

Rev. Cienc. Tecnol.

Año 13 / Nº 15 / 2011 / 53–60

## Codificación de voz mediante coeficientes de predicción lineal (LPC) sobre Microblaze.

### Voice encoding by means of lineal prediction coefficients embedded on microblaze

Alexey García Padilla, Elieser E. Gallego Martínez; Ismel Domínguez Rodríguez, Angel Correa Fernández, Juan R. Rodríguez Suárez

#### Resumen

En este trabajo se presenta el diseño de un codec de audio para telefonía VoIP implementado sobre un microprocesador MicroBlaze empotrado sobre un circuito FPGA de la familia Spartan 3E. Se describe primeramente el diseño del microprocesador y de los periféricos a emplear, entre los cuales se destaca el manejo de una memoria externa DDR y de una interfaz serie RS232 que permite comunicar el sistema con una computadora PC. En el trabajo se implementa una variante del algoritmo LPC10 con velocidad de 2.4 Kb/s y los algoritmos se implementan en el procesador en lenguaje C. La evaluación del sistema se realiza a partir del análisis de los datos enviados hacia la PC que son analizados con el procesador Matlab y comparados con resultados simulados. Este trabajo tributa a una tarea del proyecto "Plataforma de Conmutación de Paquetes", aprobado por el MIC al ISPJAE con participación de la UPR y GKT.

Palabras clave: Codec, FPGA, LPC-10, MicroBlaze, VoIP.

#### Abstract

In this work the design of an audio codec is presented for VoIP telephony, implemented on a microprocessor MicroBlaze embedded on a FPGA circuit of the Spartan 3E family. The design of the microprocessor is described first, and then, the peripherals to use, among which the handling of an external memory DDR and of a series interface RS232 allowing communication of the system with a PC computer. In this work, a variant of the algorithm LPC10 is implemented with speed of 2.4 Kb/s and the algorithms are implemented in the processor in C language. The evaluation of the system is made from the analysis of the data sent to the PC, which are analyzed with the Matlab processor and compared with simulated results. This piece of work is related to a task of the project "Platform of Commutation of Packages", approved by the MIC to the ISPJAE with participation of the UPR and GKT.

Key words: Codec, FPGA, LPC-10, MicroBlaze, VoIP.

#### Introducción

Con el comienzo de la creación de redes de computadoras a finales de la última década del siglo pasado, se vio que el principio de conmutación y transmisión de paquetes de datos en la cual esta se basaba podría ser aplicado también a la transmisión de voz previamente digitalizada. Surgía entonces una alternativa a la red pública telefónica de servicios conmutados, la telefonía sobre IP o VoIP. En lugar de lograr una conexión dedicada punto a punto entre dos usuarios remotos conectando adecuadamente las líneas necesarias, como se hace en la telefonía tradicional, en VoIP se digitaliza la señal de voz y se forman paquetes de datos que son enviados según el protocolo de Internet IP por la red de computadoras [5]. En el lado receptor se reciben los paquetes, se ordenan y se reconstruye la señal original de voz. Los recursos se comparten y de esta forma

se pueden transmitir más llamadas mediante la conmutación de paquetes de voz. Para la ejecución de servicios de VoIP, se necesita un bloque de adaptación [3], también conocido como compuerta o pasarela VoIP (Gateway VoIP) que permite conectar un aparato telefónico con una red de computadoras. Este bloque que debe trabajar en modo dúplex comprende entre otros elementos: un conversor analógico digital (ADC) para obtener las muestras de la señal de voz, la implementación de los algoritmos para instrumentar los protocolos IP para la formación, ordenamiento e identificación de los paquetes y un conversor digital analógico (DAC) para de nuevo sintetizar la señal analógica de la voz recibida.

Si se digitaliza la señal de voz a 8 kHz, con 16 bits resulta en que la velocidad de transmisión necesaria es de 128 kb/s. Se hace necesario introducir algoritmos para disminuir la velocidad de transmisión y mantener un com-

promiso con la calidad de reproducción de la voz. Estos algoritmos de compresión se denominan CODEC de voz o vocoders. [7]

Dada la existencia de una línea de transmisión de datos a lo largo de todo el país, se piensa que la telefonía sobre IP puede ser una solución económicamente viable para conectar telefónicamente algunas localidades remotas.

