 Mayo - Agosto ISSN: 1815-5928

RIELAC
Revista de Ingeniería Electrónica, Automática y Comunicaciones

Implementación en VHDL de un Detector de Envolvente para demodulación BFSK

Karel Toledo de la Garza, Jorge Torres Gómez, Juan R. Rodríguez Suárez

RESUMEN / *ABSTRACT*

El presente artículo aborda el empleo de un bloque Detector de Envolvente para demodular señales BFSK que pueda ser usado en aplicaciones donde se desconoce el tiempo del símbolo de la fuente. Presenta una estructura interna caracterizada por cuatro filtrosdel tipo FIR, que son inherentemente estables y se implementan siempre por una misma ecuación de diferenciasgenérica.

El demodulador se configura en lenguaje VHDL con un número variable de coeficientes no especificado de antemano y está sintetizado como un módulo IP con el que se buscareconfigurabilidad. Para validar el demodulador,se implementa en un circuito FPGA de Xilinx un procesador Microblaze que se comunicacon una PC mediante el puerto serie y se configura con diversos periféricos, tales como la interfaz de comunicación serieRS-232 y el módulo IP del demodulador BFSK especialmente diseñado al efecto. Para gestionar la operación del sistema se desarrolló en la PC un programa en Matlabcon una aplicación gráfica de usuario que incluye el envío y recepción de las señales moduladas y demoduladas por el circuito FPGA, así como el envío de los valores de los coeficientes empleados por los filtros FIR en una determinada aplicación.

La solución final permite la demodulación de señales BFSK a través de la interconexión de Matlab con Microblaze y de este con el módulo IP. Se presenta en detalle el modelo VHDL del demodulador, se discuten los resultados alcanzados teniendo en cuenta el efecto de la cuantificación de los coeficientes y se realiza un análisis temporal y de ocupación del circuito FPGA.

Palabras claves: BFSK, Detector de Envolvente, FPGA.

Envelope Detector Development for BFSK signals in VHDL.

This paper concerns demodulator-based Envelope Detector for recovery information in BFSK signal applied upon applications where time symbol synchronization is unknown. Its structure is characterized by 4 filters which are considered in the present article by FIR, this are unconditionally stables and are always implemented by the same generic difference equation.

In the present work this demodulator is configured in VHDL language with a variable number of coefficients not specified in advance and conformed it in an IP module we obtain a reconfigurable circuit. In order to validate the system an FPGA from Xilinxwith PC serial communication is implemented. FPGA design comprises a Microblaze microcontroller with an Envelope Detector IP module and aRS-232 IP module for communication.A Matlab user application on PC to manage demodulation process is also implemented, it takes into account not also to send and receive the modulated and demodulated signal, but to send the coefficients values employed by the FIR filters in a specific application.

The final solution allows demodulation of BFSK signals through the interconnection of Matlab with Microblaze and this with the IP module in Xilinx environment.Details of VHDL design and its results are discussed taking into account the effect of coefficient quantization, temporal analysis and FPGA occupancy.

Key words: BFSK, Envelope Detector, FPGA.

INTRODUCCIÓN

La modulación digital es el proceso mediante el cual se incorpora la información que poseen determinados símbolos digitales en formas de onda compatibles con las características del canal;lo quese logra variando la fase, amplitud o frecuencia de una señal denominada portadora de acuerdo a determinada ley. Este proceso se lleva a cabo en el bloque transmisor, que es el encargado de acoplar la señal al canal de comunicaciones para combatir efectos indeseables tales como la distorsión, el ruido, la atenuación y la interferencia. El receptor debe realizar el proceso inverso para así recuperar el mensaje.

Entre las modulaciones empleadas se encuentra BFSK (BinaryFrequencyShiftKeying, Modulación Digital de dos Frecuencias) en la cual, la información va contenida en la frecuencia instantánea de la portadora, que cambia de acuerdo a dos símbolos digitales: 0 y 1;los cuales se asocian cada uno con frecuencias distintas, quedando la señal modulada como se muestra en la ecuación (1):

$$S_i(t) = \sqrt{\frac{2E}{T}}\cos(w_i t + \Phi) \qquad 0 \le t \le T, i = 1,2,\dots,M \tag{1}$$

Donde E es la energía de la señal, T es el tiempo de símbolo, w_i es la frecuencia angular y Φ es la constante de fase.

A la portadora se le asignan dos frecuencias diferentes, cada una destinada a identificar un símbolo durante el tiempo que dure este en la fuente. En la figura 1 se muestra un ejemplo de la modulación BFSK; la gráfica superior muestra la información binaria y la inferior el tono modulado por la secuencia de la gráfica superiorde la figura 1. Los cambios de frecuencia son dos,uno para cada símbolo de información.

Las señales con las cuales se opera se encuentran almacenadas en formato para archivos de sonido tipo Microsoft WAVE (extensión .wav), de las cuales no se conoce el tiempo de símbolo y esto conlleva a utilizar métodos que prescindan de este parámetro para demodular.

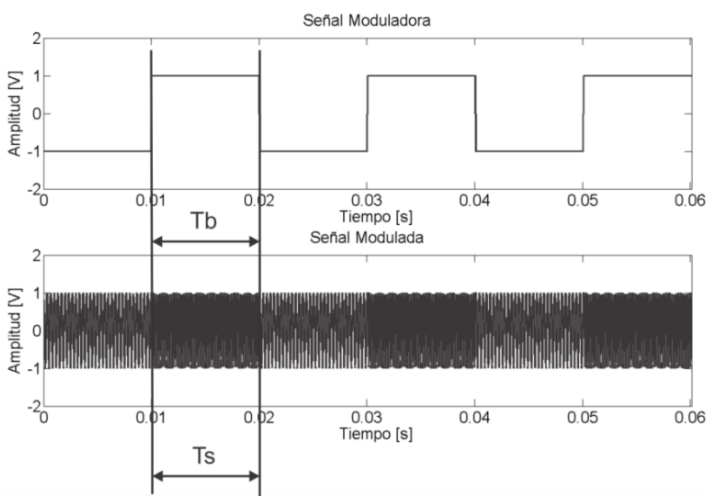


Figura 1. Señal moduladora y modulada BFSK.

