

# “DISEÑO E IMPLEMENTACIÓN EN SOFTWARE Y EN HARDWARE DE UN DECODIFICADOR VITERBI PARA EL ESTÁNDAR IEEE802.11a WLAN”

Ricardo Villafuerte<sup>1</sup>, Luis Oñate<sup>2</sup>, Nury Ramírez<sup>3</sup>, Pedro Vargas<sup>4</sup>.

<sup>1</sup>Ingeniero Electrónico en Telecomunicaciones 2007; email: [ricardo\\_dx@yahoo.es](mailto:ricardo_dx@yahoo.es)

<sup>2</sup>Ingeniero Electrónico en Telecomunicaciones 2007; email:

[betto2002om@hotmail.com](mailto:betto2002om@hotmail.com)

<sup>3</sup>Ingeniero Electrónico en Telecomunicaciones 2007; email: [nuryramirez@ieee.org](mailto:nuryramirez@ieee.org).

<sup>4</sup>Director de Tesis, Ingeniero Electrónico en Telecomunicaciones, Escuela Superior Politécnica del Litoral, 1973. Postgrado RUSIA, Instituto Electrónico de Telecomunicaciones Von Bruevich, 1976. Profesor de ESPOL desde 1985 email: [pvargas@espol.edu.ec](mailto:pvargas@espol.edu.ec).

## **RESUMEN**

El trabajo de tesis presenta el diseño y la implementación en software y en hardware de un decodificador de Viterbi para el estándar de comunicaciones inalámbricas IEEE 802.11a.

El prototipo del decodificador de Viterbi fue implementado sobre el conjunto de instrucciones de un procesador MIPS de la serie R2000, constituyendo el diseño en software, y también fue implementado en VHDL, lo que constituye el diseño en hardware. Ambas implementaciones tienen las siguientes características: Decodificador convolucional de decisión dura para la entrada y de decisión suave para la salida, Longitud Restrictiva de 7 y Tasa de Codificación de  $\frac{1}{2}$ , según el estándar.

Las implementaciones fueron sintetizadas usando las herramientas de Xilinx y Altera, fueron probados funcionalmente mediante simulación y programados sobre cuatro diferentes familias de FPGAs. La verificación mediante una cama de pruebas diseñada para este fin, la cual permitió comprobar que el diseño satisface los requerimientos planteados.

## **ABSTRACT**

This paper presents a design and implementation in software and hardware of a Viterbi decoder, based on the IEEE 802.11a standard.

The decoder prototype was implemented over the instruction set of a MIPS processor of the R2000 series, which represents the software design. Also, the decoder was implemented in VHDL, which is the hardware design. Both implementations are

capable of processing with a constraint length 7 and code rate  $\frac{1}{2}$ , according to the standard.

The implementations were synthesized using the Xilinx and Altera tools, proven by means of simulation and programmed over 4 different FPGAs families. The functional verification was done on a specific test-bench designed for our purposes, which ensures that the circuit implements intended functionality.

## **INTRODUCCIÓN**

En comunicaciones digitales los datos transmitidos se presentan en forma binaria, estos datos son modulados hasta obtener ondas analógicas que son transmitidos a través de un canal hacia un receptor. En el canal el ruido y la interferencia corrompen la señal transmitida, la cual vuelve a convertirse en datos binarios o bits en el receptor. Algunos errores de bits pueden ocurrir si la interferencia es muy fuerte, por eso se usa codificación de canal para prevenir estos errores. La codificación de canal significa que bits adicionales se añaden a los datos transmitidos para que luego estos bits puedan ser usados cuando se reconstruye la secuencia de datos transmitida en el receptor. Hay muchos métodos para codificación de canal empleados en su mayoría para los últimos estándares de comunicaciones inalámbricas como: 3GPP, GSM and WLAN, por ejemplo codificación convolucional y entre ellas la decodificación Viterbi, que representa una de las técnicas de codificación de canal más atractivas y tiene aplicación en muchos sistemas de comunicación satelitales y espaciales. El algoritmo de Viterbi fue introducido en el año de 1968 por Andrew J. Viterbi.

