

MODEM CORDIC 16QAM en FPGA para Satélites Cognitivos

Evelio Astaiza Hoyos

PhD(c) Procesado de señales y las comunicaciones
Universidad de Vigo-España
M.Sc. Ingeniería, área de telecomunicaciones,
Universidad del Cauca
Profesor Asistente- Universidad del Quindío
Grupo GITUQ
eastaiza@uniquindio.edu.co
eastaizah@unicauca.edu.co

Héctor Fabio Bermúdez Orozco

PhD(c) Procesado de señales y las comunicaciones,
Universidad de Vigo-España
M.Sc. Electrónica y Telecomunicaciones,
Universidad del Cauca
Profesor Asistente - Universidad del Quindío
Grupo GITUQ
hfbermudez@uniquindio.edu.co
hebermudez@unicauca.edu.co

Resumen— En este artículo se presentan los algoritmos de implementación para el modulador y demodulador en cuadratura 16 QAM como funciones SDR sobre Dispositivos Lógicos Programables, particularmente para plataformas FPGA. Se presenta el análisis del modelo, diseño e implementación del mismo, así como los resultados obtenidos pensando en la implantación del dispositivo como un componente de un transpondedor satelital cognitivo. El proceso de validación de la implementación se efectuó a través de las herramientas de diseño, simulación, síntesis y descarga de Xilinx sobre la FPGA Spartan 3E.

Palabras clave— Radio Definido por Software, QAM, Transpondedores Reconfigurables, CORDIC, FPGA, DSP, satélite.

Abstract— This paper presents the implementation algorithms for quadrature modulator and demodulator 16 QAM as SDR functions on programmable logic devices, particularly for FPGA platforms. It presents all the model analysis, design and implementation of it, and considering the results of implantation of the device as a component of a cognitive satellite transponder. The validation process of the implementation was done through the design tools, simulation, synthesis, and downloads on Xilinx Spartan 3E FPGA.

Keywords— Software Defined Radio, QAM, Transpondedores Reconfigurables, CORDIC, FPGAs, DSPs, satellites.

I. INTRODUCCIÓN

Las tecnologías emergentes tales como Software Defined Radio (SDR) han revolucionado las perspectivas de las telecomunicaciones; esto principalmente debido al hecho que los dispositivos hardware pueden ser reemplazados por algoritmos que se ejecutan sobre Dispositivos Lógicos Programables (PLD - Programmable Logic Devices) tales como DSP o FPGA. Considerando este nuevo paradigma, altos niveles de flexibilidad

pueden alcanzarse siempre que los algoritmos alojados en los PLD puedan ser actualizados o reemplazados completamente, lo cual equivale a cambiar un dispositivo hardware por uno nuevo, que presenta ventajas significativas en comparación con los sistemas actuales, dado que la reconfiguración, es un procedimiento sencillo, rápido y económico, que permite ser realizada en tiempo de ejecución, sin necesidad de suspender los servicios de comunicaciones prestados por la estructura del sistema.

Como se propone en [1], [2] y [3] los sistemas de comunicaciones satelitales pueden tomar ventajas de estas tecnologías emergentes y los transpondedores cognitivos podrían ser implementados basándose en tecnologías SDR. En este contexto, transpondedores satelitales cognitivos podrían ser construidos basados en arreglos de FPGA permitiendo realizar por completo las funciones de procesamiento en banda base y frecuencia intermedia a través de algoritmos de procesamiento digital de señales.

En este artículo se realiza una presentación de los resultados de la implementación del componente de modulación - demodulación para un transpondedor satelital cognitivo reconfigurable por medio de SDR utilizando plataformas FPGA, para este propósito se seleccionó el modulador - demodulador 16 QAM por su amplia utilización en el escenario de las comunicaciones satelitales. Los resultados e implementación se realizaron sobre la FPGA Spartan 3E de Xilinx por ser una de las plataformas más populares en el entorno académico. La herramienta de desarrollo utilizada se denomina System Generator, la cual es proporcionada por Xilinx.