En este trabajo se presentan los resultados en la implementación de un bloque, CODEC, para una pasarela telefónica de VoIP empleando un procesador empotrado virtualmente en un circuito FPGA de la familia Spartan3e. Además del procesador también se sintetizan otros circuitos necesarios como un controlador de conversión ADC y un controlador de conversión DAC especialmente diseñados para este trabajo, un controlador de memoria DDR externa y un controlador para comunicación RS-232. Desde el punto de vista del software se presenta además el código en lenguaje de programación C para implementar el algoritmo LPC-10E escogido para el CODEC. Los resultados obtenidos se comparan además con los obtenidos implementando el modelo desde el programa Matlab, ver. 7.0, exportando los resultados mediante la comunicación serie hacia una PC.

En las secciones siguientes se aborda brevemente lo relativo a los dispositivos electrónicos empleados en el trabajo, algunas de sus características principales, así como los algoritmos y métodos empleados para lograr la implementación del codec, y los resultados más relevantes acompañados de algunas consideraciones realizadas al respecto.

## Materiales y métodos

Este trabajo ha sido implementado sobre una tarjeta Spartan3e que contiene entre otros un circuito FPGA xc3s500e-4fg320 con alrededor de 10000 celdas lógicas, 20 multiplicadores empotrados y 20 bloques de memoria RAM para un total de 320 kb. Se ha usado también de dicha tarjeta la memoria externa DDR SRAM de 32 MB-ytesx16bits, el conversor DAC LTC2624 serie de 12 bits y el controlador de interfaz RS-232. En una tarjeta auxiliar se incorporaron un conversor serie ADC MAX187 de 12 bits y dos filtros paso bajo analógicos programables para implementar el filtro anti-réplicas y el filtro reconstructor correspondiente.

Para el desarrollo del trabajo fue necesario configurar primeramente el hardware del procesador, y elaborar el software de la aplicación a fin de implementar los algoritmos de procesamiento de la compresión de voz.

## Configuración del procesador

Por tratarse de un microprocesador empotrado por software en un circuito FPGA, se hace necesario primeramente

diseñar el microprocesador seleccionando los recursos necesarios para la aplicación en cuestión. El procesador MicroBlaze para circuitos de la firma Xilinx, es un procesador de 32 bits, con arquitectura de Harvard que brinda memorias y buses de datos separados para programas y datos, trabaja con aritmética de 32 bits de precisión del tipo punto flotante según la norma IEE574 y que posee además una unidad opcional para procesamiento en punto flotante que acelera dichas operaciones. Todas estas facilidades lo hacen especialmente atractivo para implementar algoritmos de procesamiento digital donde están involucradas operaciones matemáticas importantes.

Para el diseño del microprocesador se ha empleado la herramienta EDK8.1 en particular la plataforma XPS que posibilita además la creación de las bibliotecas de los controladores, el diseño de las aplicaciones de software empleando lenguaje C/C++, la compilación y depuración de los errores, la simulación del sistema, la valoración del consumo de tiempo y finalmente la creación del fichero .bit que contiene la información conjunta relativa al software y el hardware para programar la ejecución del procesador en el circuito en la FPGA.

Para este trabajo se ha diseñado un microcontrolador con 50 MHz de reloj, con controlador de interrupciones, un temporizador con interrupción para ajustar la frecuencia del muestreo del sistema a 8 kHz, la unidad de punto flotante, los periféricos para controlar la memoria DDR, y la interfaz RS-232.

## Creación e importación de periféricos adicionales

Un aspecto importante en el diseño de esta aplicación es la creación e importación de nuevos periféricos. Por ejemplo se necesita comunicar los conversores ADC y DAC con el procesador a fin de que este lea o escriba los valores de las muestras digitalizadas de la voz y las procese convenientemente. A diferencia de un microcontrolador típico que posee puertos dedicados, el MicroBlaze posee un conjunto de buses para conectar con otros dispositivos. Para establecer dicha conexión se hace necesario la creación de un controlador de dicho periférico al bus en cuestión. Se puede emplear el bus OPB de MicroBlaze para conectar periféricos, sin embargo en este trabajo se ha empleado la conexión a la interfaz FSL a partir de la creación del periférico que se oferta como una de las opciones del programa de generación de sistemas XSG.

