

Codificador RS(255,k) en *hardware* reconfigurable orientado a radio cognitivo¹

RS Decoder (255,k) in Reconfigurable Hardware Oriented Towards Cognitive Radio²

Codificador RS (255,k) em *hardware* reconfigurável orientado ao rádio cognitivo³

Cecilia Esperanza Sandoval-Ruiz⁴
Antonio Fedón-Rovira⁵

SICI: SICI: 0123-2126(201301)17:1<77:CRSEHR>2.0.TX;2-F

¹ Fecha de recepción: 20 de octubre de 2011. Fecha de aceptación: 11 de septiembre de 2012. Este artículo se deriva de un proyecto de investigación denominado *Optimización del consumo de potencia en el diseño de codificadores de canal*, desarrollado en el área de postgrado de la Universidad de Carabobo y en el grupo de Investigación en Tecnologías Digitales Aplicadas a Telecomunicaciones GITDAT-UNEFA, Maracay, Venezuela.

² Reception date: October 20th 2011. Admission date: September 11th 2012. This article originated from a research project titled *Optimización del consumo de potencia en el diseño de codificadores de canal*, carried out by the Applied Digital Technology and Telecommunications Research Group (GITDAT) of the Universidad de Carabobo and the Universidad Nacional Experimental de las Fuerzas Armadas in Maracay Venezuela.

³ Data de recepção: 20 de outubro de 2011. Data de aprovação: 11 de setembro de 2012. Este artigo origina-se do projeto de pesquisa denominado *Optimización del consumo de potencia en el diseño de codificadores de canal* [*Otimização do consumo de potência no projeto de codificadores de canal*], desenvolvido pelo Grupo de Investigación en Tecnología Digital Aplicada a Telecomunicaciones (GITDAT) [Grupo de Pesquisa em Tecnologia Digital Aplicada a Telecomunicações], da Universidad de Carabobo e da Universidad Nacional Experimental de las Fuerzas Armadas, Maracay, Venezuela.

⁴ Ingeniera electricista, Universidad de Carabobo, Venezuela. Magister en Ingeniería Eléctrica, Universidad de Carabobo. Docente investigadora, Universidad Nacional Experimental de las Fuerzas Armadas, Maracay, Venezuela. Correo electrónico: cecisandova@yahoo.com.

⁵ Ingeniero electricista, Universidad de Carabobo, Valencia, Venezuela. Magister en Ingeniería Eléctrica, Universidad de Carabobo. Doctorando en Ingeniería, Universidad de Carabobo. Docente de la Universidad de Carabobo. Correo electrónico: afedon@uc.edu.ve.

Resumen

Este artículo presenta la configuración paramétrica de un codificador Reed Solomon, mediante lenguaje descriptor de *hardware* VHDL, orientado a aplicaciones de radio cognitivo, sobre dispositivos FPGA, los cuales soportan la reconfiguración del *hardware*. A través de un módulo de selección de parámetros diseñado en VHDL y una arquitectura modular, con concatenación de etapas y señales habilitadoras, se permite configurar en el *hardware* el número de símbolos de información en los RS(255, k), pues son codificadores ampliamente manejados en diversos protocolos de comunicación. En el diseño del codificador, se estableció un modelo basado en la arquitectura de sus componentes; se realizaron las simulaciones y la estimación del consumo de recursos, ofrecidos por la herramienta ISE 11 de Xilinx, y se estudiaron los esquemáticos resultantes, con lo cual se validó el desempeño y profundidad lógica del circuito desarrollado. Así se obtuvo un diseño reconfigurable basado en un modelo de habilitación de etapas, lo que ofrece una alta eficiencia en cuanto a recursos de síntesis.

Palabras clave

Codificador Reed Solomon, reconfigurable, VHDL, FPGA, radio cognitivo.

Abstract

This paper presents the parametric configuration of a Reed Solomon decoder through VHDL hardware description language, oriented towards cognitive radio applications, on FPGA circuits, which support the reconfiguration of the hardware. Using a parameter selection module designed in VHDL and modular architecture, with phase concatenation and enabling signals, it is possible to configure in the hardware the number of information symbols in the RS(255, k 's), given that such decoders are widely used in different communication protocols. In the decoder design a model was established based on the architecture of its components; we carried out simulations and the estimation of resource consumption, enabled by the ISE 11 Xilinx tool, and we studied the resulting schematics, with which we were able to validate the performance and logic of the created circuit. In this way we obtained a reconfigurable design based on a model of phase enabling, which offers a high efficiency rate regarding synthesis resources.

Keywords

Reed Solomon decoder, reconfigurable, VHDL, FPGA, cognitive radio.