Uno de los principales objetivos de esta tesis fue satisfacer los requerimientos del estándar IEEE 802.11<sup>a</sup> WLAN, diseñado para proveer altas velocidades en redes de comunicaciones inalámbricas en un corto rango. El estándar tiene una amplia variedad de altas tasas de datos disponibles hasta 54 Mbps; donde es obligatorio para todos los productos tener tasas de 6, 12, 24, dependiendo de los parámetros seleccionados para el sistema. Para alcanzar estas tasas de datos se aplican diferentes combinaciones de técnicas de modulación y tasas de codificación. Particularmente, para este trabajo se usa una tasa de codificación de  $\frac{1}{2}$  donde la tasa de bits que se debe alcanzar es de 48 Mbps.

En este proyecto se construyó un decodificador Viterbi con entrada de decisión dura y salida de decisión suave, con una Longitud Restrictiva de 7, lo que equivale a 64 estados. La implementación tuvo dos objetivos: ser realizada en software y en hardware. Para el primer objetivo, primeramente se diseñó e implementó un procesador MIPS de 32 bits. Luego, la codificación en lenguaje C del algoritmo de Viterbi, permitió un análisis de los diferentes parámetros del estándar y a partir de este código se obtuvo el conjunto de instrucciones de MIPS R2000 que serían ejecutadas y verificadas sobre el procesador. Para la implementación en hardware, se codificó el algoritmo de Viterbi en el lenguaje de descripción de hardware VHDL.

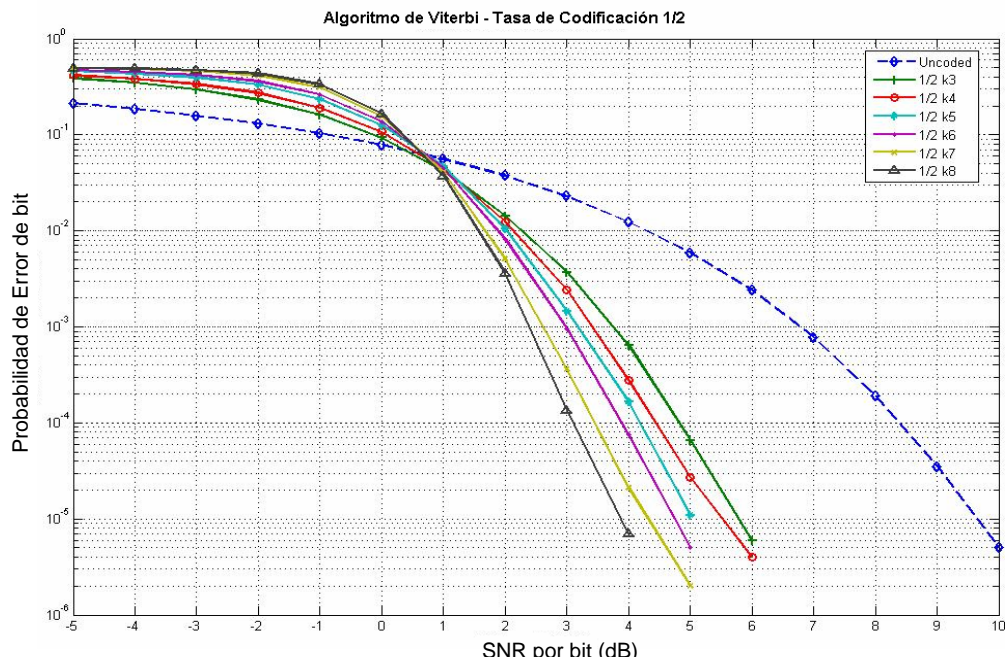
Este proyecto de Tesis fue patrocinado en su totalidad por el Centro de Diseño Electrónico –CDE- del Instituto Tecnológico y de Estudios Superiores de Monterrey –ITESM- Campus Guadalajara.

## **CONTENIDO**

### **Obtención de las curvas de BER en Matlab para diferentes parámetros de un canal de comunicación inalámbrica con codificación de canal**

Para comprender mejor el comportamiento de algoritmo de Viterbi y de las capacidades de corrección según sus características, se obtuvieron varias curvas de probabilidad de error a partir del desarrollo en lenguaje C de un canal de comunicación inalámbrico parametrizable con codificación de canal; con el fin de analizar y comprobar las diferentes ventajas y desventajas tanto en potencia y en complejidad que se obtienen cuando se utiliza codificación de canal.