XSG es una herramienta que se instala sobre Simulink de Matlab y que está optimizada para crear sistemas de procesamiento digital en los circuitos FPGA de Xilinx. Comprende un conjunto de bibliotecas de funciones y de implementación de módulos de propiedad intelectual IP, permite la simulación del circuito así como la síntesis y creación del fichero bit de configuración. Puesto que los controladores para los conversores ADC y DAC habían sido creados con dicha herramienta, se empleó la opción de



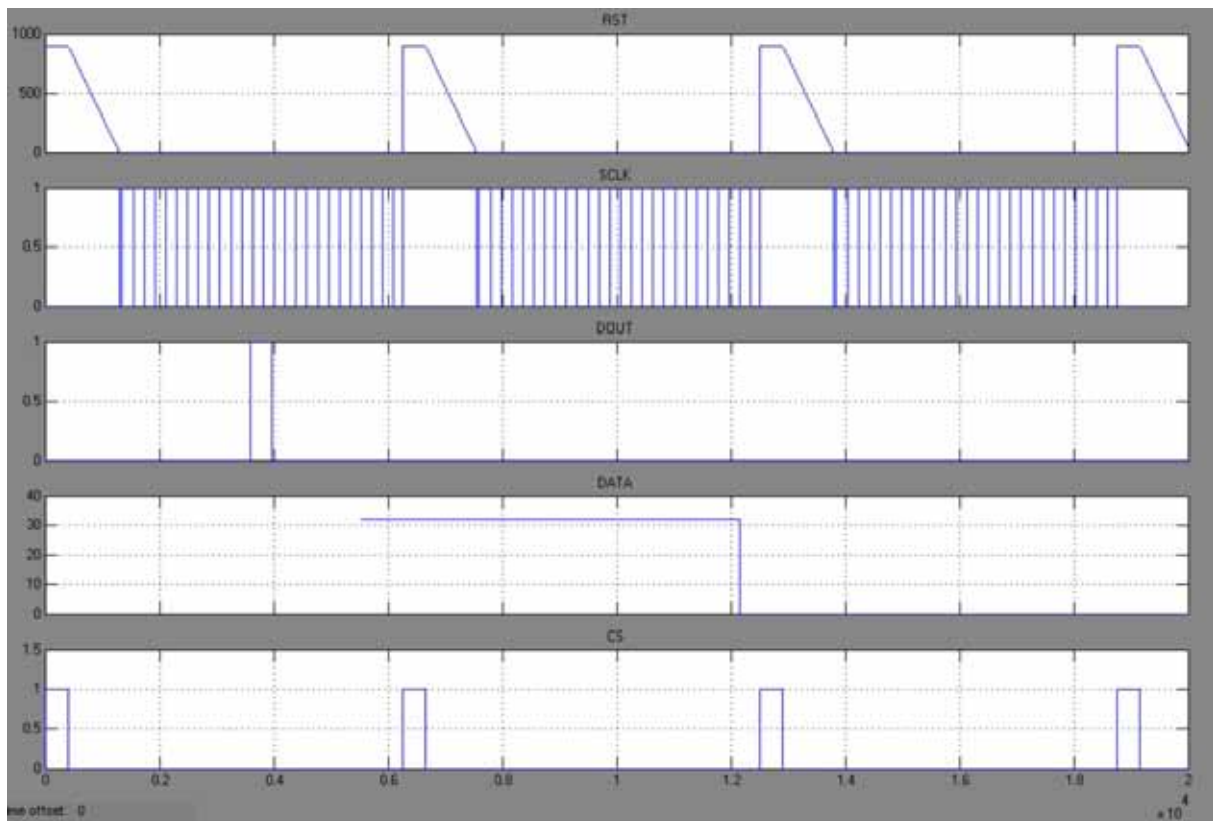


Fig. 2. Carta de tiempo del conversor MAX187.

interpretan como voltajes mayores que la mitad del voltaje de referencia y las negativas como voltajes menores.

Un procedimiento similar se emplea para controlar el conversor DAC, pues se trata también de un conversor unipolar de 12 bits de formato serie, con terminales CS, SCLK y DIN. En este caso para generar el voltaje correspondiente a un cierto código o número, se implementa primero una palabra de control de 32 bits que contiene el canal del conversor a emplear así como el código de conversor. Esta palabra se convierte de formato paralelo a formato serie y se va transmitiendo bit a bit por el terminal DIN del conversor con las transiciones del reloj SCLK del mismo.