De los métodos consultados en la literatura científica, el Detector de Correlación[1]es el que presenta mejores resultados de relación señal a ruido, pero para su funcionamiento se necesita contar con la sincronía del tiempo de símbolo[2 - 5]. Dentro de los demoduladores que no emplean el tiempo de símbolo, se encuentra el Demodulador basado en Autosincronía[6, 7], el cual es muy complejo de implementar producto de sus parámetros dinámicos; presenta, además,un alto consumo de recursos en hardware debido al bloque de tangente inversa y al oscilador controlado por voltaje que es necesario implementar.Por otra parte, el Detector Diferencial[8]es de fácil implementación y consume pocos recursos; pero tiene la desventaja de que es muy complejo lograr un retardador con valor dependiente de la frecuencia de trabajo. Entre los demoduladores con filtrado adaptativo, se encuentran los que almacenan una señal de referencia[9 - 11]para la demodulación y los que recuperan la frecuencia instantánea de la señal recibida[12 - 17]; ambos requieren pocos recursos de cómputo en su implementación. No obstante los primeros resultan muy vulnerables a los efectos indeseables del canal, dado que su convergencia es dependiente del carácter estacionario de la señal recibida; los segundos convergen de forma lenta cuando la separación de frecuencia supera 1 kHz.

El demodulador basado en el Detector de Envolvente[1]fue escogido por su fácil implementación en lenguaje VHDL debido a la simetría de sus elementos. Con solo implementar un filtro FIR, con coeficientes declarados como genéricos, diseñado y replicado de forma convenientepara que se comporte como pasa-banda o pasa-bajo, se puede sintetizar el demodulador en lo fundamental. El demodulador tiene como ventaja el bajo consumo de recursos en hardware y el empleode la misma cantidad de multiplicadores que el detector que utiliza el filtro adaptativo Notch, y tiene comolimitación que se necesitan filtros de un orden elevado para separar las frecuencias cuando la separación entre las mismas es pequeña; cuestión esta última que hará consumir más recursos y más tiempo de procesamiento. Se pueden aplicar variantes para reducir su costo computacional mediante el empleo de la modulaciónDelta-Sigma[18].

En la figura 2 se presenta el diagrama en bloques del demodulador analizado. En él se empleandos filtros pasa-banda sintonizados a las frecuencias de transmisión f_1 y f_2, dos detectores de envolvente a la salida de los filtros pasa-banda con etapas de rectificación y filtrado pasa-bajo y por último una etapa de decisión, que se encarga de determinar el símbolo transmitido al calcular la mayor envolvente. Este demodulador no necesita del conocimiento previo del tiempo de símbolo,pues se basa en medir los niveles de la amplitud yno tiene en cuenta la forma del espectro en la frecuencia, por lo que no considera la forma de onda que transcurre en el canal. Esta solución permite su empleo como demodulador de señales BFSK en tiempo real, siempre que el tiempo de procesamiento por cada muestra sea menor que el tiempo de muestreo del sistema.

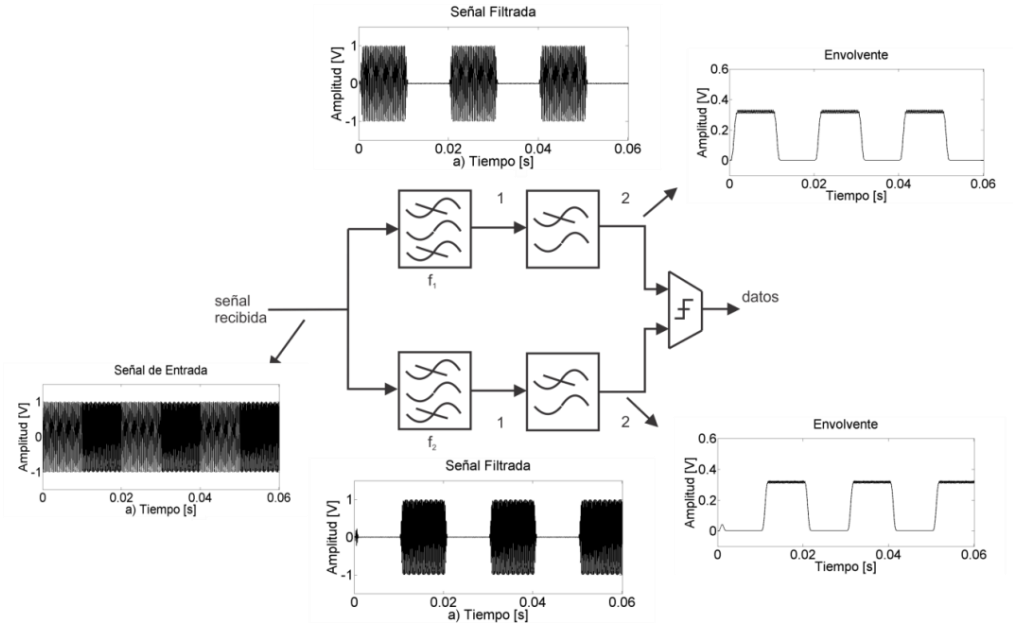


Figura 2. Detector de Envolvente.

CONFORMACIÓN DEL SISTEMA

El demodulador de la figura 2 es muy sugerente para su implementación hardware, con vista a emplearse en un sistema portable que funcione en tiempo real utilizando FPGA[19, 20] (Field ProgrammableGateArray, Arreglo de Compuertas Programables).La propuesta de sistema consiste en emplear una PC que almacena las señales moduladas y recibe las señales demoduladas por una tarjeta Xilinx Starter Kit con circuitoFPGA Spartan 3E. Ambos bloques se comunican entre sí mediante una interfaz serie RS-232. En la PC se desarrolló un programa en Matlab, con una interfaz gráfica de usuario que permite controlar la operación del sistema, que se muestra en la figura 3. La aplicación de Matlab está formada por cuatro gráficas, donde se muestran respectivamente: la señal BFSK a demodular, su transformada de Fourier, las envolventes recuperadas y la demodulación. Se presenta además un panel de control en el área 5, donde se seleccionan las opciones de demodulación mediante las funciones propias del Matlab o mediante el empleo del circuito FPGA (enmarcado en rojo).

Cuando se activa la acción de efectuar la demodulación por medio del FPGA, el programa calcula de forma automática los coeficientes de los filtros a partir de la determinación de las frecuencias de trabajo f_1 y f_2. Para ellose aplica la transformada FFT a las señales moduladas y se determinan sus máximos[21, 22],que se muestran en la gráfica 2 de la figura 3. Los coeficientes de cada uno de los cuatro filtros, se calculan mediante el empleo del método de la Ventana de Kaiser, debido a las ventajas que presenta con respecto a otro tipo de ventanas[23]. Estos coeficientes y la señal a demodular presentada en la gráfica 1 se envían al circuito FPGA mediante la comunicación RS-232. A medida que el demodulador BFSK implementado en el circuito FPGA procesa las señales, se envían las envolventes de las mismas hacia la aplicación de Matlab mediante la comunicación serie y se muestran en la gráfica 3. Por último la etapa de decisión implementada en Matlab determina y muestra la demodulación final en la gráfica 4.