Resumo

Este artigo apresenta a configuração paramétrica de um codificador Reed Solomon mediante linguagem descritora de hardware VHDL, voltada para aplicações de rádio cognitivo, sobre dispositivos FPGA, os quais suportam a reconfiguração do *hardware*. Por meio de um módulo de seleção de parâmetros projetado em VHDL e de uma arquitetura modular, com concatenação de etapas e sinais habilitadores, é possível configurar no *hardware* o número de símbolos de informação nos RS(225, k), pois são codificadores amplamente manejados nos diversos protocolos de comunicação. No projeto do codificador, estabeleceu-se um modelo baseado na arquitetura de seus componentes. Realizaram-se as simulações e a estimativa do consumo de recursos, oferecidos pela ferramenta ISE11 de Xilinx. Estudaram-se também os esquemáticos resultantes que validaram o desempenho e a profundidade lógica do circuito desenvolvido. Obteve-se um projeto reconfigurável baseado em um modelo de habilitação de etapas, o que oferece uma alta eficiência no tocante a recursos de síntese.

Palavras chave

Codificador Reed Solomon, reconfigurável, VHDL, FPGA, rádio cognitivo.

Introducción

El diseño de *hardware* con parámetros ajustables se puede orientar hacia aplicaciones en sistemas de comunicaciones bajo el paradigma de *radio cognitivo* (Fette, 2009). Un concepto que representa un antecedente importante en el área es *software radio* (SDR, por su sigla en inglés), el cual comprende funciones de comunicación implementadas por *software*, es decir, la posibilidad de definir por *software* la interfaz de comunicación que normalmente se refiere al transmisor y al receptor del equipo de comunicaciones.

Esta característica de la definición de la interfaz vía *software* implica el uso de procesadores de señal digital para reemplazar el *hardware* dedicado (Tonfat, 2008). En el contexto de SDR, la parametrización permite disminuir el tamaño del *software* que se va a descargar durante la configuración del *hardware* (Alaus, Noguet y Palicot, 2008). El procesamiento digital, el cual busca la reducción de componentes analógicos, está a cargo de una tecnología que permite la reconfiguración del sistema. Esta etapa puede implementarse a través de dispositivos de procesamiento de señales (DSP) o *Field Programmable Gate Array* (FPGA). Estos últimos fueron los seleccionados para esta investigación.

En el momento de diseñar las etapas de un sistema de comunicación orientada a radio cognitivo, se debe tener presente la necesidad de que estas cumplan con los requerimientos característicos de SDR, para el soporte de la tecnología de radio cognitivo a través de los componentes descritos. En tal sentido, entre los criterios de diseño se establece la capacidad de definir por *software* los parámetros asociados al estándar de comunicación, sobre una arquitectura flexible programada. En el área de SDR, las técnicas de parametrización son un tema de investigación de interés (Alaus, Noguet y Palicot, 2008). Estas técnicas se consideraron para la presente investigación.

Entre los trabajos desarrollados en el área de investigación de radio definido por *software*, que indica una tendencia de la aplicación de estos conceptos en módulos de comunicación reconfigurables, punto clave para el soporte de radio cognitiva, se pueden mencionar el de Alaus (2008), en el cual se trata el circuito *Linear*

Feedback Shift Register (LFSR), que puede llevar a cabo operaciones sobre señales para aplicaciones de radio cognitiva en generación de secuencias pseudoaleatorias, convoluciones, codificación cíclica, codificación Reed Solomon, entre otras, comparando la aplicación de dicho operador LFSR con los enfoques clásicos.

Los códigos Reed Solomon son uno de los más importantes en la corrección de errores (Flocke, 2005). En desarrollos de codificadores Reed Solomon descritos bajo el estándar *VHSIC Hardware Description Language* (VHDL) (Sandoval y Fedón, 2007; Sandoval, 2008 y 2012) se presentan descripciones particulares de codificadores específicos, y en este trabajo se busca generalizar el desarrollo con el modelo parametrizado. Para el desarrollo del codificador que acá se propone es importante contar con alternativas de decodificación que manejen el enfoque adaptativo de parámetros (Atieno, 2006) y cuyos componentes sean parametrizables en VHDL para mantener la compatibilidad. En tal sentido (Flocke, 2005) optimiza el diseño básico del decodificador Reed Solomon, a través del uso compartido de recursos y *pipelining*, con lo que el número de elementos de procesamiento utilizado logra disminuir significativamente. Ello da lugar a una aplicación de alto rendimiento en que el camino crítico, en cuanto a tiempo de procesamiento, está dado por la unidad de multiplicación.

Con base en lo anterior, se plantea un compromiso entre el rendimiento en función de la velocidad de procesamiento y el consumo de recursos, para lo cual se presenta la parametrización de los símbolos de información, que brindan la posibilidad de implementar un número de etapas idénticas del elemento LFSR en función del parámetro k . Por su carácter genérico, esta etapa podría reutilizarse en caso de que se requiera ahorrar recursos. Por otra parte, el componente multiplicador se ha implementado de forma paralela, con el propósito de optimizar la velocidad, obteniendo un equilibrio entre los factores de rendimiento tenidos en cuenta.