#### Controladores de filtros analógicos

Se emplean dos filtros analógicos MAX295, que son filtros paso bajo de orden 8 con aproximación de Butterworth para implementar el filtro anti-réplicas y el filtro recuperador. La frecuencia de corte del filtro se escoge como 4 kHz, la frecuencia de Nyquist del sistema, y se programa implementando un reloj de 200 kHz, que se corresponde con una frecuencia 50 x la frecuencia de corte. Puesto que este sistema no necesita ser controlado por el procesador y no tiene asociado ningún registro de lectura o escritura, se ha adicionado a uno de los bloques anteriormente descritos.

#### Vocoders [7] [5]

El término vocoder se emplea para designar a los CODECS especialmente concebidos para aplicaciones de voz. A lo largo de los años se elaboró un gran número de CODECS. Existen estándares establecidos por la Unión Internacional de Telecomunicaciones (UIT) para la codificación de la voz, a continuación se muestra una tabla con las principales características de algunos de ellos. La norma G.729 es la asignada para servicios de VoIP, sin embargo en este trabajo se ha decidido emplear la norma FS1015 debido a la simplicidad de su modelo, y así ganar experiencia primeramente en la implementación de la misma. Por otra parte mucho de sus bloques son parte integrante de otros CODECS más complejos. Esta norma se basa en emplear un modelo de predicción lineal para la generación de la voz. La idea fundamental es que a partir de una determinada muestra de la voz o trama y aplicando un modelo determinado para la generación de la misma, se obtiene un conjunto de parámetros que definen el modelo. El número total de bits asociado a estos parámetros resulta menor que el número total de bits correspondiente a la trama de voz. Por lo tanto es más factible transmitir estos parámetros del modelo y no la señal de voz. En el lado receptor se reciben los parámetros del modelo y se implementa el mismo incluyendo los estímulos necesarios. Aunque la señal generada no es una copia fiel de la trama inicial, tiene la calidad audible suficiente para distinguirse en el proceso de comunicación.

**Tabla I:** Normas de los codecs

Norma	Descripción	Ancho de banda kbps	Retardo
G.711	PCM	64	< 1ms
G.721	Adpcmfs 1015	32, 16, 24, 40	< 1ms
G.728	LD-CEL	16	~ 2.50
G.729	CS-ACELP	8	~15.00
G.723.1	Multirate CELP	6.3, 5.3	~30.00
Fs1015	LPC10	2.4	----

**Modelo de predicción lineal LPC para la voz**

La predicción lineal (LP) se aplica para obtener un modelo para la producción de señales de voz. Se puede asumir que el tracto vocal, se puede describir por un filtro todo polo de respuesta impulsiva infinita (IIR) con una función transferencial  $H(z)$  dada por:

$$H(z) = \frac{G}{1 + a_1 * z^{-1} + a_2 * z^{-2} + \dots + a_p * z^{-p}} \quad (1)$$

Los coeficientes se asumen constantes en el tiempo sólo para tramas de voz de un tamaño determinado, en dependencia de la trama la señal de excitación del sistema será un tren de impulsos si la trama se clasifica como sonora o una señal aleatoria si es no sonora. Este modelo se resume en la figura 3.

La ecuación en diferencias del sistema que modela el tracto vocal y genera las muestras resulta en:

$$s(n) = \sum_{k=1}^p s(n-k)a_k + Gu(n) \quad (2)$$

El término de la sumatoria en la ecuación (2), puede interpretarse como un predictor lineal en el cual la muestra de salida se puede generar mediante una combinación lineal de  $p$  muestras anteriores. Se puede calcular el error de predicción  $e(n)$  como:

$$e(n) = s(n) - s'(n) = Gu(n) \quad (3)$$

Comparando las ecuaciones (1) y (3), puede considerarse que la generación de la secuencia de error de predicción

es un sistema FIR inverso al sistema IIR que modela el tracto vocal.

El problema fundamental de la predicción lineal de la voz es la determinación del conjunto de coeficientes  $\{a_k\}$  para cada trama que haga mínimo el error de predicción  $e(n)$  medio cuadrático. Una vez determinados los coeficientes para esa trama, si se aplica la excitación adecuada al sistema determinado por la ecuación (1), se obtendrá a la salida una secuencia que reproducirá la trama en cuestión no exactamente en el dominio del tiempo pero si de sus propiedades espectrales. El procedimiento básico de un CODEC de voz comprende entonces dos partes:

Codificación de la trama: cálculo de los parámetros del modelo de la trama.

