XVII SIMPOSIO DE INGENIERÍA ELÉCTRICA (SIE-2017)

DESARROLLO DE APLICACIONES PARA LA RADIO DEFINIDA POR SOFTWARE

APPLICATIONS DEVELOPMENT FOR SOFTWARE-DEFINED RADIO

José Carlos Cruz Sandoval[1], Yakdiel Rodríguez-Gallo Guerra[2]

1-José Carlos Cruz Sandoval. División Centro MoviTel. Dirección Territorial Ciego de Ávila. Cuba. E-mail:josecarloscruzsandoval1990@gmail.com
2-Yakdiel Rodríguez-Gallo Guerra. Departamento de Telecomunicaciones y Electrónica. Facultad de Ingeniería Eléctrica. Universidad Central "Marta Abreu" de Las Villas, UCLV.Santa Clara. Cuba. E-mail: yrodriguez-gallo@uclv.edu.cu

Resumen: El rápido despegue de las telecomunicaciones, de conjunto con la tendencia cada vez más actual de aprovechar las bondades computacionales que brindan diversos sistemas, ha hecho posible la creciente evolución de la tecnología de Radio Definida por Software. Una muestra de ello lo ofrece el avance que ha venido teniendo particularmente el desarrollo de los estándares de comunicaciones inalámbricas y el hecho de implementar sistemas de comunicaciones que empleen eficazmente los recursos del software para realizar las operaciones de procesamiento de señales. En el presente trabajo se realizaron modelos de transmisión de datos con técnica de espectro extendido por secuencia directa, que utilizan Registros de Desplazamiento con Realimentación Lineal, los cuáles se modelaron y simularon utilizando las herramientas de simulación MATLAB 2013 y System Generator. Al final se evidencia la funcionalidad del sistema y los beneficios de implementar un Registro de Desplazamiento con Realimentación Lineal extendido en el modelo, lo que resulta de gran utilidad como aplicación de la SDR para sistemas de comunicaciones encriptados.

Abstract: *The rapid take-off of telecommunications, in conjunction with the increasingly current tendency to take advantage of the computational benefits offered by different*

systems, has made possible the evolution of Software Defined Radio technology. An example of this is the progress that has been made particularly in the development of wireless communications standards and the implementation of communications systems that effectively use software resources to perform signal processing operations. In the present work, data transmission models using direct sequence extended spectrum technique using Linear Feedback Displacement Registers were used, which were modeled and simulated using the MATLAB 2013 and System Generator simulation tools. In the end, the functionality of the system and the benefits of implementing a Linear Feedback Displacement Register extended in the model are shown, which is very useful as an application of the SDR for encrypted communications systems.

Palabras Clave: SDR; LFSRe; Simulink; SysGen; Espectro; FPGA.

Keywords: SDR; LFSRe; Simulink; SysGen; Spectrum; FPGA.

1. Introducción

Las comunicaciones inalámbricas en el mundo actual avanzan a tal velocidad que los dispositivos de comunicaciones quedan prácticamente obsoletos poco después de su producción. Para superar este problema, los sistemas de comunicaciones deben ser diseñados para maximizar la inserción transparente de nuevas tecnologías en cualquier fase de su ciclo de vida, teniendo en cuenta que los dispositivos actualizados deben ser capaces de comunicarse entre ellos y con el resto de los mismos[11].

La Radio Definida por Software o SDR *(Software Defined Radio)* es una arquitectura flexible que es aplicable a varios estándares de radio. Ella es un sistema de radiocomunicaciones que proporciona control por software para una variedad de métodos de modulación, filtrado, operaciones de banda ancha o de banda estrecha y técnicas de espectro extendido[14].

En SDR, los procesamientos de las señales en la comunicación de radio son hechas en software en vez de hardware, convirtiéndola en una tecnología menos rígida y más configurable para su explotación y uso, con bajo precio y complejidad. Por tanto, su utilización está concebida en la esfera digital y no en la esfera analógica, como ocurre en la radio convencional.