II. DISEÑO DEL MODULADOR

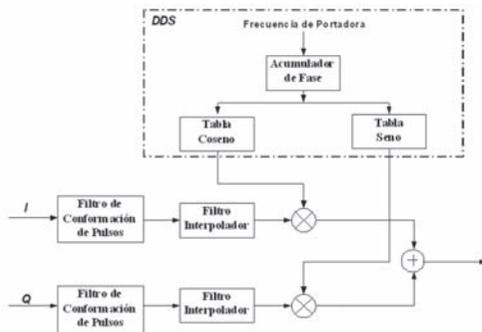
El diagrama en bloques del modulador QAM con salidas en fase y cuadratura se muestra en la Fig 1 con salidas de la forma (1) y (2)

$$I_{out}(n) = I(n)\cos(w_c n) + Q(n)\sin(w_c n) \quad (1)$$

$$Q_{out}(n) = Q(n)\cos(w_c n) - I(n)\sin(w_c n) \quad (2)$$

Donde $I(n)$ y $Q(n)$ son los símbolos de datos conformados e interpolados en fase y cuadratura [4], y al realizar una implementación directa de estas salidas, se requiere un total de cuatro multiplicaciones reales y dos sumas reales como lo muestra la Fig 1.

Fig. 1. ESQUEMA DEL MODULADOR QAM CON SALIDAS EN FASE Y CUADRATURA.

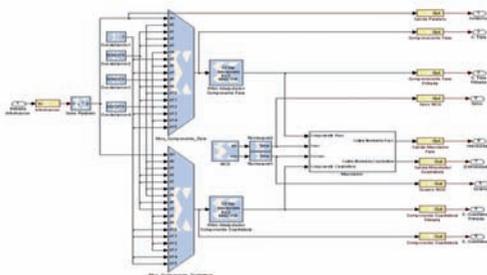


Fuente: Autor del proyecto

III. IMPLEMENTACIÓN DEL MODULADOR

El diagrama de la implementación del modulador 16QAM en la herramienta de desarrollo de Xilinx System Generator se muestra en la Fig. 2.

Fig. 2. MODULADOR 16QAM IMPLEMENTADO EN XILINX SYSTEM GENERATOR



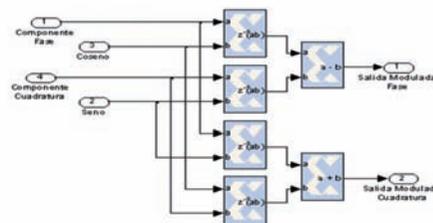
Fuente: Autor del proyecto

El modulador se encuentra constituido por un bloque de conversión serie a paralelo, encargado de realizar la conversión de los datos seriales de entrada a símbolos 16QAM, estos símbolos a su vez son empleados para seleccionar la salida de

los multiplexores, los cuales se encargan de la conformación de las componentes en fase y cuadratura de la señal QAM. Posteriormente las componentes en fase y cuadratura de la señal pasan a los filtros de transmisión, los cuales se diseñan como interpoladores de paso bajo que permite obtener a la salida de los mismos las componentes en fase y cuadratura limitadas en banda. La salida de los filtros posteriormente es convertidahacia arriba en el bloque mezclador cuya implementación se muestra en la Fig. 3. El bloque mezclador, hace uso de un NCO para la generación de las señales Seno y Coseno a la frecuencia deseada de portadora. Este NCO utiliza una ROM para generar sinusoides, y un integrador el cual opera como acumulador que genera una fase, la que se mapea en la ROM como direcciones que permite obtener el valor del senoide de salida de acuerdo a la fase de entrada; el incremento y desplazamiento de fase pueden ser definidos como constantes o pueden establecerse de forma dinámica a través de los puertos del oscilador, estos valores se definen en términos de ciclos por muestra, lo cual implica que por ejemplo para un incremento de fase de un décimo (1/10), un senoide se completa después de diez periodos de muestreo. Una vez el incremento de fase se acumula, el desplazamiento de fase se adiciona al resultado.

La implementación del bloque mezclador se muestra en la Fig. 3, el cual realiza el producto de las componentes en fase y cuadratura de la señal que permite el proceso de conversión hacia arriba; este producto se efectúa con un multiplicador complejo, el cual consta de cuatro bloques multiplicadores reales y dos sumadores restadores reales que se encargan de entregar la parte real y la parte imaginaria del producto, las cuales corresponden a las componentes en fase y cuadratura moduladas.