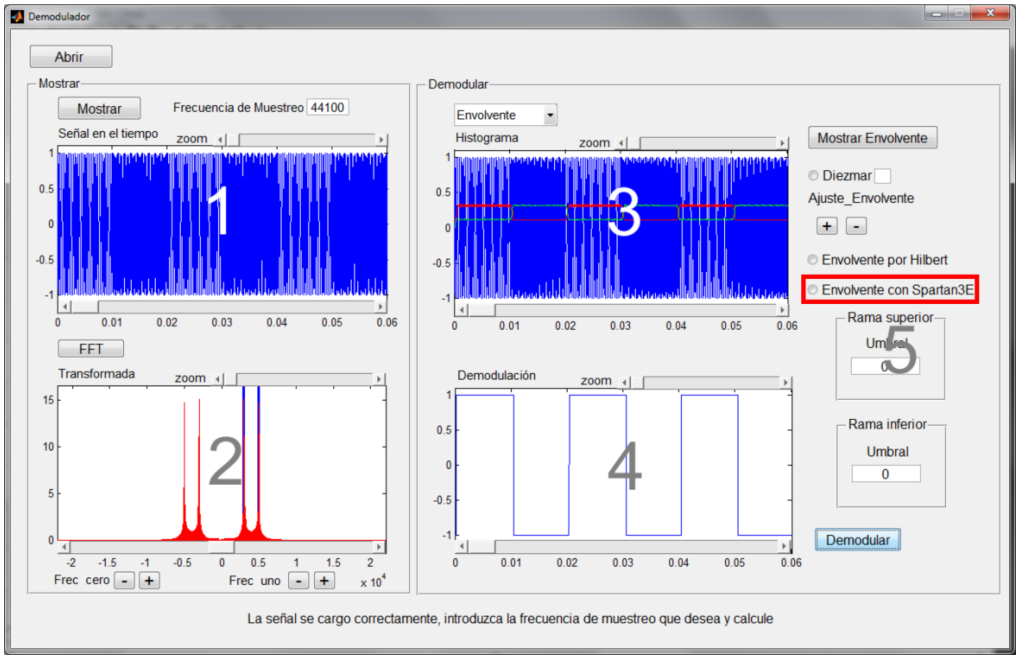


Figura 3. Interfaz gráfica para demodular señales BFSK con el Spartan-3E.

En el circuito FPGA se implementa un procesador MicroBlaze configurado con dos periféricos conectados al bus PLB: el controlador estándar de interfaz RS-232 xps_uartlite y el módulo IP del demodulador BFSK especialmente diseñado para estainvestigación. La figura 4 muestra el esquema general del circuito sintetizado en FPGA.

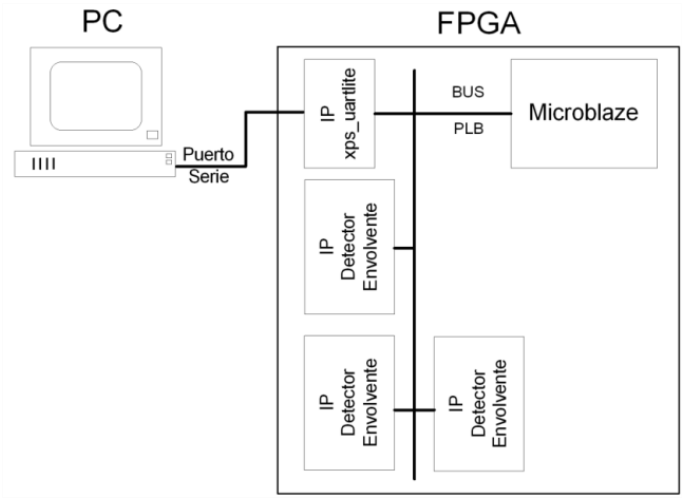


Figura 4. Esquema general del sistema.

El procesador Microblaze se configura con una memoria de datos y de programa de 16 kB, se le añade además un módulo de depuración mdm, sin especificar la unidad de punto flotante, ni emplear el bus FSL, ni la unidad de multiplicación de enteros. La interfaz RS-232 se programa con una velocidad de transferencia de 115200 bauds, con datos de 8 bits, sin empleo de bit de paridad y con una capacidad de 128 bits de su memoria FIFO de transmisión y recepción. Para este diseño se consume el 28% de los Slices del Spartan-3E y 3 de sus 20 multiplicadores de 18 bits[24].

El módulo IP que implementa el Detector de Envolvente necesita de la conformación de cuatro filtros FIR, estos definidos a partir de una ecuación en diferencias genérica dada por la ecuación (2)[25]:

$$y(n) = \sum_{i=0}^{N_{coef}-1} w_i \cdot x(n-i) \tag{2}$$

La respuesta del filtro $y(n)$, depende de la suma de productos de las entradas retardadas y los coeficientes w_i del filtro en cuestión. Si se implementa esta ecuación directamente, se necesitarán tantos multiplicadores como coeficientes tenga el filtro. Como el número de multiplicadores está limitado a 20 en el circuito Spartan-3E usado, se decidió emplear la estructura conocida como multiplicador y acumulador MAC[26]. Esta estructura emplea solamente un multiplicador por filtro FIR y posee dos bloques por división en tiempo (TDM) para manejar las señales retardadas y sus coeficientes, los cuales trabajan a una frecuencia de reloj mayor que la frecuencia de muestreo.

La declaración de laentidad para operar con el filtro FIR se refleja de forma esquemática en la figura 5 y en el Código 1se destaca el carácter genérico de la variable **`coeff_number`**, que es reutilizada en la señal **`coeficientes`** para reservar el espacio necesario según el total de sumandos a emplear en (2) y en la arquitectura para determinar el espacio de memoria de la señales de operación interna. Todas las operaciones se realizan con precisión con formato punto fijo con signo $s(16,15)$, teniendo en cuenta la capacidad del buffer del módulo xps_uartlite del FPGA; pudiendo esta característica ser modificada con el consiguiente análisis del total de multiplicadores a emplear.