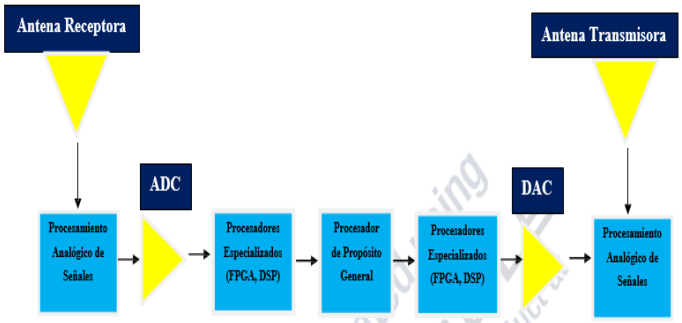

Figura 1. Arquitectura en bloques de la Radio Definida por Software. Fuente: Elaboración propia.

La complejidad creciente, el tamaño y los requisitos de actuación de las aplicaciones de la SDR, han determinado que se haya manejado en la industria la amplia utilización de los FPGA como los dispositivos primarios de procesamiento.

Un arreglo de campo de compuerta programable (FPGA) es un circuito integrado consistente en bloques interconectados llamados bloques lógicos configurables o CLB *(Configurable Logic Block)*. Los FPGAs son diseñadas para ser configuradas usando un lenguaje de descripción de hardware como VHDL o Verilog[4].

Este trabajo plantea como problemática la siguiente interrogante: ¿Qué hacer para implementar en una FPGA aplicaciones de la Radio Definida por Software utilizando herramientas de simulación? Aquí se profundiza en el tema de las aplicaciones de la Radio Definida por Software y su funcionamiento de conjunto con la plataforma de hardware FPGA, valiéndose de herramientas de simulación como MATLAB y System Generator.

2. Características de SDR

En la figura 1 se muestra una arquitectura genérica de SDR. El bloque de procesamiento analógico de señales, es donde se ubica la sección de RF y la sección de FI. Esta es la responsable de transmitir/recibir las señales de radiofrecuencia para adecuarlas y convertirlas en frecuencias intermedias, en el caso de la recepción, o amplificar y modular las señales de FI adecuándolas para la transmisión en el aire, en el caso de la transmisión. La sección de FI es la encargada de pasar la señal de FI a banda base y digitalizarla para la recepción, o pasar la señal de banda base a FI y hacer la conversión digital-analógica de la señal para la transmisión. Los encargados de la conversión analógica-digital o digital-analógica de la señal son los módulos

ADC/DAC[12].

3. Características de las FPGA

Un Arreglo de Campo de Compuerta Programable (FPGA) es un circuito integrado consistente en la interconexión de bloques llamados Bloques Lógicos Configurables o CLB *(Configurables Logic Blocks)*. Cada CLB está compuesto de slices y cada slice está compuesto por bloques construidos de forma digital como Look Up Tables (LUT), flip-flops etc. Un LUT puede ser programado para realizar lógica combinacional y cuando son combinados con flip-flops y otros recursos de hardware, pueden sintetizar cualquier sistema digital pequeño, en el rango de un circuito simple a una Unidad Central de Procesamiento (CPU) o una computadora a escala completa corriendo en un sistema operativo. Los FPGA son diseñados para ser configurados usando lenguaje de descripción de hardware como VHDL o Verilog[13].

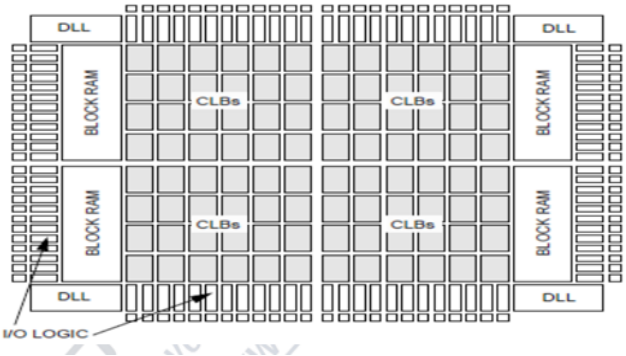

Figura 2. Estructura en bloques de la FPGA Spartan II[3]

4. Herramientas de software

MATLAB es un software desarrollado por la compañía MathWorks que contiene programación de alto nivel y se acompaña de un conjunto de librerías o herramientas *(toolboxes)*. Dentro de sus prestaciones se encuentran: la manipulación de matrices, la implementación de algoritmos, la creación de interfaces de usuario, la comunicación con programas en otros lenguajes y con dispositivos de hardware.