Fig. 3. MEZCLADOR MODULADOR 16QAM XILINX SYSTEM GENERATOR



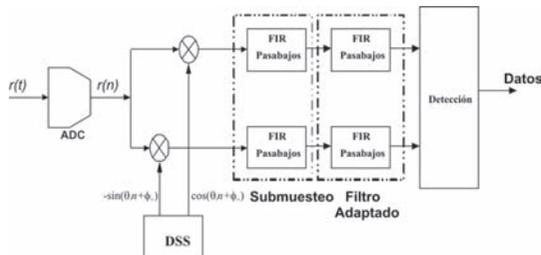
Fuente: Autor del proyecto

IV. DISEÑO DEL DEMODULADOR

El diagrama general del demodulador QAM se muestra en la Fig. 4.

Los multiplicadores de cuadratura y los filtros producen una estimación de las amplitudes en cuadratura, las cuales son las bases de la identificación de símbolos. La sincronización de portadora realiza la conversión a banda base de las componentes en cuadratura de la señal mediante réplicas de fase coherentes de las portadoras en cuadratura.

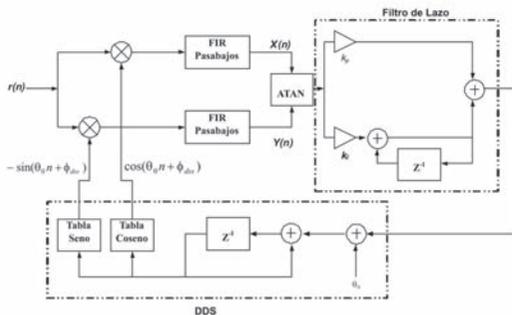
Fig. 4. ESQUEMA BÁSICO DEL DEMODULADOR QAM



Fuente: Autor del proyecto

Para el caso de la implementación del demodulador, uno de los componentes más importantes es el DPLL (Digital Phase Locked Loop), el cual emplea un filtro sumador proporcional integrativo y un camino directo escalado como se muestra en la Fig. 5.

Fig. 5. DPLL PARA SINCRONIZACIÓN DE FASE



Fuente: Autor del proyecto

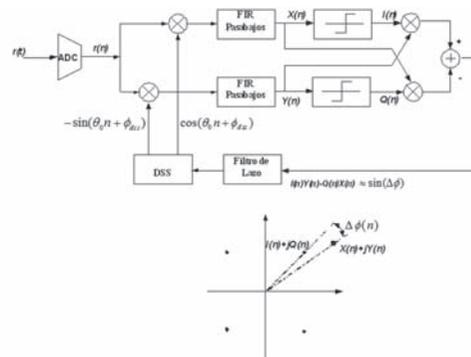
Los coeficientes del filtro K_p y K_i controlan el ancho de banda del PLL y el factor de derrape; en esta implementación el NCO (Oscilador Controlado Numéricamente) se conforma por medio de un sintetizador digital directo (DDS) y el detector de fase se implementa a través de la tangente inversa por medio del bloque que lleva su nombre.

Puesto que la fase de la señal modulada en QAM es dependiente de los datos, el detector de fase debe demodular la señal recibida y produ-

cir una señal proporcional a la diferencia de fase entre las portadoras locales y las recibidas. La complejidad del detector de fase se puede reducir por medio del cálculo de una señal proporcional al seno de la diferencia de fase $\Delta\varphi = \varphi - \varphi_{dbs}$ y como la función $\sin(\Delta\varphi)$ es monótonica en $-\pi/2 \leq \Delta\varphi \leq \pi/2$, se puede considerar como un buen estimador de fase en este intervalo, y dado que para pequeños valores de $\Delta\varphi, \sin(\Delta\varphi) \approx \Delta\varphi$, entonces la función seno se aproxima a un detector de fase ideal para valores pequeños de $\Delta\varphi$.

El error de fase se obtiene al comparar la diferencia de fase entre la señal recibida $x(n)+jy(n)$ y el punto más cercano de la constelación $I(n)+jQ(n)$ tal como se muestra en la Fig. 6.