En consideración de los avances previos, se ha enfocado la presente investigación hacia el diseño de un codificador Reed Solomon descrito en lenguaje VHDL, para su implementación sobre dispositivos que soporten la reconfiguración dinámica del *hardware* diseñado. En este caso se emplea esta propiedad para el ajuste de parámetros del codificador $RS(255, k)$, a través de un módulo de selección diseñado en VHDL y una arquitectura modular, con concatenación de etapas y señales habilitadoras. Ello permite configurar en el *hardware* el número de símbolos de información k de los mencionados codificadores, haciendo del codificador diseñado un módulo reconfigurable de acuerdo con las condiciones de selección.

Entre los objetivos de la investigación están interpretar el comportamiento del codificador Reed Solomon, reconocer las estructuras generales en la arquitectura del codificador (para la descripción en función de parámetros en VHDL de los componentes) y analizar la síntesis de los codificadores para la estimación de consumo de recursos y profundidad lógica del diseño en función de los esquemas *Register-Transfer Level* (RTL).

El diseño del codificador se ha establecido bajo una metodología modular, con el fin de reducir el tiempo de desarrollo y ofrecer el código en VHDL del diseño para futuras aplicaciones en las cuales los diseñadores puedan optimizar, combinar o ampliar el diseño a través de la filosofía de *hardware* libre (Cenditel, 2012).

1. Hardware reconfigurable en radio cognitivo

En el momento de diseñar módulos reconfigurables para las etapas de un sistema de comunicación es necesario revisar un conjunto de conceptos asociados, entre ellos el de SDR, el cual es un conjunto de técnicas que permiten la reconfiguración de un sistema de comunicaciones sin la necesidad de cambiar ningún *hardware* del sistema (Blust, 2002). En este los parámetros pueden ser controlados dinámicamente, y este constituye el componente clave en la implementación de radio cognitivo, donde el sistema SDR puede percibir las condiciones del entorno para ajustar los módulos al esquema más apropiado, a fin de lograr un eficiente desempeño del sistema de comunicaciones y la mejor utilización del espectro en sistemas inalámbricos. Estas características lo definen como un sistema de comunicación inteligente.

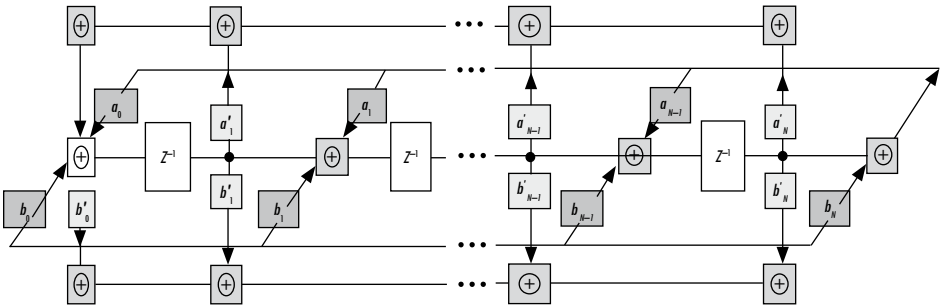
Para la implementación de estos conceptos se requieren dispositivos lógicos reconfigurables en *hardware*, sobre los cuales el diseñador maneje de forma paramétrica la configuración de la arquitectura, a través del lenguaje VHDL. De estos conceptos podemos observar que el diseño se aplica a los dispositivos capaces de mantener su funcionamiento sin interrupción, mientras varias partes de la matriz lógica se están configurando o se habilitan de forma selectiva, lo que en nuestro caso de estudio corresponde al ajuste de los parámetros del codificador RS(n,k) sobre el dispositivo FPGA. En tal sentido, se han considerado las tendencias hacia diseños reconfigurables de forma dinámica utilizando FPGA (Astarloa, 2005).

Una de las técnicas para la implementación de SDR está basada en la coexistencia de todos los módulos. Se define la arquitectura de soporte para los diversos modos y se conmuta para seleccionar el que se requiere (Fitzek, 2006). En esta técnica se debe analizar el efecto en la eficiencia respecto al costo y tamaño del

diseño. En atención a la eficiencia se propuso una técnica basada en compartir el módulo de *hardware* diseñado en funciones comunes de diversas aplicaciones del sistema (Jondral, 2002), para, finalmente, encontrar entre las técnicas de parametrización el diseño de módulos de *hardware* que comparten funciones y operadores (Al Ghouwayel, Louet y Palicot, 2006), con lo que un mismo operador se puede configurar para implementar funciones en diferentes aplicaciones.