SIMULINK es un ambiente de software que corre sobre el MATLAB y utiliza una interfaz de usuario gráfica (GUI) que es utilizada para construir modelos de sistemas, en el dominio del tiempo o la frecuencia, para alguna operación específica de procesamiento, simulación, así como análisis de los resultados. En SIMULINK los modelos son jerárquicos y pueden ser discretos, continuos o híbridos[12]. SIMULINK

tiene además una amplia variedad de bloques de procesamiento de señales y pueden ser generados varios tipos de códigos: C/C++ para MCU y DSP; VHDL/Verilog para FPGA/ASIC y SPICE para dispositivos analógicos.

El software System Generator es una herramienta de simulación para el procesamiento digital de señales, que pertenece al entorno de Xilinx ISE y que se integra con la herramienta de simulación MATLAB, específicamente con el ambiente de software SIMULINK y consiste en una biblioteca de SIMULINK llamada Xilinx Blocksets. Dicha herramienta permite simular y verificar los modelos a nivel de bit y ciclo con los resultados de los modelos de referencia.

System Generator traza elementos del bloque Xilinx definidos en SIMULINK, como las arquitecturas, entidades, señales, puertos y atributos. También produce los archivos de comando para la síntesis en la FPGA y la simulación HDL. La herramienta guarda la jerarquía de SIMULINK cuando convierte en VHDL. Cuando se diseña en el System Generator es posible acceder a los rasgos importantes en la FPGA, como son los multiplicadores de gran velocidad y también es posible incorporar los bloques de usuario-definido de VHDL al modelo. Para la comprobación y prueba, el System Generator puede generar automáticamente los bancos de prueba o testbench dónde el estímulo de entrada de SIMULINK a la entrada del bloque, puede ser grabado para la simulación de VHDL. Las salidas pueden ser entonces comparadas con los resultados grabados de la simulación de SIMULINK en la simulación VHDL. La naturaleza del System Generator posibilita el rápido desarrollo de algoritmos para el enrutamiento de datos. Las funciones de encriptación son comúnmente implementadas utilizando arquitecturas de enrutamiento de datos, pues existe una compatibilidad natural entre estas funciones y la herramienta System Generator[3].

5. Sistema de encriptación de datos

En el diseño de los sistemas de encriptación de datos, juega un papel preponderante los Registros de Desplazamiento con Realimentación Lineal o LFSR (*Lineal Feedback Shift Register*), por ser un mecanismo de generación de datos encriptados en forma aleatoria. No obstante, los LFSR pueden ser modificados, en función de optimizar el aprovechamiento de las capacidades actuales de los procesadores, mediante estructuras de diseño algebraicas, llamadas cuerpos compuestos de Galois. Este nuevo diseño se conoce como Registro de Desplazamiento con Realimentación Lineal extendido o LFSRe

(*Lineal Feedback Shift Register extended*).

El hecho de realizar registros de desplazamiento extendidos consiste en definir los LFSR sobre cuerpos compuestos de Galois, $GF(2^n)^m$, en lugar de los tradicionales $GF(2^n)$.

Un ejemplo del diseño de un LFSRe se puede observar en la figura 4.

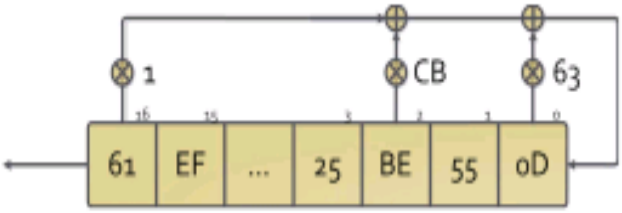

Figura 4. LFSRe de 17 etapas definido en GF (2ⁿ8)[16].