La interfaz conformada permite la comunicación con el filtro de forma asíncrona.Una vez configurados todos los coeficientes, en la señal **`data_in`**se colocan sucesivamente las muestras a filtrar mientras que en **`ready_in`**se indica la validación.Por último la señal **`ready_out`**definida en la arquitectura del filtro indica el fin de procesamiento de cada muestra de la señal **`data_out`**.

En esta investigación no se ha empleado la herramienta para la generación de filtros CoreGenerator de Xilinx debido a que la misma implementa filtros con un número de coeficientes y valores determinados, los cualesse corresponden a unas especificaciones dadas de la respuesta de frecuencia. En este caso se requiere que los filtros tengan un número de coeficientes variable atendiendo a las exigencias de los valores de las frecuencias empleadas en la modulación BFSK. El diseño propuesto para el filtro FIR en VHDL tiene la característica de permitir variar dinámicamente tanto el valor de los coeficientes como su cantidad mediante el uso de una especificación de tipo genérico en la declaración de entidad.

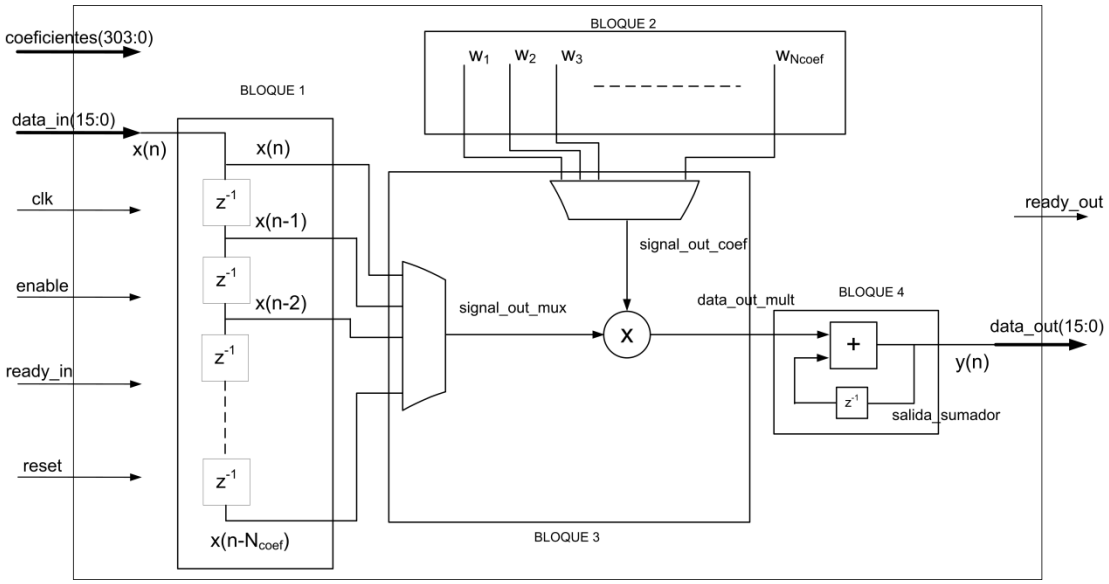


Figura 5. Esquema simplificado de filtro FIR.

```
entity filtro_fir is
generic (
          coeff_number : integer := 18);  -- total de coeficientes del filtro
port (
          clk: in  std_logic;  -- reloj de entrada
          reset: in  std_logic;  -- en nivel alto la salida es cero
          enable: in  std_logic; -- en nivel alto detiene la operación
          coeficientes: in std_logic_vector(coeff_number*16-1downto 0); --coeficientes
          ready_in: in std_logic;  -- en nivel alto indica un dato válido en la entradadata_in
          data_in:instd_logic_vector(16-1 downto 0);--señal de entrada a procesar
          ready_out:outstd_logic;--conunfrentedesubidaindicaquelaoperación
                                  -- se realizó con éxito
          data_out: outstd_logic_vector(16-1 downto 0));--señal de salida del filtro FIR
endfiltro_fir;
```

Código 1. Declaración de entidad para diseño del filtro FIR genérico.

La arquitectura para la entidad mostrada en el Código 1responde directamente a la ecuación (2), se compone de cuatro bloquesfundamentales mostrados en la figura 5 y su diseño se concibe para emplear la estructura MAC. La comunicación entre los bloques se verifica con las señales: **`signal_out_mux`**, **`coeficientes`** y**`signal_out_mult`,** definidas con dimensiones según el número total de coeficientes;utilizando en todos los casos el formato de 16 bits de datos. Las funciones de cada bloque se correspondencon:

- BLOQUE 1: Realiza la operación de muestreo de la señal de entrada y almacena los retardos necesarios. Se programa en VHDL sin estar especificado de antemano el total de retardadores. La figura 6 evidencia la operación de retardo en la señal**`signal_out_mux`** en función de la variable de recorrido i, con esto sedetermina el término i-ésimo de la sucesión de índices de la señal, dadopor el punto inicial 16(i-2) y el punto final 16(i-1)-1 con i comenzandoen 2 y terminando en N. El Código 2 emplea la instrucción secuencial **`for`**y la variable genérica **`coeff_number`** para implementar un proceso deretardo en la señal **`signal_out_mux`**, comenzando desde i=N decrementandohasta 2 con el objetivo de realizar los retardos apropiados sin sobreescribir la señal útil, esto en función del total de retardadoresindicado en la variable **`coeff_number`**. Este retardo está sincronizadosegún el frente de subida del reloj interno de$50\ MHz$.

- BLOQUE 2: Recibe y almacena el total de coeficientes del filtro que se especifican en la señal **signal_out_coef**, conformada de forma genérica con la instrucción:

```
signalcoeficientes :std_logic_vector(coeff_number*16-1 downto 0) := (others=>0);
```

- BLOQUE 3: Multiplica de forma serie cada coeficiente por la muestra retardada correspondiente y el resultado de esta operación se va sobre-escribiendo en la señal **data_out_mult**. El multiplicador se realiza con formato de 16 bits.
- BLOQUE 4: Realiza la acumulación de los sucesivos resultados colocados en la señal **data_out_mult**.