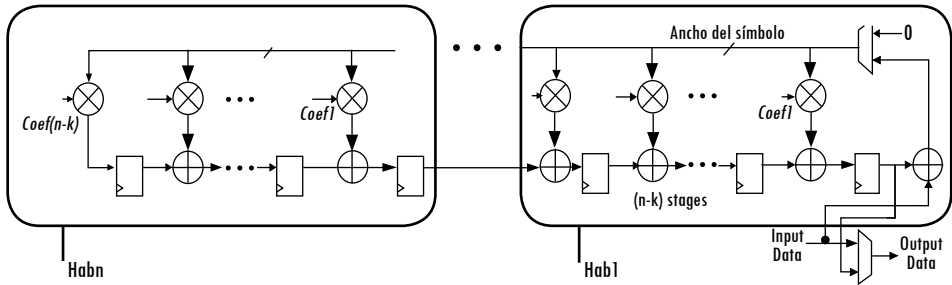
En trabajos previos, Alaus, Noguét y Palicot (2008) presentan las técnicas de factorización de las funciones y operaciones comunes para la parametrización de etapas del R-LFSR (*Reconfigurable-Linear Feedback Shift Register*), donde se enfatiza en que la reconfiguración se logra, a menudo, a costa de la lógica de configuración adicional y en que la ventaja de factorización debe analizarse cuidadosamente, por lo que varias arquitecturas basadas en esta estructura R-LFSR se estudian en términos de complejidad, con el fin de seleccionar la mejor aplicación. Estas consideraciones se han tenido en cuenta para la propuesta del codificador Reed Solomon, en cuyo componente principal está el módulo del generador de los símbolos de redundancia un LFSR, que deberá ser parametrizado en función del número de símbolos de información k , para su diseño reconfigurable. En la figura 1 se presenta el R-LFSR.

Figura 1. Arquitectura del R-LFSR con operadores comunes



Fuente: (Alaus, Noguét y Palicot, 2008).

Dentro de su arquitectura, el codificador Reed Solomon presenta un elemento circuital que coincide con el modelo del LFSR, en el generador de los símbolos de redundancia, donde se puede identificar como elementos comunes las operaciones XOR y los elementos de memoria. Adicionalmente, cuenta con elementos multiplicadores en campos finitos de Galois, los cuales son los responsables de obtener los datos parciales que se van a operar en las compuertas XOR, como se muestra en la figura 2.

Figura 2. Arquitectura del codificador RS(n,k) con habilitación de etapas

Fuente: presentación propia de los autores.

De manera tal que, realizando una reestructuración del circuito, podemos intercalar un componente multiplicador, a fin de emplear el LFSR reconfigurable como modelo base del generador de redundancia del codificador. En este caso se emplea la definición de parámetros para manejar la estructura y los coeficientes por etapas en función del parámetro k .

2. Diseño del codificador RS(255,k)

Inicialmente, se consideró el desarrollo de un codificador RS con la longitud del código fijo para establecer el modelo en VHDL (Sandoval, 2007). A este diseño se le desarrollaron versiones extendidas, a través de las cuales se ha ampliado la longitud del código para 255 símbolos de 8 bits, donde se ha expresado el parámetro k como un valor genérico, el cual puede ser definido en tiempo de configuración.

Seguidamente, se ha propuesto desarrollar el codificador RS(255, k) reconfigurable, estableciendo el valor de k apropiado a través de una entrada de selección, que sería suministrada por la etapa cognitiva del sistema. Para el diseño de la arquitectura del codificador se ha considerado la alternativa de utilizar habilitadores (tristados) para el manejo de las etapas. De esta manera obtenemos un codificador de arquitectura reconfigurable, el cual es el insumo fundamental para el diseño bajo el enfoque de radio cognitivo.

El parámetro k permite fijar el número de etapas, $255-k$, siendo k la entrada para el módulo de configuración. La arquitectura del generador de redundancia, implementado con el LFSR se definió de forma modular, en función del número de etapas correspondientes para generar los símbolos de redundancia. En este caso se han considerado valores específicos de acuerdo con los codificadores estándar empleados, donde se establecen 8, 16, 24 y 32 símbolos de redundancia

para las pruebas del diseño, que describen el componente LFSR reconfigurable, de ocho etapas.

Este componente puede ser habilitado de forma selectiva según la configuración requerida hasta un total de cuatro etapas LFSR. De esta manera, se calcularon los recursos de *hardware* utilizados según el parámetro k . Los elementos de memoria (*flip-flops*) requeridos por el diseño determinan el número de *Slice* por uso de elementos de memoria FF. Este depende del número de etapa de memoria del LFSR en el codificador $n-k$ y del número de bits por símbolos $b=8$, como se presenta en la tabla 1.

Tabla 1. Estimación de los recursos hardware en función de k con símbolos de 8 bits

n	k	hab_4	hab_3	hab_2	hab_1	LFSR activos	Etapas $n-k$	Reg. FF
255	223	1	1	1	1	4	32	256
255	231	0	1	1	1	3	24	192
255	239	0	0	1	1	2	16	128
255	247	0	0	0	1	1	8	64

Fuente: presentación propia de los autores.

Para la programación se empleó el *software* ISE 11 de Xilinx. Se inició por la definición del módulo de configuración, el cual será el responsable de generar los habilitadores para configurar la arquitectura del codificador, ya que estos dependen de la configuración $RS(255,k)$, esto a través de lenguaje descriptor de *hardware* VHDL. Por medio de este módulo se establecen los parámetros que serán entrada para el codificador en función del estándar seleccionado por medio del número de símbolos de información k . Así, una vez establecidos los habilitadores, la arquitectura del codificador quedará ajustada a las etapas correspondientes.