En esta nueva arquitectura los contenidos de las celdas St+i y los coeficientes de realimentación ya no son bits sino elementos de GF (2ⁿ8) expresados en notación hexadecimal. Este hecho afectará sobre todo las operaciones de suma y multiplicación que deben realizarse sobre ellos. En este ejemplo la función de realimentación es la siguiente:

$$S_{t+18} = 1 \otimes S_{t+17} \oplus CB \otimes S_{t+2} \oplus 63 \otimes S_t$$

6. Diseño de la Aplicación

A. Desarrollo en MATLAB/SIMULINK

En la figura 6 se muestra la implementación de un sistema de comunicaciones donde se vincula los LFSR como una aplicación de SDR, específicamente en la construcción de sistemas que buscan ofrecer seguridad. Se puede notar que el LFSR introducido se coloca para generar secuencias codificadas que se combinan con los datos que se producen de forma aleatoria y son recepcionados bajo una mayor seguridad en la información.

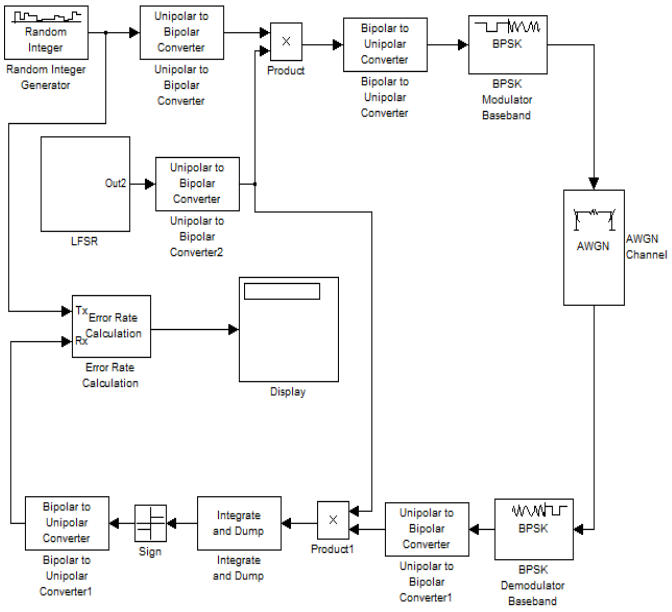

Figura 6. Modelo de transmisión de datos utilizando DSSS. Fuente: Elaboración propia.

El Registro de Desplazamiento implementado en el modelo es un LFSR extendido n=8. En el modelo, los datos son generados por un Random Integer Generator, que es un bloque de la biblioteca Communications System Toolbox de SIMULINK. Su función es generar de forma aleatoria distribuciones de números enteros y como la transmisión es binaria, las secuencias que se transmiten se ubican entre 0 y 1. Dentro de los parámetros de configuración del bloque se encuentran el Tiempo de Muestreo (Sample Time), que es el tiempo que se demora en generar un bit y el consecutivo y el número M-ario (*M-ary number*) que es el entero positivo que determina el rango de los valores de salida. Para el primero de los parámetros se adoptó el valor de 1/1000 y en el segundo M=2. En la figura 7 se muestra la salida de la secuencia cifrada para el LFSRe con n=8.

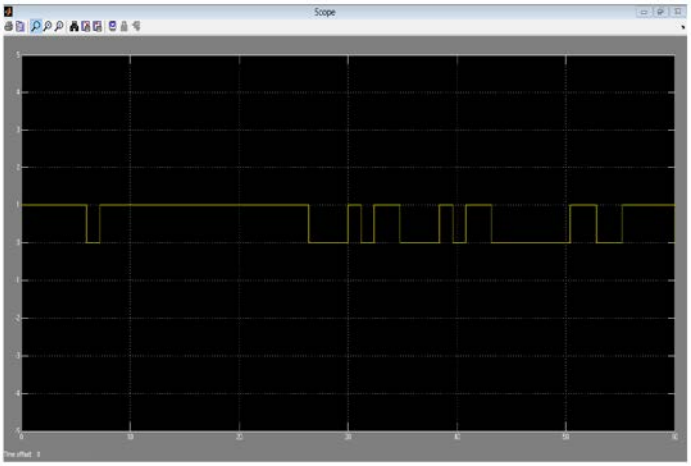

Figura 7. Secuencia de salida cifrada del LFSRe con n=8. Fuente: MATLAB.