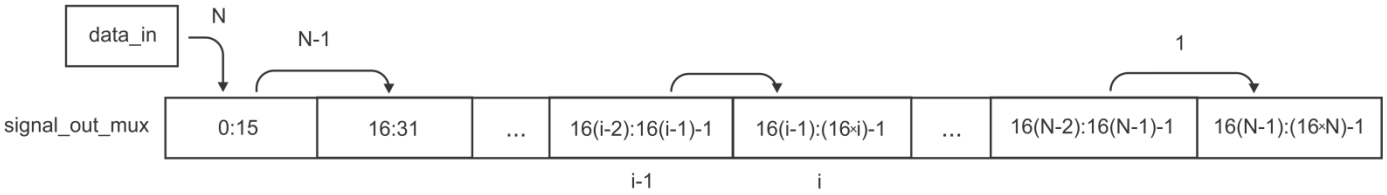


Figura 6. Esquema genérico para los retardos donde N es la cantidad de coeficientes.

```
if(clk'event and clk='1') and flag_signal='1' then
--retraso y muestreo
   for i in coeff_numberdownto 2 loop
      signal_out_mux(16*i-1 downto16*(i-1))<=signal_out(16*(i-1)-1 downto16*(i-2));
   end loop;  -- i
   signal_out_mux(bits_number-1 downto 0)<=data_in;
endif;
```

Código 2. Descripción del bloque de retardadores.

Cada bloque se programa como un componente y su operación se define de forma sincrónica mediante una máquina de estados tipo Moore, que controla apropiadamente las entradas del multiplicador y del acumulador para funcionar como se indica en la ecuación (2). Luego de estar almacenados todos los coeficientes, el orden de las operaciones por cada muestra recibida se define según los cuatro estados siguientes:

1. Estado **st1_wait_flag**: El sistema espera hasta la activación de una bandera que indique el frente de subida en la señal **ready_in**. Con esta activación se pasa al segundo estado.
2. Estado **st2_rdy_out_ret**: se habilita el BLOQUE 1, se realiza el retardo de las muestras anteriores y se muestrea el nuevo dato en la señal **data_in.**
3. Estado **st3_rdy_out_mult_add**: Se deshabilita el BLOQUE 1, se habilita el BLOQUE 3, se realiza la multiplicación de los primeros 16 bits de la señal **signal_out_mult** con los primeros 16 bits de la señal **coeficientes**; luego se deshabilita el BLOQUE 3 y se habilita el BLOQUE 4 para realizar la primera acumulación. Este estado termina una vez que se hayan realizado todas las multiplicaciones y acumulaciones, según se indique de forma genérica en **coeff_number**.
4. Estado **st4_rdy_out_sistema**: Se coloca la muestra filtrada en **dato_out** y se conforma un frente de subida en **ready_out**.

Por último la conformación del demodulador según la figura 2 necesita de la interconexión de las cuatro estructuras de filtro FIR mostradas en la figura 5: dosfiltros configurados como pasa-banda y dos filtros como pasa-bajo. La transferencia de las muestras procesadas entre filtros se controla por otra máquina de estados. La recepción de los coeficientes de cada filtro se realiza de forma serie síncrona y luego se completan apropiadamente en cada módulo. La envolvente obtenida en ambas ramas se envía a la PC a través de la interfaz RS-232 y se procesa por la etapa de decisión previamente desarrollada en Matlab. La etapa de decisión se implementa en Matlab debido a que las señales moduladas reales pueden tener variaciones de amplitud, provocadas por ejemplo por efecto Doppler; esto puede provocar errores en la demodulación. En estos casos puede ser necesario corregir el diseño de los filtros, por lo quese visualiza en la interfaz de usuario las envolventes antes de demodular.La figura 7 muestra el resultado de la recuperación de las envolventes de una señal modulada con $f_1 = 3\ kHz$ y $f_2 = 5\ kHz$.

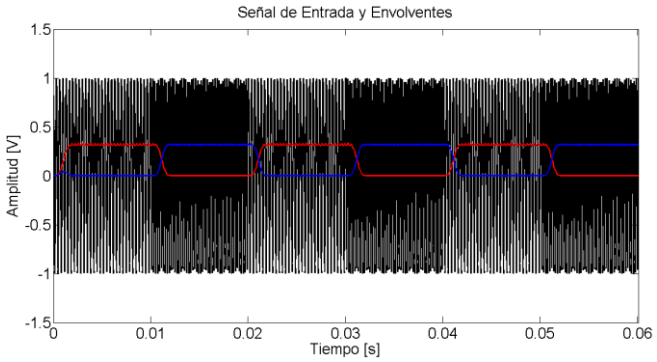


Figura 7. Envolvente recuperada para señal BFSK por el módulo IP.

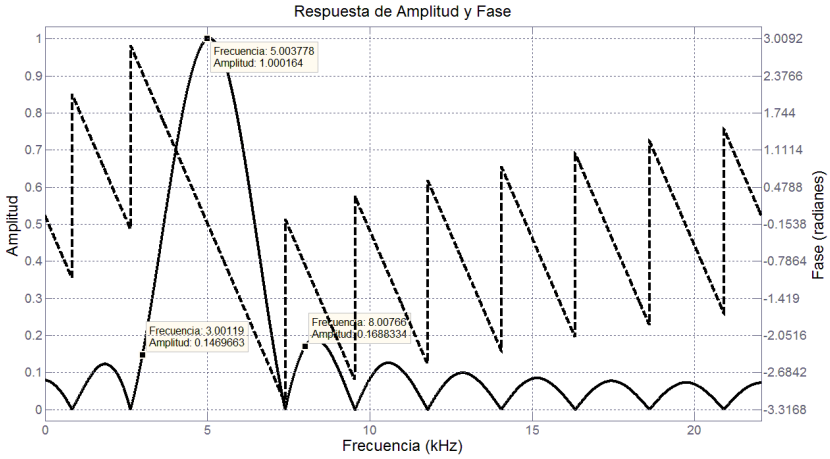


Figura 8. Respuesta de amplitud y fase del filtro pasa-banda de la rama superior.

Para valorar el efecto del ruido de cuantificación se considera a modo de ejemplo un caso con filtros de orden $\mathbf{18}$, los cuales según la herramienta fdatool de Matlab introducen una atenuación de aproximadamente $\mathbf{20}\ \boldsymbol{dB}$.