- V: clasificación de la trama como sonora o sorda.
- G: nivel de energía de la trama.
- Conjunto de coeficientes  $\{a_k\}$
- P: Tiempo que transcurre entre pulsos de excitación consecutivos para trama sonora.

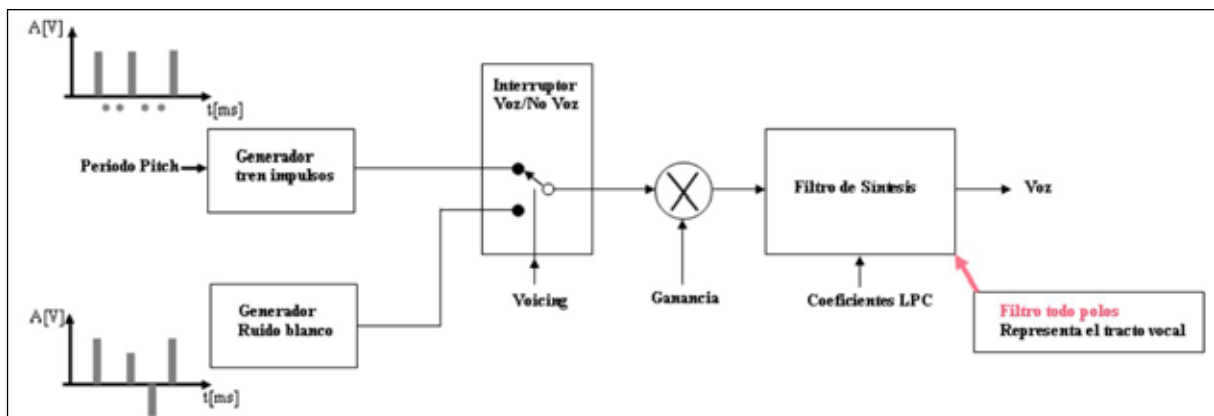
Se transmiten estos parámetros, lo cual conlleva a transmitir menos bits en relación a la trama original.

Decodificación de la trama. Se reciben los parámetros y se implementa con ellos el modelo de producción de voz para esa trama.

**Modelo LPC-10 [2]**

Los diagramas de bloques relativos al codificador y decodificador del vocoder LPC-10 se muestran en las figuras 4 y 5.

En el bloque de codificación la señal de voz adquirida de la digitalización a 8 kHz se conforma en tramas de 20 ms, con un solapamiento entre ellas de 10 ms. La trama es procesada en varias trayectorias a fin de determinar los parámetros del modelo. Estos procesos comprenden un filtro de pre-énfasis, la clasificación de la trama en sonora o sorda para definir el tipo de estímulo, el cálculo de tiempo



**Fig. 3** Modelo de producción de la voz

de pitch para tramas sonoras, el cálculo de los coeficientes del filtro LPC y el cálculo de la ganancia.

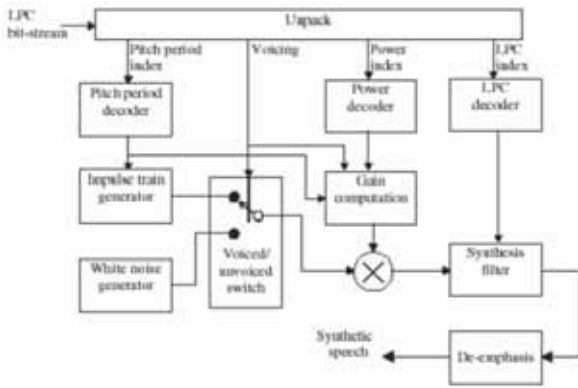


Fig. 4. Codificador LPC completo usando la norma FS1015.

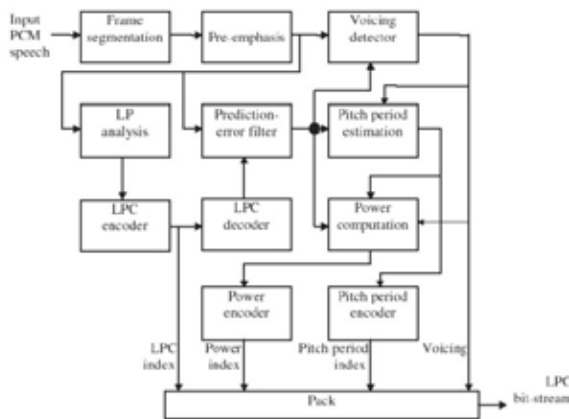


Fig. 5. Decodificador LPC completo usando la norma FS1015.