Fig. 6. DPLL IMPLEMENTADO CON PRODUCTOS CRUZADOS Y LA CONSTELACIÓN



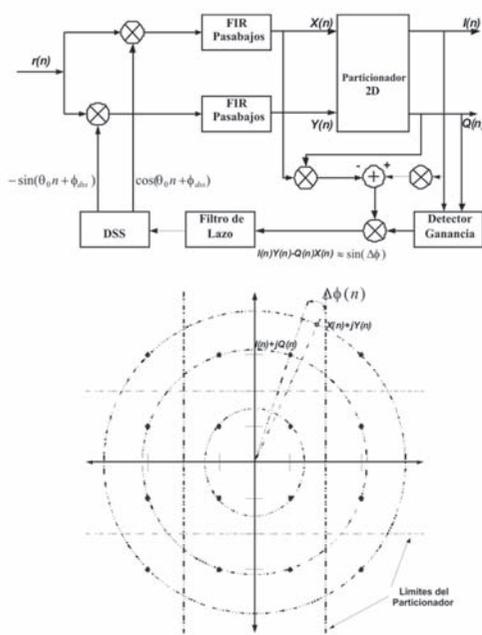
Fuente: Autor del proyecto

Para la constelación de ejemplo, los cuatro puntos pertenecientes a la constelación se ubican en el centro de los cuatro cuadrantes, luego el punto más cercano de la constelación se calcula por medio del signo de las magnitudes de las componentes en fase y cuadratura y el seno de la diferencia de fase puede expresarse como:

$$\sin(\Delta\varphi) = \frac{\{I(n)Y(n) - [Q(n)X(n)]\}}{\sqrt{I^2(n) + Q^2(n)}\sqrt{X^2(n) + Y^2(n)}} \quad (3)$$

La ecuación (3) muestra que el error de fase es proporcional al seno de la diferencia de fase, que puede ser generado a partir de la diferencia de los productos cruzados $I(n)Y(n) - Q(n)X(n)$, por lo tanto, el detector de fase para una constelación densa como 16QAM debe contener más comparaciones para determinar el punto más cercano de la constelación, por esta razón esta función se implementa mediante un particionador como se muestra en la Fig. 7.

Fig. 7. DPLL PARA 16QAM



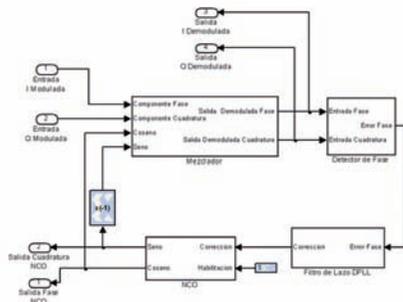
Fuente: Autor del proyecto

Los puntos de la constelación de 16QAM residen en uno de tres contornos e indican su distancia desde el origen, el radio del contorno mayor, representa señales con una alta relación señal a ruido y la medición de fase para puntos ubicados en este contorno presentan menor ruido por lo que el PLL debe controlar la ganancia de acuerdo al radio del símbolo detectado, función desempeñada por el bloque de detección de ganancia que se muestra en la Fig. 7.

V. IMPLEMENTACIÓN DEL DEMODULADOR

El diagrama de la implementación del demodulador 16QAM en la herramienta de desarrollo de Xilinx System Generator se muestra en la Fig. 8.

Fig. 8. DEMODULADOR 16QAM EN XILINX SYSTEM GENERATOR

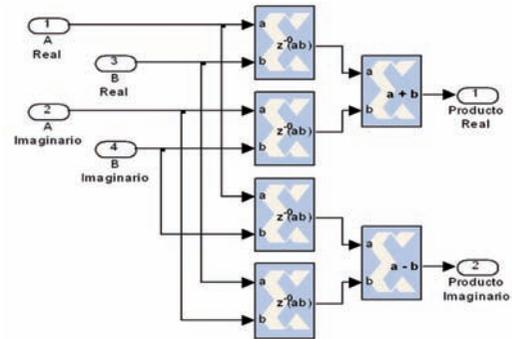


Fuente: Autor del proyecto

Está conformado por un bloque mezclador el cual se muestra en la Fig. 3, como el utilizado por el modulador; un bloque detector de fase que cumple la función de la recuperación de la fase de la portadora de la señal modulada puesto que la fase de la señal modulada en QAM es dependiente de los datos, el detector de fase debe producir una señal proporcional a la diferencia de fase entre las portadoras locales y las recibidas, que permite calcular una señal proporcional al seno de la diferencia de fase $\Delta\phi = \phi - \phi_{dss}$ como se describe en el numeral anterior de diseño. Un bloque DPLL para el enganche de portadora, que permite generar el error de fase alimentado al NCO, el cual se implementa de igual manera al utilizado por el modulador.