La respuesta de frecuencia por efecto de la cuantificación $\widehat{\boldsymbol{H}}(\boldsymbol{e^{jw}})$ de un filtro FIR de orden M se describe mediante la ecuación (3). Donde:$\widehat{\boldsymbol{h}}(\boldsymbol{n})$es la respuesta al impulso en forma cuantificada, $\boldsymbol{h}(\boldsymbol{n})$ es la respuesta no cuantificada, $\Delta\boldsymbol{h}(\boldsymbol{n})$ es el error en la respuesta al impulso, $\boldsymbol{H}(\boldsymbol{e^{jw}})$la respuesta de frecuencia ideal y $\Delta\boldsymbol{H}(\boldsymbol{e^{jw}})$ el error en la respuesta de frecuencia por cuantificación[25].

$$\widehat{H}(e^{jw}) = \sum_{n=0}^{M} \hat{h}(n) \cdot e^{-jw} = \sum_{n=0}^{M} [h(n) + \Delta h(n)] \cdot e^{-jw} = H(e^{jw}) + \Delta H(e^{jw}) \tag{3}$$

Debido a la relación aditiva que se presenta espectralmente en (3), es posible calcular de forma aislada una cota superior para el efecto de cuantificación. El estimado del error en la respuesta de frecuencia por cuantificación para 16 bits de precisión, viene dado por (4) como:

$$\left|\Delta H(e^{jw})\right| \leq \sum_{n=0}^{M} |\Delta h(n)| \leq M|\Delta h(n)_{max}| = (M+1)\frac{2 \cdot Xm}{2^{B+1}} = (18+1)\frac{2 \cdot 2}{2^{15+1}} \approx 0.001 = 10^{-3} \tag{4}$$

Para este caso se concluye que el ruido de cuantificación con carácter aditivo no perjudica considerablemente la respuesta de frecuencia,en el peor caso en la banda de atenuación se obtendrá un valor del mismo de $1.001 \cdot 10^{-1}$, lo cual no representa una variación considerable con respecto al valor de $1 \cdot 10^{-1}$ sin cuantificar. Este valor en la banda de atenuación para el peor caso permite realizar el diseño del filtro FIR a partir de la Forma Directa I como sugiere la figura 5; aunque para preservar la característica de fase lineal generalizada no resulte el más idóneo[25].

En la figura 8 se muestra la magnitud y fase de la respuesta de frecuencia de un filtro pasa-banda de la rama superior de orden 18, centrado en $f_2 = 5\ kHz$ y calculado según el método de la ventana de Kaiser. Para validar el funcionamiento del filtro FIR implementado en FPGA se aplica un tono a la entrada con amplitud unitaria y frecuencia variable. Cuando la frecuencia de dicho tono varía con valores de 3 kHz, 5 kHz y 8 kHz se obtienen a la salida del filtro valores de amplitud de 0.1631, 0.995 y 0.1952. Esto se corresponde aproximadamente con lo esperado según lo mostrado en la figura 8.

COMUNICACIÓN MATLAB, MICROBLAZE Y MÓDULO IP DEMODULADOR

En la figura 9 se muestra un diagrama de la secuencia de comunicación entre la PC, el procesador Microblaze y el módulo IP del demodulador. Primeramente se cargan los coeficientes de los filtros calculados en Matlabhacia el FPGA. Se comienza por enviar desde Matlab a Microblazeel número de coeficientes de cada uno de los cuatro filtros FIR que se emplearán. En Microblaze se crea un arreglo para almacenarlos, se envían secuencias de 8 coeficientes de acuerdo a la capacidad del buffer del módulo xps_uartlite, los cuales se almacenan en la memoria de MicroBlaze. Al término de recibir cada secuencia de coeficientes, el procesador MicroBlaze envía una confirmación al programa Matlab para indicar que está listo para recibir nuevas secuencias. De esta forma se envían todos los coeficientes de cada tipo de filtro.Luego de la recepción de todos los coeficientes solo falta decodificar la dirección apropiada en el módulo IP Detector de Envolvente y enviarle los coeficientes de cada filtro y sus señales correspondientes.

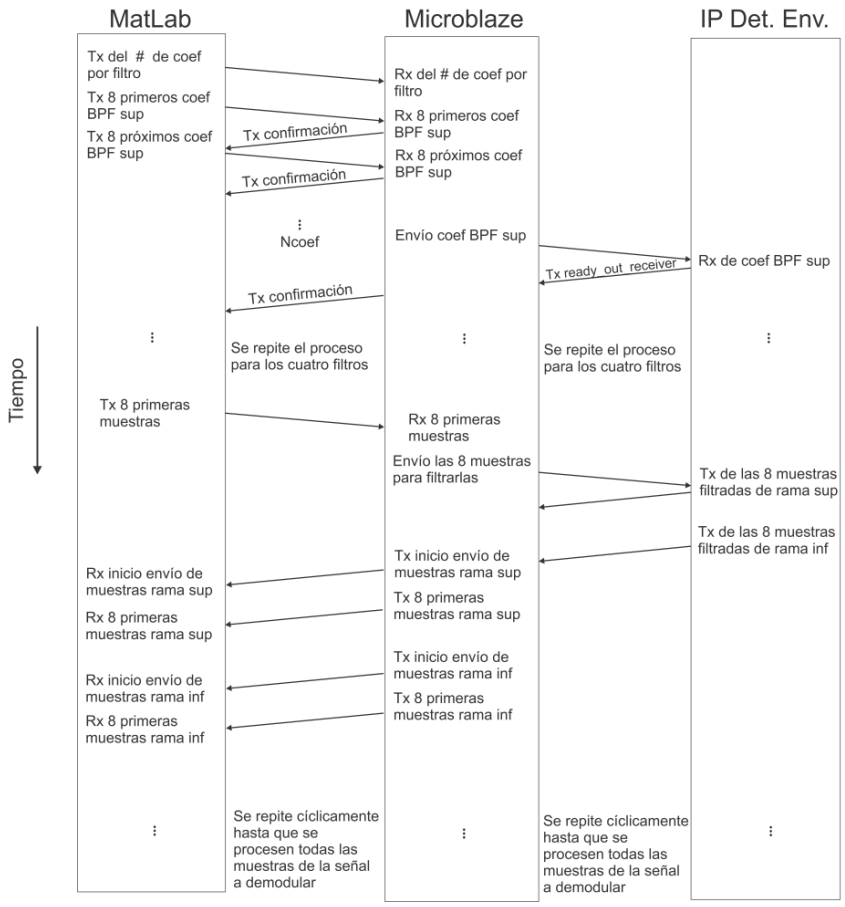


Figura 9. Diagrama de comunicación entre Matlab, Microblaze y el módulo IP Detector de Envolvente.

Una vez que se han almacenado todos los coeficientes de cada uno de los filtros en el módulo IP correspondiente, se envía la señal modulada BFSK. Las muestras se envían en secuencia de 8 datos, primeramente desde Matlab hacia Microblaze y de este hacia el módulo IP Detector de Envolvente. Luego de procesadas por cada una de las ramas superior e inferior del demodulador, se efectúa el proceso inverso y se reciben en el programa Matlab. Este proceso se repite hasta que se demodulen todas las muestras de la señal BFSK.