### Pre-énfasis

El espectro típico de la señal de voz tiene una atenuación en las altas frecuencias debido al efecto de radiación del sonido desde los labios [2] [4]. Esto puede afectar el cálculo de estas componentes de frecuencia, por ello la trama de voz se procesa por un filtro paso alto que incremente la energía de las componentes de alta frecuencia. Este filtro se denomina de pre-énfasis puede implementarse como un filtro FIR con un cero cercano a la unidad con una ecuación en diferencias dada por:

$$y(n) = x(n) + x(n-1) * 0.9378 \quad (4)[5]$$

### Clasificación de la trama

El límite de la clasificación V/UV no siempre está claro. Fundamentalmente esto ocurre para tramas de transición en las que la señal pasa de voz a no voz, o viceversa. Esta es una de las limitaciones de este modelo.

Los parámetros fundamentales a tener en cuenta para realizar esta clasificación son: Función Magnitud Suma (MSF), Cruces por Cero (ZC), Predicción de Ganancia

(PG). Para obtener una clasificación eficiente de la trama se deben tener en cuenta el resultado de los tres criterios.

La Función Magnitud Suma (MSF) es igual a la energía calculada para la trama por lo tanto se usa en mayor cantidad de oportunidades por las facilidades que brinda para su cálculo. Esta función se expresa como sigue:

$$MSF[m] = \sum_{n=m-N+1}^m |s[n]| \quad (5)$$

La voz tiene concentrada la mayor cantidad de energía en la región de bajas frecuencias, es por eso que para alcanzar una mejor discriminación de ella se introduce un filtro paso bajo antes de efectuar el cálculo de la misma. Un ancho de banda de 800 Hz es adecuado para este propósito pues la más alta frecuencia del diapason (de la voz humana) es alrededor de los 500 Hz.

Un valor elevado, aproximadamente de 250 000 de la MSF estará relacionado por consiguiente con una trama de voz y un valor bajo con una trama de no voz. [3]

La proporción de cruces por cero (ZC) de una trama de voz que termina en el instante  $m$  está definida por:

$$ZC[m] = \frac{1}{2} \sum_{n=m-N+1}^m |\text{sgn}(s[n]) - \text{sgn}(s[n-1])| \quad (6)$$

Para una trama de voz sonora la proporción de cruces por cero es relativamente baja,  $ZC < 20$ , debido a la presencia de la componente de frecuencia del diapason (de naturaleza de baja frecuencia). Y para una trama sorda los cruces por cero son altos  $> 20$ , debido a la apariencia de ruido de la señal con una alta porción de la energía en las altas frecuencias.

La predicción de ganancia PG se define como la proporción de la energía de la señal y la energía del error de predicción:

$$PG[m] = 10 * \log_{10} \left( \frac{\sum_{n=m-N+1}^m s^2[n]}{\sum_{n=m-N+1}^m e^2[n]} \right) \quad (7)$$

Para que ocurran cambios en la determinación de la condición de las tramas teniendo en cuenta la PG en voz o no, estas deben como promedio tener un valor de 3 dB o más de diferencia, entonces se dice que la trama actual cambia su condición con respecto a la anterior, principalmente debido al hecho de que la periodicidad implica una mayor correlación entre las muestras, y así son más fácil de predecir. Las tramas de no voz son más aleatorias y por consiguiente menos predecibles.

### Cálculo de tiempo Pitch

El tiempo sucesivo en que se abren las cuerdas vocales es llamado período fundamental, período de diapason, o período del pitch. En términos de tiempo, para los



hombres es aproximadamente de 4 ms a 20 ms, mientras que para las mujeres, es de 2 ms a 8 ms [6]. El diseño de un algoritmo de estimación del período fundamental es complejo y de su correcta determinación depende la calidad del vocoder, y de varios factores que atentan contra la correcta determinación del mismo, en particular que la trama en realidad no tiene una periodicidad perfecta y el ruido e interferencia siempre están presentes.

En la práctica la estimación del período fundamental es implementada como un compromiso entre la complejidad computacional y la calidad que se desea obtener. La señal de voz es pasada por un filtro paso bajo antes de ser usada para la estimación del pitch puesto que la frecuencia fundamental asociada con la voz está localizada en la región de frecuencias menores que 500 Hz. En este trabajo se emplea la autocorrelación de la trama para determinar el pitch.