Dentro de la etapa de transmisión, las secuencias comandadas por el *Random Integer Generator* son convertidas a señales bipolares, al igual que las generadas por el LFSR mediante el convertidor de señales unipolares a bipolares de la sub-biblioteca *Utility Blocks*, para luego ser multiplicadas ambas señales y convertidas a señales unipolares nuevamente. La secuencia a transmitir es modulada utilizando modulación digital de fase, en este caso BPSK (*Binary Phase Shift Key*), mediante un bloque específico que realiza esta modulación. Ahora en la etapa de recepción, se realiza la operación inversa, pues los datos son demodulados utilizando demodulación BPSK, convertidos a señales bipolares y multiplicadas por la secuencia codificada del LFSRe. El proceso de filtrado se realiza empleando el bloque *Integrate and Dump* de la sub-biblioteca *Comm Filters*. El bloque crea una suma cumulativa de la señal de entrada en tiempo discreto. Cuando comienza la simulación, el bloque desecha el número de muestras que se especifica en el parámetro Desplazamiento (Offset). En la figura 8 se muestra los resultados de la simulación.

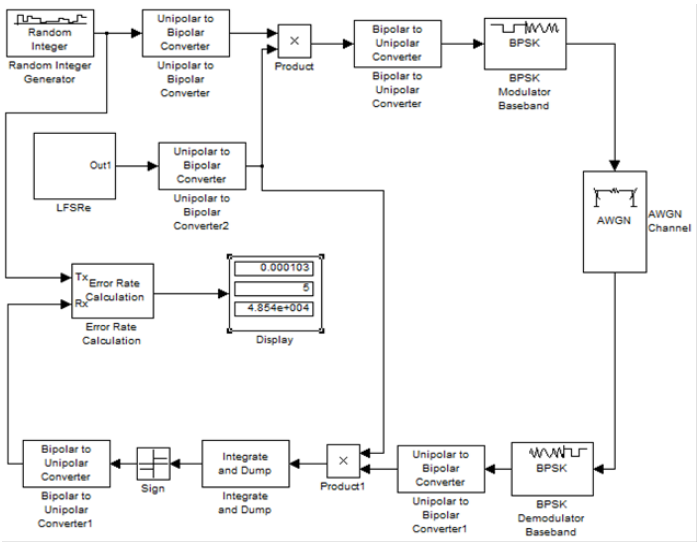

Figura 8. Resultados de la simulación. Fuente: MATLAB.

B. Desarrollo en System Generator.

En la figura 9 se muestra la fuente de datos utilizando el LFSRe.

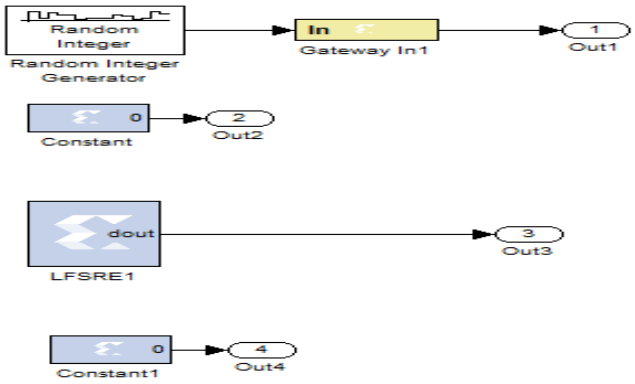

Figura 9. Fuente de Datos con LFSRe. Fuente: MATLAB.

En el subsistema de transmisión se realizan los procesos de conversión unipolar a bipolar y viceversa, multiplicación y modulación BPSK. Las conversiones son posibles gracias al bloque *MCod*e de la biblioteca Control Logic de Xilinx. Este es un contenedor que ejecuta una función de MATLAB proporcionada por el usuario dentro de SIMULINK. El bloque ejecuta el M-código para calcular los rendimientos del mismo durante la simulación y se traduce de una forma transparente equivalente al comportamiento VHDL cuando se genera en el hardware. La figura 10 muestra el transmisor.