La etapa de decisión se implementa conla interfaz de usuario de la figura 3. A partir del resultado de las muestras procesadas por el módulo IP, se comparan las envolventes de las dos ramas muestra a muestra y finalmente se presenta el mensaje transmitido en lagráfica 4 de la aplicación de usuario de la figura 3.

DESEMPEÑO Y CONSUMO DEL SISTEMA

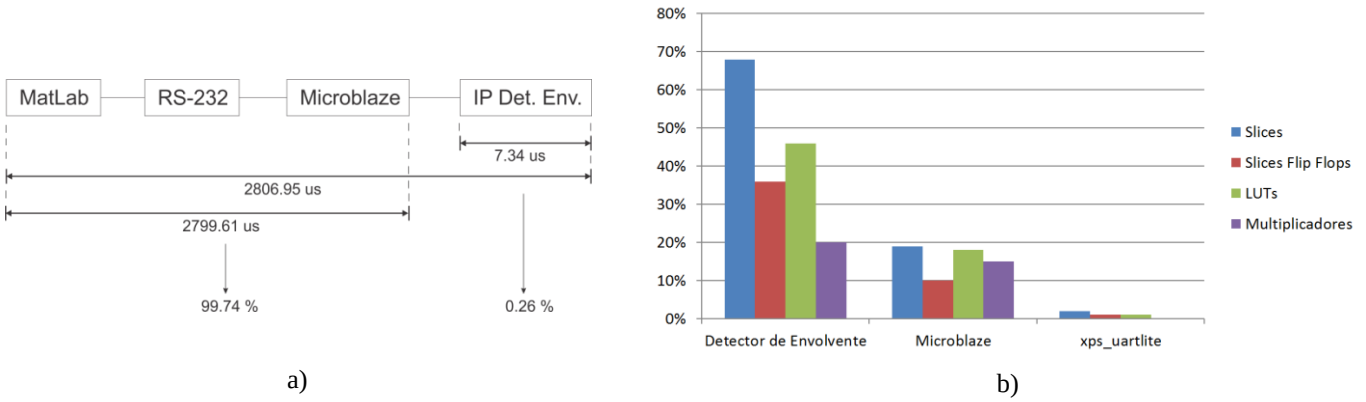
En la configuración del procesador Microblaze de la figura 4,se incluyó un módulo IP temporizadorpara contar pulsos de reloj.Este se emplea para calcular de forma precisa la demora de procesamiento del módulo IP Detector de Envolvente.La demora obtenida empleando filtros de orden 18**,** para una muestra de 16 bits es de 367 pulsos de reloj y equivale aproximadamente a 136240 muestras filtradas por segundo. Esta sería la frecuencia máxima de trabajo para poder implementar en tiempo real el demodulador FSK con un reloj de $50\ MHz$. Para una señal como la mostrada en la parte inferior de la figura 1, que contiene 2652 muestras, se obtiene una demora de 0.019465 segundos. Para una señal de 10 minutos y 22 segundos de duración, que contiene 13717440 muestras, se obtiene una demora de 100.685848 segundos.

La limitación fundamental del tiempo de procesamiento se introduce por el uso la comunicación serie RS-232. La figura 10 a) muestra el porciento de empleo del tiempo de procesamiento por cada una de las etapas. Se evidencia que se hace muy lenta la demodulación con respecto a la demora en el filtrado de la señal con el módulo IP Detector de Envolvente. Por este motivo es recomendable utilizar otra interfaz de comunicación más rápida, para así reducir el tiempo de procesamiento de la señal.

Por otra parte, si se realiza la demodulación de señales de larga duración con las funciones propias del Matlab, por ejemplo el caso de 13717440 muestras, el consumo de memoria RAM en la PC sería de aproximadamente 1.5 GB, esto representa una capacidad significativa para PCs con 2 GB de memoria RAM, la cual quedaría inutilizada durante el tiempo de procesamiento. Si se emplea la tarjeta Spartan-3E como demodulador se evita el consumo de memoria RAM de la PC.

Las simulaciones obtenidas han permitido comprobar el correcto funcionamiento del módulo IP Detector de Envolvente. Su verdadera potencialidad radica en la velocidad de procesamiento, esta será explotada cuando se use como demodulador en tiempo real mediante su interconexión con un conversor analógico digital que reciba las señales moduladas directamente.

El Spartan-3E del tipo XC3S500E tiene disponible para su utilización 4656 Slices, 9312 SliceFlipFlops, 9312 LUTs y 20 multiplicadores de 18 bits[24]. De estos recursos, el diseño del sistema que incluye el módulo IP Detector de Envolvente consume la mayoría de los Slices, como se muestra en la figura10 b). Al consumir un módulo IP Detector de Envolvente el 68% del total de Slices y el 46% de las LUTs, se sugiere utilizar por una parte un circuito con más nivel de integración. Este consumo de recursos se pudiera disminuir utilizando un módulo de memoria RAM externa disponible en la tarjeta empleada. En dicho módulo se almacenarían la señal coeficientes y la señal a la salida del bloque de retardadores de la figura 5. Ambas señales tienen una longitud igual al número de coeficientes, por lo que al no tener que guardar esta cantidad de información en lógica del circuito FPGA el consumo de recursos deberá disminuir apreciablemente.



a) b)

Figura 10.a)Porciento de empleo del tiempo total por cada uno de las etapas. b)Consumo de recursos del diseño en el FPGA.

CONCLUSIONES

De los distintos demoduladores para señales BFSK que prescinden del tiempo de símbolo, se escogió el Detector de Envolvente, principalmente por la simetría de sus elementos. Basta implementar la estructura de un solo filtro, replicarla adecuadamente, y ajustar sus coeficientes para que se comporte como pasa-banda o pasa-bajo. El demodulador basado en el Detector de Envolvente está conformado por cuatro filtros. Se realizó un modelo de filtro FIR para funcionamiento genérico con una estructura MAC. Este modelo genérico permite la reconfiguración según el número de coeficientes que se requieran. La estructura MAC permite minimizar el número de multiplicadores.

La solución de demodulación de señales BFSK en comunicación con una PC permite aliviar el consumo de recursos. Se pueden procesar señales de gran duración en PCs con bajas capacidades de memoria. El diseño se proyecta para otros empleos, como su uso compartido mediante una interfaz de comunicación Ethernet,pero puede implementarse también un demodulador en tiempo real si se emplea con un convertidor analógico digital.

La principal limitación en cuanto a la demora es producida por la interfaz de comunicación serie, por lo que se sugiere la utilización de una interfaz de comunicación más rápida.