**Cálculo de los coeficientes de regresión**

Para el cálculo de los coeficientes de regresión se emplea el método de Levinson-Durbin que permite calcular recursivamente la solución a una ecuación que involucra la matriz de Toeplitz, que resulta al evaluar el error medio de predicción. [5]

$$\begin{bmatrix} r(0) & r(1) & \dots & r(p-1) \\ r(1) & r(2) & \dots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{bmatrix} \quad (8)$$

Donde  $r(k)$  son los coeficientes de autocorrelación de la trama calculados y  $a_k$  son los coeficientes del filtro de síntesis que se utiliza en el modelo LPC, en este caso como se trata del vocoder LPC10 se calculan 10 coeficientes del filtro y por tanto 11 valores de la función de autocorrelación.

Debido a la simetría que presenta la matriz de Toeplitz, pues sus diagonales contienen los mismos elementos, el algoritmo puede ser usado para resolver el sistema. Las ecuaciones resultantes de este procedimiento son [5]:

$$E^{(0)} = r(0) \quad (9)$$

$$k_i = \frac{r(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} r(i-j)}{E^{(i-1)}} \quad (10)$$

$$a_i^{(i)} = k_i \quad (11)$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \quad (12)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \quad (13)$$

**Ganancia**

En este trabajo se emplea el cálculo de la ganancia  $G$ , dada por la ecuación (14):

$$G^2 = R(0) - \sum_{k=1}^p a_k R(k) = \varepsilon_p \quad (14)$$

En esta expresión se muestra la dependencia de la ganancia de los coeficientes de autocorrelación y de los coeficientes del filtro calculado por el método de Levinson.[1]

**Resultados y discusión**

Los algoritmos anteriormente explicados se implementaron en Matlab y en código C para el procesador Microblaze, soportado sobre la tarjeta FPGA Spartan-3E de Xilinx.

Se generó una señal de prueba que es procesada paralelamente por Matlab y el Microblaze y se compararon los resultados obtenidos, este proceso de comprobación fue realizado para cada uno de los bloques del codec LPC10, pero se mostrarán solamente los resultados relevantes de una trama de 160 muestras, o sea, los parámetros que se envían al decodificador.

En la tabla siguiente se muestra la comparación cuantitativa de los resultados obtenidos para los coeficientes LPC, el período fundamental, pitch, la clasificación de voz y la ganancia para una trama de prueba, tanto en Matlab como en Microblaze.

**Tabla II.** Comparación de los resultados.

Parámetro	Valor en Microblaze	Valor en Matlab
	1	1.0000
	-1.34549761	-1.3442
	0.295958906	0.2926
	0.24395521	0.2474
	0.191256672	0.1900
	0.11971584	0.1199
	0.0444279015	0.0429
	-0.0346283019	-0.0317
	-0.0925646871	-0.0952
	-0.145064414	-0.1438
Coefficientes LPC	0.341273516	0.3411
Pitch	28	28
Voice/Unvoice	0	0
Ganancia	3.63050723	3.6305

Finalmente se realizó la comparación de la señal original y la señal sintetizada a través de los parámetros del codec LPC10 procesada en Matlab y en Microblaze, y el análisis de la calidad de servicio (QoS). Para ello la señal es enviada desde Matlab al procesador mediante el puerto serie, y este devuelve la voz sintetizada por el Microblaze y se compara con la voz sintetizada por el Matlab.

La figura 10 muestra los resultados de esta comparación.

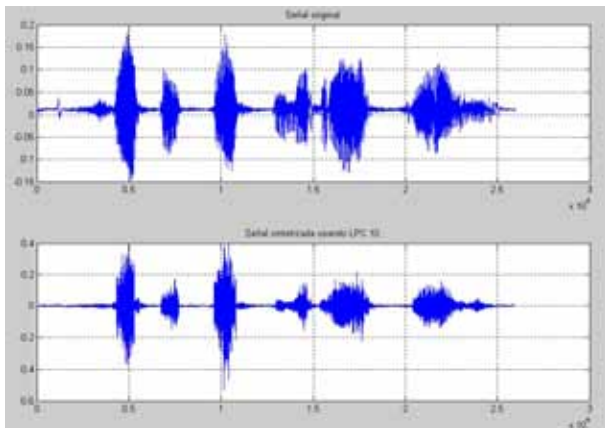


Fig. 10. Comparación de la señal original con la sintetizada usando Matlab.