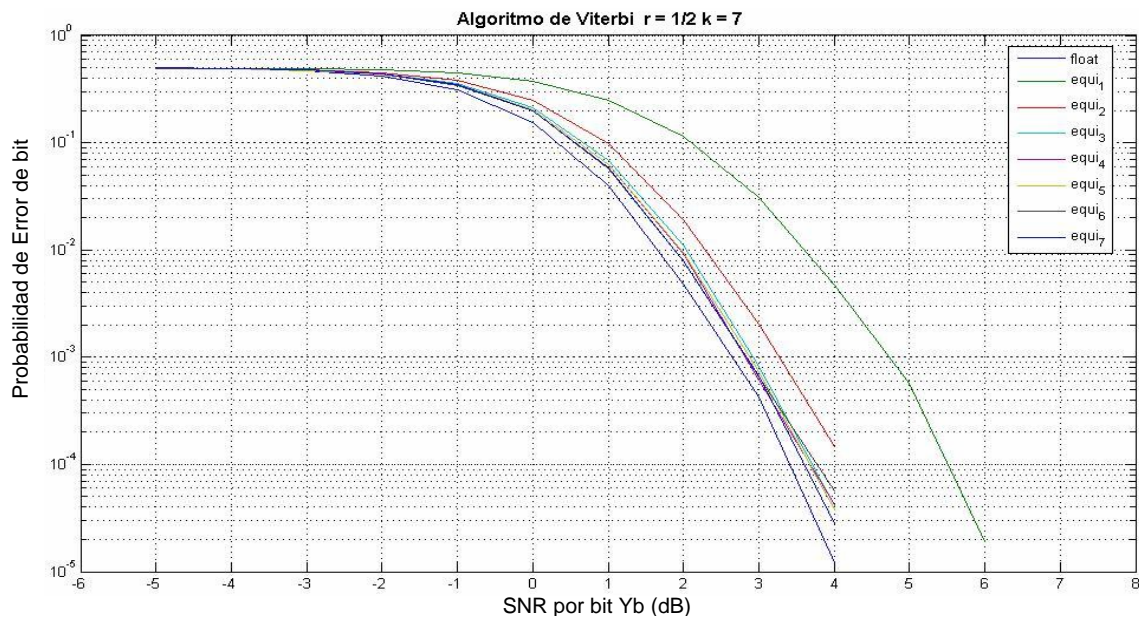
La figura 1 muestra la probabilidad de error para una secuencia de entrada de 1 millón de datos, un decodificador de Viterbi con punto flotante, tasa de codificación  $\frac{1}{2}$  para diferentes longitudes restrictivas desde  $K=3$  hasta  $K=8$ . La línea punteada indica una transmisión sin codificación de canal.



**Figura 1.** Decodificador Viterbi,  $r = \frac{1}{2}$

Entre las observaciones que podemos destacar es la ganancia de probabilidad de error de una señal transmitida sin codificar con respecto a las codificadas para un rango de 0dB a 10dB. También podemos decir como a medida que se aumenta K la curva se compacta más logrando mejor probabilidad de error<sup>1</sup>.

La relación de ganancia de ruido con la cantidad de bits cuantificación fue otro análisis importante que se hizo usando este programa. Si comparamos diversos decodificaciones para 1,2,3,4,5,6,7 bits de cuantificación ( $Q=2,4,8,16,32,64,128$ ) se observa en la figura 2 que logramos la peor relación de BER cuando solo se usa 1 bit de cuantificación con respecto la curva de punto flotante o ideal y a medida que aumentamos esta cuantificación la curva se cierra más obteniendo mejor ganancia hasta una diferencia deseable de 0,5dB con respecto a la original. A partir de  $Q=32$  casi todas las curvas se cierran obteniendo cada vez menos margen de ganancia favorable con respecto a la complejidad de diseño y de espacio que involucra hacer una implementación con mayor índice de cuantificación.



**Figura 2.** Curvas de diferente Cuantización

El diseño del decodificador Viterbi tanto en software como en hardware fue realizado para el caso de Longitud Restrictiva  $K=7$  y tasa de codificación  $r=1/2$ . Este opera con decisión suave de 3 bits de entrada.

La implementación en software se realizó sobre un prototipo de procesador MIPS, el cual también fue diseñado e implementado sobre un FPGA para la consecución del proyecto.

### **Resultados de la implementación del decodificador Viterbi sobre un procesador MIPS (Implementación en Software)**

El propósito fundamental de evaluar el desempeño del MIPS es investigar cuanto sería su máxima velocidad de operación es decir la cantidad de instrucciones ejecutadas por segundo.

El procesador MIPS fue sintetizado usando tres herramientas de desarrollo de descripción de hardware: Xilinx, Altera y Mentor. De Xilinx se sintetizó para dos tarjetas de desarrollo a nuestra disposición: Spartan 3 y Spartan 3E y de Altera Cyclone II.