El alto consumo de recursos en un FPGASpartan-3E XC3S500Esolo da margen a implementar un módulo IP Detector de Envolvente. Se propone la utilización de un módulo de memoria RAM externa, disponible en la tarjeta FPGA, para así almacenar la señal a la salida de los retardadores y la señal coeficientes. Esto permitirá disminuir el consumo de lógica interna del Spartan-3E y aumentar la capacidad para la implementación de otros módulos.

El diseño del Detector de Envolvente implementado en FPGA no superó los tiempos de procesamiento de Matlab, pero se demostró que para poder realizar demodulaciones a señales de millones de muestras en una PC es necesario al menos 2 GB de memoria RAM. Esto equivale a una inversión monetaria muy superior a la de una FPGA de la familia Spartan-3E.

REFERENCIAS

1. **Sklar B.** Digital Communications, Fundamentals and Applications. Second. Prentice Hall; 2001.
2. **Brewster RL, Jibrail WWS**. Detection of FSK and DPSK data signals by pulse compression. Communications, Radar and Signal Processing, IEE Proceedings F. 1982;129(4).
3. **Farrell KA, McLane PJ**. Performance of the cross-correlator receiver for binary digital frequency modulation. IEEE Transactions on Communications. 1997 Apr;45(5):573 –582.
4. **Shehab H, Ismail W, Singh M**.Low power FSK detection at low probability bit- errors, in International Conference on Electronic Design, 2008. ICED 2008, pp. 1–4.
5. **Al-Hussaini E, Al-Bassiouni A**. Performance of MRC Diversity Systems for the Detection of Signals with Nakagami Fading. IEEE Transactions on Communications. 1985 Diciembre;33(12):1315 – 1319.
6. **Tervo R, Zhou K.**DSP based self-tuning BFSK demodulation, in IEEE Pacific Rim Conference on Communications. Computers and Signal Processing, 1993, vol. 1, pp. 68–71.
7. **Al-Moosa N, Al-Araji S, Al-Qutayri M,** Fast acquisition digital tanlock loop with adaptive time delay,in TENCON 2004. IEEE Region 10 Conference, 2004, vol. 1, pp. 629–632.
8. **Huang K-H, Wang C-K**.A cost effective binary FSK demodulator for low-IF radios, in International Symposium on VLSI Technology, Systems, and Applications, 2001. Proceedings of Technical Papers, 2001, pp. 133–136.
9. **Widrow B, Glover J.R., McCoolJ. M., Kaunitz J, WilliamsC. S., HearnR. H., ZeidlerJ. R., Eugene DongJ, GoodlinR. C,** Adaptive noise cancelling: Principles and applications. Proceedings of the IEEE, 1975, vol. 63, no. 12, pp. 1692–1716.
10. **Ahmed N, Vijayendra S.** An algorithm for line enhancement. Proceedings of the IEEE,1982, vol. 70, no. 12, pp. 1459–1460.
11. **Zhao Z, Fu B, Xu C**.An Adaptive Demodulation Method for MFSK Signals under Alpha-Stable Distribution Pulse Noise, inCongress on Image and Signal Processing. CISP '08, 2008, vol. 1, pp. 65–69.
12. **Griffiths LJ**. Rapid measurement of digital instantaneous frequency. IEEE Trans. on ASSP. 1975;ASSP-23:207–22.
13. **Friedlander B**. A Recursive Maximum Likelihood Algorithm for ARMA Line Enhancement. IEEE Trans. on ASSP. 1982;30(4).
14. **David R, Stearn S, Elliot G, Etter D.M.** IIR algorithms for adaptive line enhancement, in Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '83., 1983, vol. 8, pp. 17–20.
15. **Werter M.J.**FSK demodulation with an adaptive direct form II filter, in IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP-93, 1993, vol. 3, pp. 328–331.

16. **Martin KW, Sun T**.Adaptive filters suitable for real-time spectral analysis. IEEE Journal of Solid-State Circuits, 1986, vol. 21, no. 1, pp. 108–119.
17. **Kadambari K, Rangarao KV, Mallik RK**. Demodulation of BFSK signals by adaptive digital notch filtering. Hyderabad; 2000. p. 219–217. Available from: http://rkmallik.tripod.com/~rkmallik/
18. **Thompson AC, Hussain ZM, O P**. A single-bit digital non-coherent baseband BFSK demodulator. Signal Processing Letters. Chiang Mai, Thailand: IEEE Region 10 Conference; 2004, vol. 1,pp. 515–518.
19. **Cui X, Yu D, Sheng S, Cui X**.A CORDIC Demodulator Platform for Digital-IF Receiver, in 8th International Conference on Solid-State and Integrated Circuit Technology, 2006. ICSICT ’06, 2006, pp. 2025–2027.
20. **Cui X, Yu D, Zhang X**.A 2-Level FSK Demodulator for Digital-IF Receiver, in IEEE Conference on Electron Devices and Solid-State Circuits. EDSSC, 2007, pp. 1175–1178.
21. **Boashash B**. Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals. Proceedings of the IEEE. 1992 Abril;80(4):520 –538.
22. **Holm S**.Optimum FFT-based frequency acquisition with application to COSPAS-SARSAT. IEEE Transactions on Aerospace and Electronic Systems, 1993, vol. 29, no. 2, pp. 464–475.
23. **Kaiser J, Schafer R**.On the use of the I0-sinh window for spectrum analysis. IEEE Transactions on Acoustics, Speech and Signal Processing, 1980, vol. 28, no. 1, pp. 105–107.
24. **Xilinx**. Spartan-3E Starter Kit Board User Guide. 2006.
25. **Oppenheim AV, Schafer RW, Buck JR**. Discrete-Time Signal Processing. Second. New Jersey: Prentice Hall; 1998.
26. **Orfanidis SJ**. Introduction to Signal Processing. Prentice Hall; 2010.

AUTORES

Karel Toledo de la Garza,Ingeniero en Telecomunicaciones y Electrónica, CITI, CUJAE, La Habana, Cuba, karel@udio.cujae.edu.cu

Jorge Torres Gómez, Ingeniero en Telecomunicaciones y Electrónica, Master en Sistemas de Comunicaciones, CITI, CUJAE, La Habana, Cuba,jorge.tg@electrica.cujae.edu.cu

Juan Raúl Rodríguez Suárez, Licenciado en Física, Dr.C.Técnicas, UPR, Pinar del Río, Cuba, jotar@tele.upr.edu.cu