La implementación de los bloques que componen el demodulador se muestra en las Fig. 9 a 17.

Fig. 9. MULTIPLICADOR COMPLEJO ENTRADA B COMPLEMENTADA EN XILINX SYSTEM GENERATOR



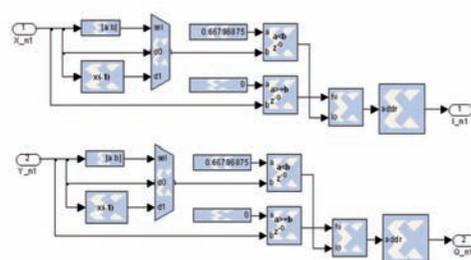
Fuente: Autor del proyecto

Fig. 10. DETECTOR DE FASE EN XILINX SYSTEM GENERATOR



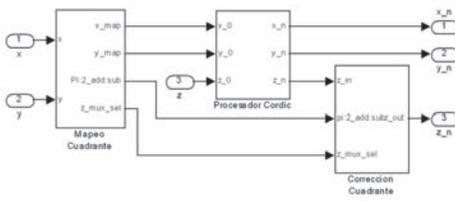
Fuente: Autor del proyecto

Fig. 11. SLICER EN XILINX SYSTEM GENERATOR



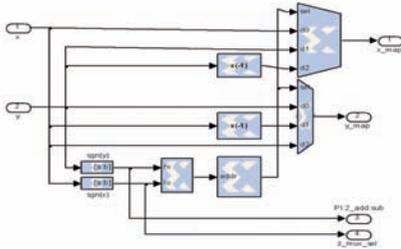
Fuente: Autor del proyecto

Fig. 12. BLOQUE CORDIC EN XILINX SYSTEM GENERATOR



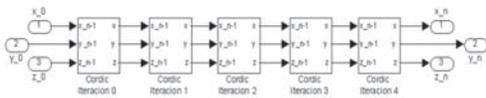
Fuente: Autor del proyecto

Fig. 13. PROCESADOR CORDIC EN XILINX SYSTEM GENERATOR



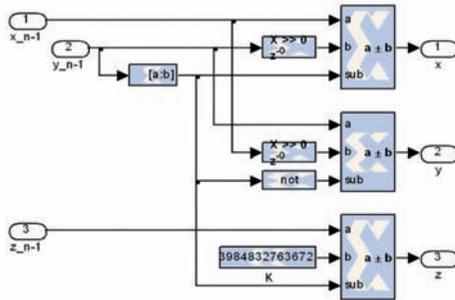
Fuente: Autor del proyecto

Fig. 14. PROCESADOR CORDIC EN XILINX SYSTEM GENERATOR



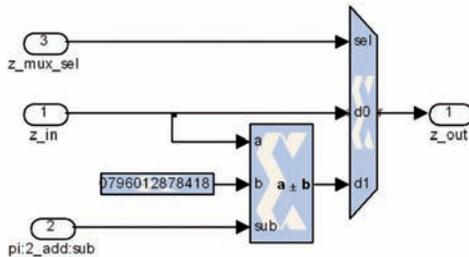
Fuente: Autor del proyecto

Fig. 15. CORDIC ITERACIÓN N EN XILINX SYSTEM GENERATOR



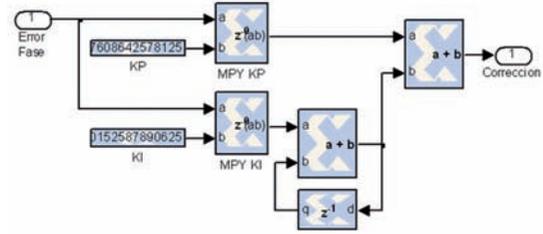
Fuente: Autor del proyecto

Fig. 16. CORRECCIÓN CUADRANTE EN XILINX SYSTEM GENERATOR



Fuente: Autor del proyecto

Fig. 17. FILTRO DE LAZO DPLL EN XILINX SYSTEM GENERATOR



Fuente: Autor del proyecto

VI. RESULTADOS DE IMPLEMENTACIÓN

Con la herramienta de estimación de recursos se calcularon los recursos requeridos en la FPGA para realizar la implementación, estos resultados se presentan en las Tablas I y II.