La tabla I nos muestra el resultado de síntesis para las diferentes herramientas de desarrollo de hardware. Aunque el factor espacio es importante, el factor tiempo es el que más relevancia tiene para saber hasta que velocidad puede correr el procesador

MIPS. El resultado nos muestra que la tarjeta Spartan 3 nos entrega la mayor frecuencia de operación.

La medición de tiempo de todo el proceso de decodificación de una trama de 256 bits se presenta en la figura 3. que nos indica que se lleva un total de 26.4634 milisegundos de ejecución en terminar el proceso, es decir 1'323.170 ciclos de reloj.

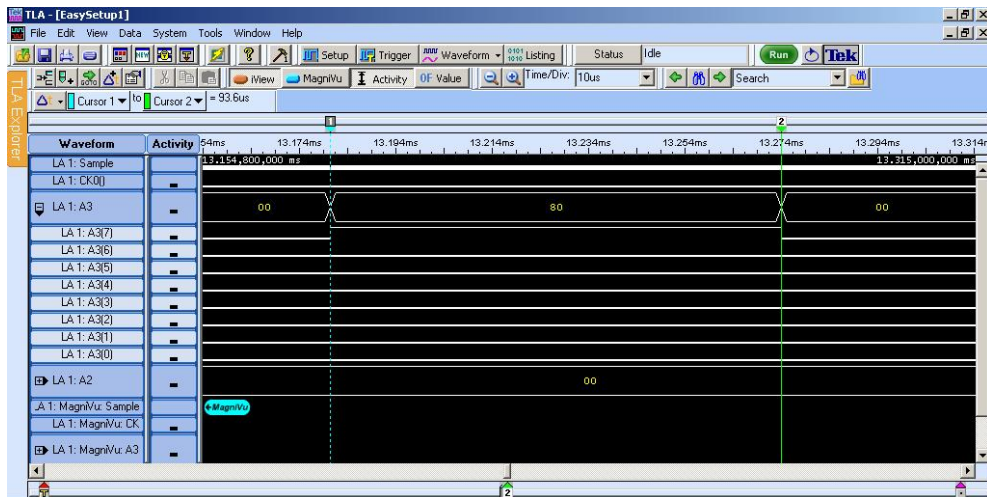


Figura 3. Tiempo total de ejecución del decodificador

**Tabla I.** Resultado de síntesis para el procesador MIPS

Herramienta de Síntesis	<i>Quartus</i>			<i>ISE</i>			<i>Mentor Graphics</i>		
FPGA	Recursos		Máxima Frecuencia	Recursos		Máxima Frecuencia	Recursos		Máxima Frecuencia
	LUTs	I/O	MHz	LUTs	I/O	MHz	LUTs	I/O	MHz
<i>Cyclone II</i> EP2C35F672C6	68%	44%	64,41	-	-	-			
<i>Spartan3</i> xs3S200FT256	-	-	-	30%	80%	270.64	28.52%	79.77%	247.709
<i>Spartan3E</i> xs3S500EFG320	-	-	-	14%	12%	256.54	11.83%	78.41%	133.708

Una vez analizado los tiempos y un correcto desempeño del algoritmo se procedió a realizar una cama de prueba que consiste en enviar por medio de una UART un archivo de imagen codificado convolucionalmente con diferentes niveles de ruido para medir la capacidad de corrección del algoritmo implementado.

Un programa realizado en C permite recibir los bits decodificados y comparar estos datos con la secuencia original y obtener una curva real de probabilidad de error en base al desempeño del decodificador implementado en el procesador MIPS.

Con estos archivos decodificados también podemos reconstruir las diferentes imágenes y observar si existen irregularidades que se dan en la figura para los diferentes niveles de ruido. Por ejemplo la figura 4 nos muestra una imagen decodificada con 5dB y otra con 0 dB.



**Figura 4.** Imagen decodificada con 5dB y 0dB

### **Resultados de la implementación del decodificador Viterbi en VHDL**

El código en VHDL del decodificador fue sintetizado utilizando las herramientas de síntesis: ISE 7.1 y 8.2i de Xilinx, QuartusII de Altera, y Mentor Graphics; y fue probado sobre cuatro familias diferentes de FPGAs, para permitir comparaciones entre diferentes plataformas de implementación. Además fue simulado en en ModelSim SE XX. La cama de pruebas para la verificación fue realizada por medio de una interacción serial entre una computadora y el FPGA.