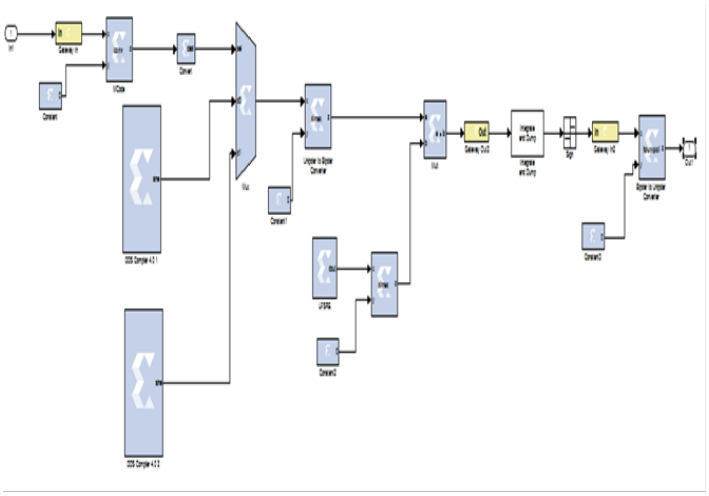

Figura 10. Subsistema de transmisión en SysGen. Fuente: MATLAB.

El canal de ruido es el mismo canal predeterminado en SIMULINK y contiene los mismos parámetros del bloque previamente configurado, explicado con. En la recepción, intervienen prácticamente los mismos bloques de la parte de transmisión. Se incluyen de igual forma como parte del filtrado de la señal de espectro extendido, los bloques de SIMULINK, *Integrate and Dump y Sign.* La figura 11 muestra el subsistema de recepción.

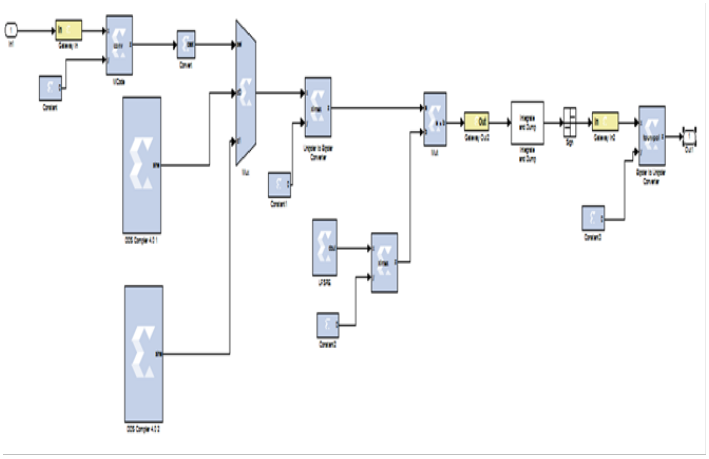

Figura 11. Subsistema de recepción. Fuente: MATLAB.

El SysGen posibilita utilizar una herramienta que realiza un análisis temporal de dicho diseño, en aras de conocer. La FPGA tiene por defecto un requerimiento de tiempo de 25ns. La figura 12 muestra el *Timing Analyzer* en *System Generator*.

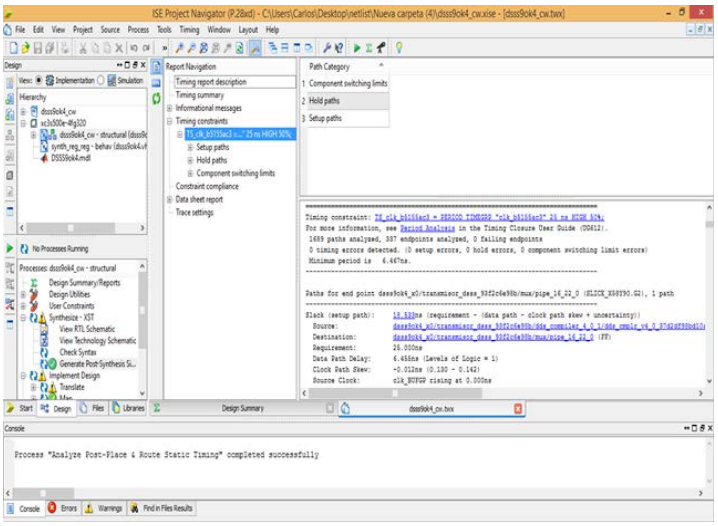

Figura 12. Comprobación del análisis temporal en SysGen del modelo DSSS con LFSRe. Fuente: MATLAB.