El polinomio generador de redundancia se implementó sobre el LFSR configurable. Los coeficientes asociados están fijados en la descripción del codificador, al asignar los puertos del componente multiplicador, bajo la sintaxis *port map*. Estos coeficientes serán empleados por el codificador para el cálculo de los símbolos de redundancia. Es de hacer notar que se han asignado habilitaciones para grupos de multiplicadores y que los restantes quedaron sin implementar, para evitar la asignación paramétrica de los coeficientes, lo que permite optimizar el diseño para disminuir el número de señales, planteamiento presentado en la arquitectura modular (figura 2).

Los componentes multiplicadores se diseñaron estableciendo un modelo del LFSR concurrente (Sandoval, 2010), el cual emplea como base el operador

concatenación. De esta manera se generan de forma combinacional los arreglos binarios a la salida de la operación $\text{mod}_P(x)$ y se logra una optimización significativa. Estos multiplicadores permiten obtener el resultado de la operación de forma instantánea, manejando un modelo de arreglos parciales que coexisten en el mismo instante de tiempo. De esta manera se calculan los productos, para que sean operados por las compuertas XOR, que son descritas en el LFSR reconfigurable del codificador. Los coeficientes empleados están asociados al valor del parámetro k ; entre tanto, el número de multiplicadores operativos está dado por el número de etapas activas, habilitados por hab_i .

3. Análisis de resultados

Se realizó la simulación del codificador Reed Solomon (255, k), mediante la herramienta ModelSim XE III 6.3c, que permitió validar el funcionamiento del diseño para $k = 247, 239, 231, 223$, respectivamente, comprobando los valores teóricos. En la figura 3 se presentan los resultados de la simulación para el caso de $k = 239$.

Figura 3. Resultados del codificador RS (255,k) con $k = 239$

d_in	X240	X241	X242	X243	X244	X245	X246	X247	X248	X249	X250	X251	X252	X253	X254	X255
k	239															
dk	[Pulsos de reloj]															
hab	[Pulsos de habilitación]															
salida	X1	X126	X147	X48	X155	X224	X3	X157	X29	X226	X40	X114	X61	X30	X244	X75

Fuente: presentación propia de los autores.

En el análisis de la implementación circuital del codificador se pueden observar los componentes descritos en el diseño, a través de los esquemas RTL generados por la herramienta de desarrollo del ISE11. De esta manera se comprueba la arquitectura descrita. En la figura 4 se presenta una sección del LFSR reconfigurable, donde los elementos de memoria son habilitados por la señal hab_i dispuesta para la configuración de cada etapa LFSR.

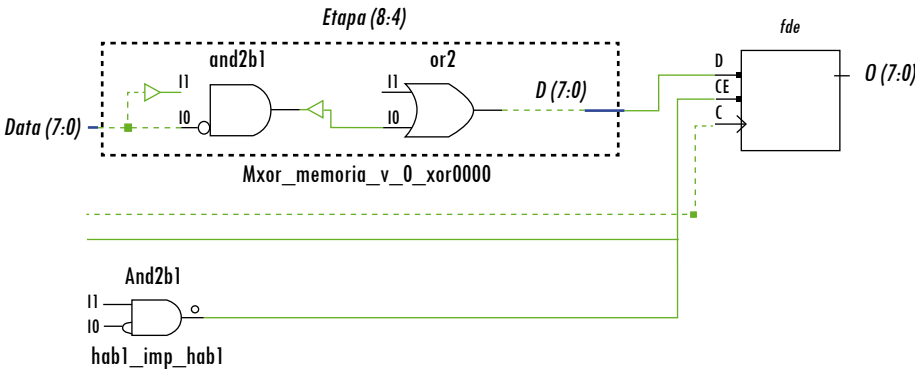
Como se puede observar, la señal de habilitación hab_i es conectada a los habilitadores de las etapas (figura 4) en todo el hardware, y aun cuando en la descripción se emplean instrucciones condicionales, su implementación no obedece a la comparación de una condición en el software, por lo que se ha tratado como una señal del circuito, pues permite manejar la configuración del diseño. El módulo de configuración genera las habilitaciones, con el propósito de que las etapas no implementadas para cierta configuración no consuman recursos del dispositivo; de esta manera se logra optimizar el consumo de potencia y quedan

deshabilitados los componentes que constituyen dichas etapas. A partir de la entrada k , se establecen los habilitadores en sintaxis VHDL. Para generar esta configuración en el caso de $k = 239$ se describe, *when "11101111" => hab1 <= '1'; hab2 <= '1'; hab3 <= '0'; hab4 <= '0'*. De esta manera, la descripción realizada a través de la instrucción *case* es implementada por lógica combinacional que permite generar las señales de habilitación (en este caso *hab4*). Cuando se presenta la combinación binaria correspondiente, se presenta la *and* de los 8 bits de entrada de la señal k , como se puede observar en la figura 5.