La tabla II muestra los resultados de síntesis de la implementación completa en VHDL del decodificador Viterbi. Se observa que la máxima frecuencia para el decodificador está dada en la familia Spartan3 y es de 104.896Mhz, mientras que el FPGA Cyclonell EP2C35F672C6 presenta el menor porcentaje de uso de los recursos de hardware.

A partir de estos resultados se puede resaltar que esta implementación satisface y supera el requerimiento de velocidad del decodificador según el estándar IEEE 802.11a, la cual es de 48 Mbps de Bit Rate.

Los FPGAs: Cyclonell, Spartan3 y Spartan3E tiene una frecuencia de reloj de 50MHz, mientras que el VirtexIIPro es de 100MHz. Así, el decodificador Viterbi fue verificado para un FPGA de la familia de los VirtexIIPro, específicamente el equipo XC2VP30-7ff896 a una velocidad de grado 7.

En promedio un bit decodificado se obtiene luego de 258 pulsos de reloj lo que equivale a 2,58 microsegundos.

**Tabla XVIII.** Resultados de Síntesis del Decodificador Viterbi en VHDL

Herramienta de Síntesis	<i>Quartus</i>				<i>ISE</i>				<i>Mentor Graphics</i>		
FPGA	Recursos			Máxima Frecuencia	Recursos			Máxima Frecuencia	Recursos		Máxima Frecuencia
	LUTs	I/O	BRAM	MHz	LUTs	I/O	BRAM	MHz	LUTs	I/O	MHz
<i>Cyclone II</i> EP2C35F672C6	17 %	3 %	10 %	87.84	-	-	-	-	23.37%	2.74%	98.551
<i>Spartan3</i> xs3S200FT256	-	-	-	-	161%	7%	50%	104.896	152.27%	6.94%	61.935
<i>Spartan3E</i> xs3S500EFG320	-	-	-	-	66%	5%	30%	83.785	62.79%	6.82%	49.324
<i>Virtex2Pro</i> xc2vp307ff896	-	-	-	-	21%	2%	2%	163.825	-	-	-

## **CONCLUSIONES**

En esta tesis se presenta el diseño e implementación en software y en hardware de un decodificador Viterbi que cumple con el estándar IEEE 802.11a WLAN. El prototipo de un procesador MIPS, diseñado e implementado sirvió de plataforma para la programación en software del decodificador Viterbi. La implementación en VHDL del decodificador de Viterbi propuesto en este trabajo cumple con el requerimiento de velocidad del estándar IEEE 802.11a, aunque con un mayor porcentaje de recursos utilizados que la implementación en software.

La implementación de este proyecto ha sido por demás una experiencia enriquecedora, que nos permitió poner en práctica los conceptos adquiridos en la etapa universitaria, adquirir muchos nuevos y relacionar la teoría con la práctica. Además nos permitió interactuar y aprender con profesionistas destacados de la industria del diseño electrónico en Guadalajara, México.

## **BIBLIOGRAFÍA Y REFERENCIAS**

1. R. Villafuerte, L. Oñate, N. Ramírez. "Diseño e implementación en software y hardware de un decodificador de Viterbi para el estándar IEEE 802.11a WLAN" (Tesis, Facultad de Ingeniería en Electricidad y Computación, Escuela Superior Politécnica del Litoral, 2007).
2. J. G. Proakis, Digital Communications (4ta edición, McGraw-Hill International Editions, 2001)
3. R. Johannesson, K. Zigangirov, Fundamentals of Convolutional Coding (IEEE Press, Piscataway, New Jersey, 1999).
5. R. Rizwan, A. Menouni, Reconfigurable Viterbi Decoder for Mobile Platform (Sophia Antipolis, France, Institute Eurecom, 2005)
6. Wikipedia, Febrero 2007, Introducción a Códigos Convolutivos, [http://en.wikipedia.org/wiki/Convolutional\\_code](http://en.wikipedia.org/wiki/Convolutional_code)
7. Ingeniería en Ciencias de Sistemas y Automática, Universidad de Valladolid, Codificación de canal y conceptos generales, <http://www.isa.cie.uva.es/proyectos/codec/teoria1.html>
8. D. Patterson, J. Hennessy, Computer Organization and Design: The Hardware/Software Interface (3ra. Edición, Morgan Kaufmann, 2003).