En este caso el tiempo máximo de configuración de camino de los datos es de 19.837ns, con retardo de 5.163ns. Si bien es cierto que la capacidad de los microprocesadores actuales soporta los elementos del LFSRe, este genera secuencias que se ven incrementadas a una tasa de 8 bits por cada pulso de reloj. Como puede compararse, es un resultado cuantitativamente mayor, debido a que el costo computacional y en cuanto a recursos a emplear, exigen de este modelo mayor requerimiento de tiempo, aunque a los efectos de la práctica, la diferencia de tiempo no es tan significativa, sólo de 0.63ns,

por lo que este modelo es viable en este aspecto.

La figura 13 muestra la utilización de recursos de la FPGA que requirió el diseño del sistema.

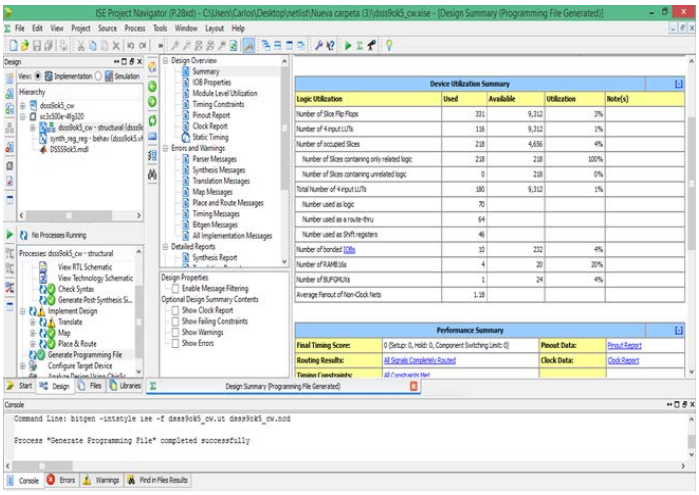

Figura 13. Utilización de recursos de la FPGA. Fuente: MATLAB.

La siguiente tabla muestra un resumen de los recursos utilizados en la FPGA.

Tabla I. Recursos utilizados de la FPGA. Fuente: Elaboración propia.

Recursos utilizados de la FPGA	Modelo con LFSRe	Total de recursos de la FPGA
Slice Flip Flops	*331 (3%)*	*9312*
4 entradas LUTs	*116 (1%)*	*9312*
Slices ocupadas	*218 (4%)*	*4656*
Total de 4 entradas LUTs	*180 (1%)*	*9312*
Uniones IOB	*10 (4%)*	*232*
RAMB 16s	*4 (20%)*	*20*
BUFGMUXs	*1 (4%)*	*24*

7. Resultados de la implementación

Para implementar el diseño realizado, se utilizó el Kit de desarrollo Nexys2 de Digilent, perteneciente a la familia Spartan 3E-500 y con la ayuda del osciloscopio digital se pudo obtener las señales de salida que resultan de la simulación y generación del código de

lenguaje de programación de alto nivel de hardware en el software MATLAB/SIMULINK y System Generator. Es importante destacar que mediante los bloques utilizados en el diseño previamente, pertenecientes a la biblioteca Xilinx Blockset de SIMULINK, se le asignaron los puertos de salida de la FPGA. En este caso se utilizaron los puertos K12 y M15 de la FPGA. Luego de ser previamente generado el modelo en el System Generator, donde internamente se le asignan los atributos y características de la señal, se creó el archivo con extensión .bit y mediante el emulador ADEPT, se transfiere toda la información del sistema a la FPGA y de esta forma se obtiene el resultado esperado. En la figura 14 se muestra la señal en el osciloscopio.

Figura 14. Implementación del modelo en la FPGA. Fuente: Elaboración propia.