En la figura anterior se muestra una comparación en el dominio del tiempo, de las señales involucradas en el experimento realizado. En la parte superior se muestra la señal original que se sometió al proceso de codificación empleado el algoritmo ya mencionado, y en la parte inferior la forma de onda de la señal que se sintetizó posterior a los procesos de codificación y decodificación. Como se puede observar la forma de onda de ambas señales es casi idéntica, sin embargo es preciso señalar que subjetivamente, o sea, al escuchar ambas señales, se puede notar una pequeña diferencia entre ellas.

## Conclusiones

Se diseñó e implementó el codec para voz IP LPC10 soportado sobre un microcontrolador Microblaze empotrado en una tarjeta Spartan 3E, teniendo en cuenta todas las recomendaciones que ofrece el algoritmo Lineal Prediction Code, para las transmisiones de voz a 2.4 kbp.

Se comprobó además el correcto funcionamiento del mismo evaluando su calidad de servicio y comparándolo con un procesamiento similar en Matlab, permitiendo la evaluación de cada uno de los bloques funcionales que conforman el algoritmo de codificación, mediante la herramienta de simulación virtual que proporciona XPS.

A partir de este trabajo se ha logrado obtener una plataforma experimental para la evaluación de los distintos tipos de codec de audio. Se pretende, usando la misma plataforma y metodología, evaluar la implementación de codecs de voz de prestaciones superiores, tales como el G729 o G723.

## Referencias Bibliográficas

1. Campbell, J. P., Tremisín, Jr. T. E. *Voiced/Unvoiced classification of speech with application to the U.S. Government LPC-10E algorithm*, 1° ed., New York, Prentice-Hall, 1982.
2. Haykin, S., *Communication Systems*, 4° ed., New York, Jhon Wiley & Sons, 2008.
3. Lin, W., Ngeeh Koh, S., Lin, X., *Mixed Excitation Linear Prediction Coding of Wideband Speech at 8 kbps*, 1° ed., Massachusetts, Jhon Wiley & Sons, 2006.
4. Peleg, N., *Linear Prediction Coding*, 1° ed., New York, Jhon Wiley & Sons, 2009.
5. Rabiner, L.R., *Digital Processing of speech Signal*, 1° ed., New Jersey, Prentice-Hall, 1978.
6. Rabiner, L. R., *A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition*, 1° ed., New Jersey, Prentice-Hall 1980.
7. Therrien, C. W., *Discrete Random Signals and Statical Signal Processing*, 1° ed., New York, Prentice-Hall, 1992.

Recibido: 02/02/09

Aprobado: 04/11/11

- C. Juan Raúl Rodríguez Suárez<sup>1</sup>  
Profesor Auxiliar del Departamento de Telecomunicaciones y Electrónica de la Universidad de Pinar del Río. Cuba. Miembro del Grupo de Investigación para el Diagnóstico Avanzado de Maquinarias GIDAM. Trabaja en las temáticas de procesamiento digital de señales, sensores e instrumentación.
- Ángel Correa Fernández<sup>1</sup>  
Profesor Instructor del Departamento de Telecomunicaciones y Electrónica de la Universidad de Pinar del Río. Cuba. Miembro del Grupo de Investigación para el Diagnóstico Avanzado de Maquinarias GIDAM. Trabaja en las temáticas de procesamiento digital de señales.
- Elieser Gallego Martínez<sup>1</sup>  
Miembro del Grupo de Investigación para el Diagnóstico Avanzado de Maquinarias GIDAM.
- Alexey García Padilla<sup>1</sup>  
Miembro del Grupo de Investigación para el Diagnóstico Avanzado de Maquinarias GIDAM.
- Ismel Domínguez Rodríguez<sup>1</sup>  
Miembro del Grupo de Investigación para el Diagnóstico Avanzado de Maquinarias GIDAM.

1. Universidad de Pinar del Río, Cuba, Martí 270, final. Departamento de Telecomunicaciones y Electrónica.  
(alexey@tele.ur.edu.cu, elieser@tele.upr.edu.cu, ismel@tele.upr.edu.cu, angel@tele.upr.edu.cu, jotar@tele.upr.edu.cu)