Resulta importante el estudio de la profundidad lógica en la implementación del diseño sobre el dispositivo FPGA, puesto que este factor incide en los retardos de las señales, al igual que en el consumo de potencia por propagación de *glitches* (Sutter y Boemo, 2007). La figura 6 presenta el esquemático de la tecnología para las primitivas asociadas a la salida *datox_0*, es decir, los recursos de *hardware* en la construcción de la mencionada señal de salida. Este esquema permite observar la profundidad lógica del circuito implementado, donde se identifican tres capas de LUT.

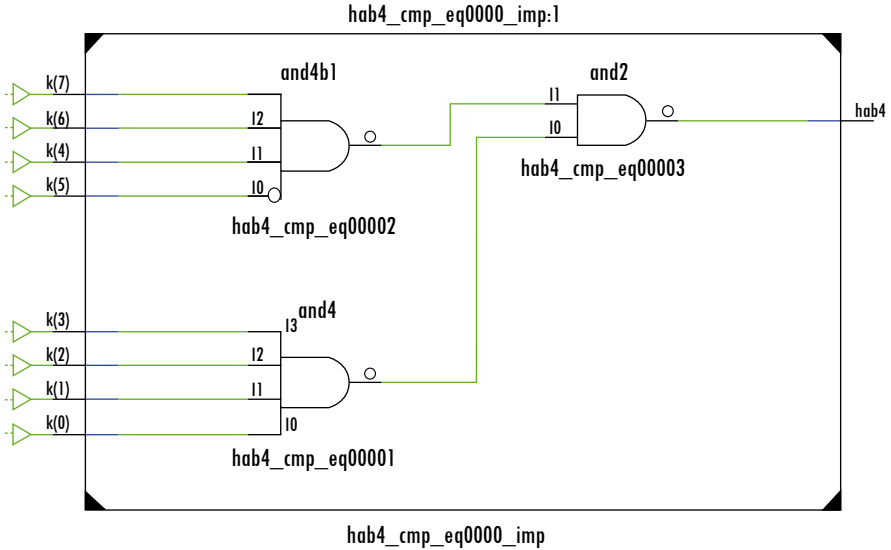
Seguidamente, se compararon los reportes de síntesis del codificador $RS(255,k)$. En la tabla 2 se observa el consumo de recursos de estos en los casos de a) $k = 223$ y b) $k = 239$.

Figura 4. Esquemático RTL de una sección del LFSR configurable



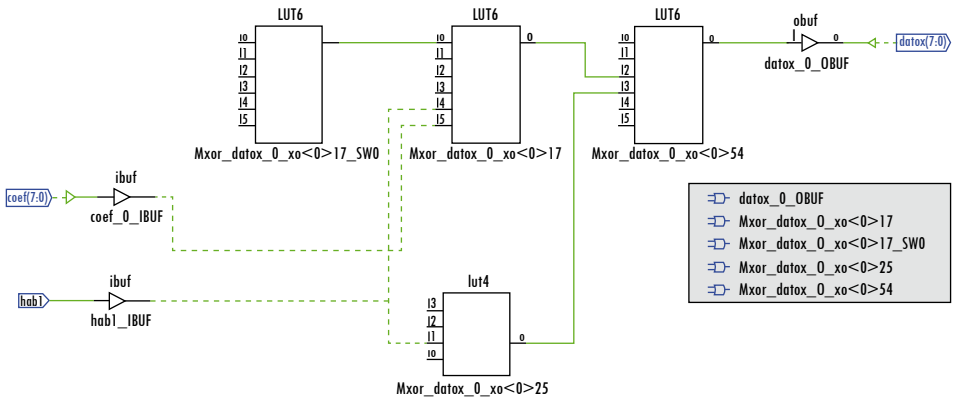
Fuente: presentación propia de los autores

Figura 5. Esquemático RTL del módulo para configuración de la arquitectura *hab4*



Fuente: presentación propia de los autores.

Figura 6. Esquemático RTL de una sección del LFSR configurable



Fuente: presentación propia de los autores

Tabla 2a. Reporte de síntesis del codificador RS(n,k): estimación de recursos utilizados para el codificador RS(255,223)

Recursos utilizados	Slice/CLB	LUT	Registros FF
Diseño paramétrico para RC	256	305	256
CCSDS (Xilinx, 2011)	337	329	316
OC-192 (Lattice, 2005)	239	435	327
Jinzhou (Jinzhou <i>et al.</i> , 2012)	-	460	278

Fuente: presentación propia de los autores.

Tabla 2b. Reporte de síntesis del codificador RS(n,k): estimación de recursos utilizados para el codificador RS(255,239)

Recursos utilizados	Slice/CLB	LUT	Registros FF
Diseño paramétrico para RC	128	162	128
ETSI (Xilinx, 2011)	199	194	186
CCSDS (Lattice, 2005)	141	235	196
2.5G (Altera, 2011)	374	40	186
Gianni (Gianni, Claudio y Corteggiano, 2007)	-	353	170

Fuente: presentación propia de los autores.