TABLA I

ESTIMACIÓN DE RECURSOS TOTALES DE IMPLEMENTACIÓN

Recurso	Modulador	Demodulador	Total
Slices	786	1191	1977
FFs	488	52	540
BRAMs	0	1	1
LUTs	1277	22280	3507
IOBs	193	0	193
Mult. Emb.	20	28	48
TBUFs	0	0	0

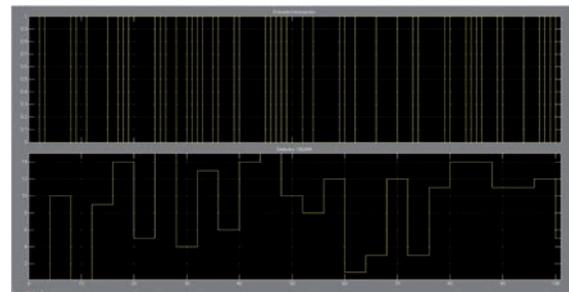
TABLA II

ESTIMACIÓN DE RECURSOS DE IMPLEMENTACIÓN POR BLOQUES FUNCIONALES

Recurso	Mezclad.	Det. Fase	DPLL	NCO
Slices	314	782	68	18
FFs	0	0	32	20
BRAMs	0	0	0	1
LUTs	618	1521	104	20
IOBs	0	0	0	0
Mult. Emb.	12	12	4	0
TBUFs	0	0	0	0

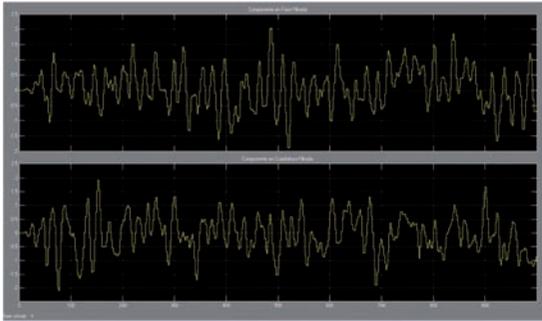
En las Fig. 18 a 24, se muestran los resultados de implementación.

Fig. 18. INFORMACIÓN DE FUENTE Y SÍMBOLOS 16 QAM



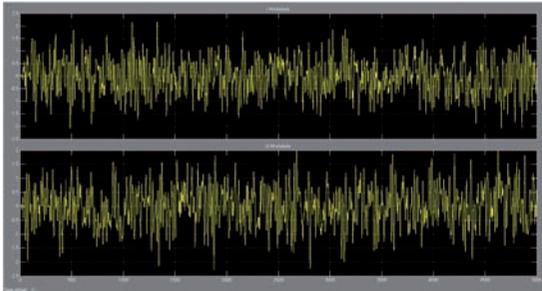
Fuente: Autor del proyecto

Fig. 19. COMPONENTES EN FASE Y CUADRATURA POST FILTRO DE TRANSMISIÓN.



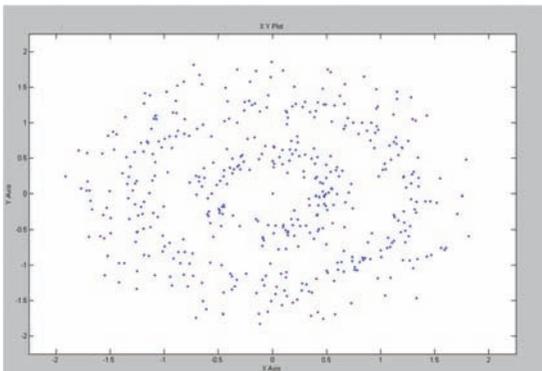
Fuente: Autor del proyecto

Fig. 20. COMPONENTES EN FASE Y CUADRATURA MODULADAS.