8. Conclusiones

Con el desarrollo de este trabajo se abordó la tecnología de Radio Definida por Software y se implementaron aplicaciones de la misma, mediante la utilización de las herramientas de simulación MATLAB 2013 y System Generator de Xilinx. La Radio Definida por Software es una tecnología flexible, que se puede adaptar fácilmente a cualquier estándar de comunicaciones. Los dispositivos lógicos programables constituyen plataformas de hardware que poseen varios bloques lógicos, cuya evolución ha posibilitado desarrollar de forma satisfactoria cualquier aplicación de la SDR. Con el desarrollo de la simulación, se pudo comprobar que, con el modelo de transmisión de datos que emplea un Registro de Desplazamiento con Realimentación Lineal extendido, a pesar de su complejidad, se obtienen buenos resultados con requerimientos relativamente bajos en cuanto a parámetros de potencia, demora y probabilidad de error de bits. La implementación en el

Kit de desarrollo *Nexys2* corroboró la efectividad del modelo diseñado en las herramientas de simulación, para el desarrollo de aplicaciones de la Radio Definida por Software.

9. Referencias Bibliográficas

[1] HSIN-HUNG CHO, CHIN-FENG LAI, TIMOTHY K. SHIH, y HAN-CHIEH CHAO, «Integration of SDR and SDN for 5G», *IEEE Access*, vol. 2, oct. 2014.
[2] G. Baldini, T. Sturman, A. Rahim Biswas, R. Leschhorn, y G. Gódor, «Security Aspects in Software Defined Radio and Cognitive Radio Networks: A Survey and A Way Ahead», *IEEE Commun. Surv. Tutor.*, vol. 14, 2012.
[3] P. Borensztejn, «Diseño de Sistemas con FPGA». 2012.
[4] N. Katole, «FPGA Based SDR for DPLL Application», *Int. J. Innov. Eng. Sci.*, vol. 2, 2017.
[5] D. Rother, F. Jackisch, y J. Zöllner, «A Generic Framework for Evaluating the Performance of Software Defined Radio Algorithms», *IEEE Int. Conf. Consum. Electron.*, 2016.
[6] J. Varghese y L. Mathews, «Low Power Area Optimized Novel Architecture for Software Defined Radio in FPGA», *IEEE Int. Conf. Adv. Commun. Control Comput. Technol.*, 2014.
[7] Z. Zhao, Y. Shen, y Y. Bai, «Design and Implementation of the BPSK Modem Based on Software Defined Radio», *Int. Conf. Instrum. Meas. Comput. Commun. Control*, 2011.
[8] D. Reddy Nallapa Yoge y N. Chandrachoodan, «GPU Implementation of a Programmable Turbo Decoder for Software Defined Radio Applications», *25th Int. Conf. VLSI Des.*, 2012.
[9] A. O. Olopade, A. Hasan, y M. Helaoui, «Concurrent Dual-Band Six-Port Receiver for Multi-Standard and Software Defined Radio Applications», *IEEE Trans. Microw. THEORY Tech.*, vol. 61, dic. 2013.
[10] N. García Rodríguez, A. Rey Domínguez, y L. G. Raymond Rodríguez, «Diseño FPGA de un modulador DTMB para canalización de 6MHz», *RIELAC*, vol. 33, 2012.
[11] N. Alonso Torres, «RDS (Radio Definido por Software). Desafíos y consideraciones para su implementación», FACULTAD DE INGENIERÍA ELÉCTRICA. INSTITUTO SUPERIOR POLITÉCNICO JOSÉ ANTONIO ECHEVERRÍA, La Habana. Cuba, 2012.
[12] R. Supriya, S. Murugan, y R. Biradar, «Design and Implementation of Software Defined Radio Using Xilinx System Generator», *Int. J. Sci. Res. Publ.*, vol. 2, n.º 12, dic. 2012.
[13] P. Jovanović, D. Rakić, y B. Marković, «FPGA Implementation of Demodulators in Software Defined Radio Systems», *INFOTEH-JAHORINA*, vol. 12, mar. 2013.
[14] B. Naeem y A. Nyamapfene, «A Software Radio Design Approach for Heterogeneous Wireless Access Protocol Transceivers», *Pac. J. Sci. Technol.*, vol. 14, n.º 1, may 2013.
[15] B. Sklar, *Digital Communications. Fundamentals and Applications*, 2da ed. USA: Prentie Hall, 2006.

[16] O. . Mohatar y A. . Sabater, «Implementación Software de Registros de Desplazamiento sobre Cuerpos Extendidos», *Univ. Int. Castilla Ón*, p. 2012.

[17] A. Griffiths, «Building a Dircet sequence spread spectrum Model», *Univ. Staffs. GBK*, 2012.