A partir de los resultados obtenidos se puede contrastar el consumo de recursos de *hardware* entre el codificador paramétrico para radio cognitivo diseñado, cuya síntesis se realizó a través del ISE11 sobre el xc5vlx30-3ff324, y los resultados reportados por desarrollos previos (Altera, 2011; Lattice, 2005; Xilinx, 2011 y 2012). Respecto al diseño previo de mayor eficiencia (Xilinx, 2011) se puede observar una reducción en el consumo de LUT del 9,5 % para el caso de $k = 223$ y del 18,6 % para $k = 239$. Por su parte, la reducción en el consumo de FF fue del 18,98 % para el caso de $k = 223$ y del 31,18 % para $k = 239$.

Para los casos estudiados, los resultados de los reportes de síntesis (tablas 2a y 2b) se obtuvo que el número de FF corresponde con $(255 - k) \times 8$ elementos de memoria, lo que coincide con los valores obtenidos en la tabla 1. Así queda demostrado que el consumo de recursos estará dado por la configuración definida a través del parámetro k . Así mismo, el número de tablas de búsqueda para la implementación de las funciones lógicas del LFSR y multiplicadores asociados a las etapas activas resulta proporcional al número de etapas implementadas, lo que permite observar que el consumo de recursos está dado por la configuración establecida, en una proporción $(n-k)/etapas_LFSR$.

Los resultados obtenidos están establecidos para un polinomio irreducible $p(x)$ y un número de bits por símbolo fijos. La eficiencia encontrada en el diseño permite proponer una extensión del tratamiento empleado, a fin de obtener una versión del codificador completamente generalizada, en cuanto a todos los parámetros que los definen.

Se analizó el reporte del consumo de potencia, empleando para ello la herramienta *XPower Analyzer*, incorporada en el ISE11, y se obtuvo un consumo de potencia dinámica de 19 mW respecto al reportado en la versión fija del codificador que fue de 18,36 mW. Se observó que el factor que presenta mayor efecto en esta corresponde a la potencia consumida en las señales del diseño, puesto que al realizar el codificador reconfigurable se ha requerido definir nuevas señales para el manejo de parámetros que en la versión previa eran fijos; sin embargo, el incremento no es significativo en relación con la flexibilidad obtenida en el diseño paramétrico orientado a radio cognitivo.

Conclusiones

En esta investigación se validó el codificador Reed Solomon (255,k) reconfigurable del código VHDL desarrollado, el cual permite el manejo modular y selectivo de las etapas del circuito LFSR, lo que optimiza el consumo de recursos, a la vez ofrece a los diseñadores una base importante para desarrollos futuros en *hardware* libre. De esta manera, se han considerado técnicas de optimización, como parametrización y modularidad de los componentes, al lograr que el dispositivo reporte un menor consumo de potencia, a través del manejo de habilitaciones para desactivar secciones del circuito. Por otra parte, se empleó la paralelización del circuito LFSR del componente *multiplicador* en campos finitos de Galois, con lo que se logró el procesado en los multiplicadores sin señal de reloj asociada *clk*.

Esta implementación se basó en un modelo concurrente que reporta una profundidad lógica apropiada para la complejidad del circuito, lo que se interpreta como un circuito que maneja retardos aceptables. Adicionalmente, se realizó un manejo de los coeficientes como señales configurables en lugar de variables; esto debido a la frecuencia de cambio de las mencionadas señales, las cuales solo serán modificadas al momento de una nueva configuración. De ser tratadas como señales reportarían un efecto en la síntesis del diseño.

Entre los aportes se puede mencionar el modelo del codificador Reed Solomon, a partir de la generalización de un caso particular. El modelo generalizado cuenta con la capacidad de configurar parámetros y habilitaciones de forma autónoma bajo condiciones del canal preconsideradas; además, puede ser reconfigurado de

forma dinámica para ajustar su arquitectura en casos de optimización requeridas del *hardware*, gracias a la modularidad y manejo de señales paramétricas en la descripción VHDL del codificador diseñado, pues estas alternativas de configuración aprovechan al máximo la flexibilidad del *hardware* y ofrecen: versatilidad al diseño, menor consumo de recursos *hardware* para la implementación del codificador RS(255,k) respecto a las referencias contrastadas y optimización del consumo de potencia dinámica y velocidad de procesamiento, a través de la habilitación de los componentes de forma paralela. Con base en estas características se puede concluir que el modelo ofrece un equilibrio entre los indicadores de eficiencia considerados.

Referencias

- AL AUS, L. A reconfigurable linear feedback shift register operator for software defined radio terminal, wireless pervasive computing. *ISWPC. 3rd International Symposium*, 2008, pp. 319-323.
- AL AUS, L.; NOGUET, D. y PALICOT, J. A reconfigurable LFSR for tri-standard SDR transceiver, architecture and complexity analysis. *Proceedings of 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools*. 2008, pp. 61-67.
- AL GHOUWAYEL, A.; LOUET, Y. y PALICOT, J. A reconfigurable architecture for the FFT operator in a software radio context. *IEEE International Symposium on Circuits and Systems*, 2006.
- ALTERA. *RS Encoder*. 2011 [documento en línea]. <<http://www.altera.com/literature/an/an642.pdf>> [consultado: 09-05-11].
- ASTARLOA, A. *Reconfiguración dinámica de sistemas modulares multi-procesador en dispositivos SoPC*. s. l.: Universidad del País Vasco, 2005.
- ATIENO, L.; ALLEN, J.; GOECKEL, D. y TESSIER, R. An adaptive Reed-Solomon error-and-erasures decoder [documento en línea]. *Proceedings ACM/SIGDA 14th international symposium on Field programmable gate arrays*. 2006, pp. 150-158. <<http://dl.acm.org/citation.cfm?id=1117224>>.
- BLUST M., S. *Software defined radio enabling technologie*. New York: Wiley, 2002.
- CENDITEL. *Proyecto hardware libre* [documento en línea], 2012. <<http://www.cenditel.gov.ve>> [consultado: 03-08-11].
- FETTE, B. *Cognitive radio technology*. 2nd. ed. Philadelphia: Academic Press-Elsevier, 2009.
- FITZEK, F. H. P. *Cooperation in wireless networks, principles and applications: real egoistic behavior is to cooperate*. New York: Kluwer, 2006.
- FLOCKE, A. *et al.* Implementación y modelado de parametrizables decodificadores de alta velocidad Reed Solomon en FPGAs. *Advances in Radio Science*. 2005, vol. 3, núm. 14, pp. 271-276 [documento en línea]. <<http://www.adv-radio-sci.net/3/271/2005/ars-3-271-2005.pdf>> [consultado: 15-09-12].

- GIANNI, P.; CLAUDIO, G. D. y CORTEGGIANO, F. Implementación en FPGA de un código Reed Solomon RS (255,239). *Actas de la Escuela Argentina de Microelectrónica, Tecnología y Aplicaciones*. 2007, vol. 3, pp. 77-81 [documento en línea]. <<http://lcd.efn.uncor.edu/pdf/pp.77-81%281%29.pdf>> [consultado: 05-03-11].
- JINZHOU, Z.; XIANFENG, L.; ZHUGANG, W. y WIEMING, X. The design of a RS encoder. *Future Computing, Communication, Control and Management*. 2012, vol. 144, pp. 87-91.
- JONDRAI, F. *Software defined radio enabling technologie: A technique for SDR Implementation*. New York: Wiley, 2002.
- LATTICE, S. C. *Dynamic block Reed-Solomon encoder*, 2005 [documento en línea]. <<http://www.latticesemi.com/lit/docs/datasheets/isp/lever/dynamic-reed-solomon.pdf>> [consultado: 16-07-11].
- SANDOVAL RUIZ, C. Codificador RS(n,k) basado en LFCS: caso de estudio RS(7,3). *Revista de la Facultad de Ingeniería de la Universidad de Antioquia*. 2012, núm. 64, pp. 45-55.
- SANDOVAL RUIZ, C. Modular programming of functions for turbo product codes on FPGA. *Revista Técnica de la Facultad de Ingeniería, Universidad del Zulia*. 2008, vol. 31, núm. 3, pp. 294-301 [documento en línea]. <<http://www.scielo.org.ve/pdf/rtfiuz/v31n3/art10.pdf>> [consultado: 09-06-09].
- SANDOVAL RUIZ, C. Multiplicador paralelo en campos finitos de Galois GF (2m) aplicado a códigos Reed Solomon con longitud ajustable sobre FPGA. *Congreso Internacional de Investigación UC*, 2010.
- SANDOVAL RUIZ, C. E. y FEDÓN, A. Codificador y decodificador digital Reed-Solomon programados para hardware reconfigurable. *Ingeniería y Universidad*. 2007, vol. 11, núm. 1, pp. 17-32 [documento en línea]. <<http://redalyc.uaemex.mx/redalyc/src/inicio/ArtPdfRed.jsp?iCve=47711102>> [consultado: 09-08-10].
- SUTTER, G. y BOEMO, E. Experiments in low power FPGA design. *Latin American Applied Research*. 2007, vol. 37, núm. 1, pp. 99-104.
- TONFAT, J. y SILVA, C. Diseño de un modulador FM basado en la tecnología software Radio. *XIV Iberchip Workshop*, México, 2008
- XILINX. *IPCore RS Encoder 7.1*, 2011 [documento en línea] <http://www.xilinx.com/support/documentation/ip_documentation/rs_encoder_ds251.pdf> [consultado: 04-03-11].
- XILINX. *IPCore RS Encoder 8.0*, 2012 [documento en línea] <http://www.xilinx.com/support/documentation/ip_documentation/rs_encoder/v8_0/pg025_rs_encoder.pdf> [consultado: 07-02-12].