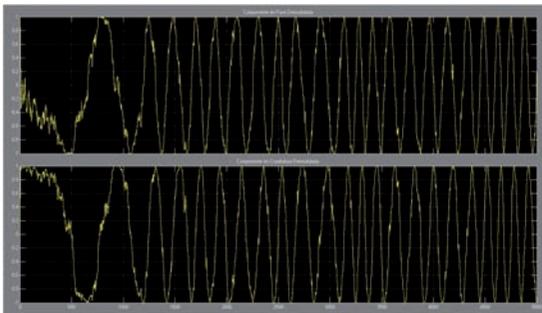
Fuente: Autor del proyecto

Fig. 21. DIAGRAMA DE CONSTELACIÓN DE LA SEÑAL MODULADA DESPUÉS DEL CANAL.



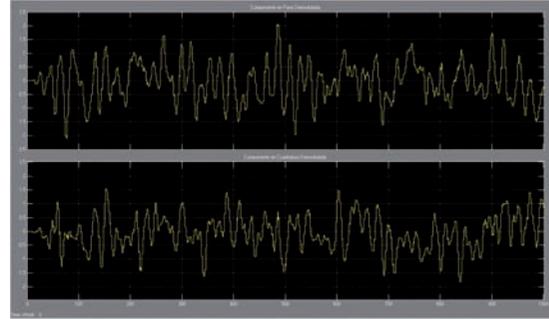
Fuente: Autor del proyecto

Fig. 22. RESPUESTA DEL NCO DE RECEPCIÓN Y RECUPERACIÓN DE FASE.



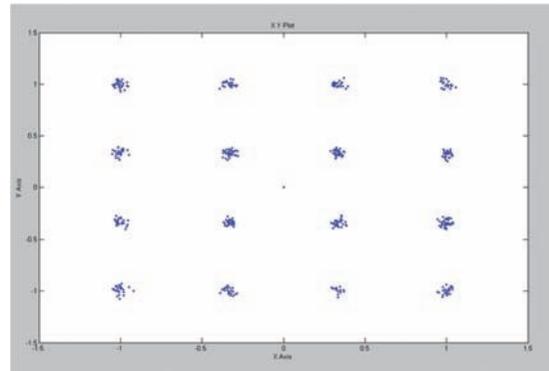
Fuente: Autor del proyecto

Fig. 23. COMPONENTES EN FASE Y CUADRATURA DEMODULADAS.



Fuente: Autor del proyecto

Fig. 24. DIAGRAMA DE CONSTELACIÓN DE LA SEÑAL DEMODULADA.



Fuente: Autor del proyecto

VII. CONCLUSIONES Y RECOMENDACIONES

En la medida en que el procesamiento digital de señales incrementa su complejidad, para permitir la implementación de funciones más sofisticadas y óptimas sobre plataformas hardware cada vez más robustas, el desarrollo de software y propiedad intelectual llegarán a ser más importantes que la implementación de la plataforma hardware que soporta la funcionalidad del sistema, debido a que la funcionalidad de la mayoría de los sistemas serán desarrolladas en núcleos de propiedad intelectual (*IP Cores*) que serán los encargados de proporcionar la funcionalidad requerida en sistemas cognitivos.

Se considera que SDR es una tecnología promisoría la cual contribuirá al mejoramiento de los sistemas satelitales, sin embargo, su implementación en transpondedores satelitales cognitivos reales puede llevar algunos años, lo anterior, de acuerdo a los avances tecnológicos que puedan alcanzarse en el desarrollo de ADC que cumplan con las especificaciones requeridas para la implementación de SDR, y del desarrollo de dispositivos

lógicos programables como DSP y FPGA con las suficientes capacidades de procesamiento requeridas para la implementación de los algoritmos.

Las implementaciones mediante SDR de las funciones de radio tradicionalmente analógicas, abren las puertas a nuevos niveles optimizados de funcionalidad de los dispositivos en órbita y de la implementación de nuevas tecnologías de radio inteligente como lo es radio cognitivo[5 -20].

REFERENCIA

- [1] Paillassa, B.; Morlet, C. "Flexible satellites: software radio in the sky". 10th International Conference on Telecommunications. ICT 2003. Volume 2, 23 Feb - 1 March 2003 Pp:1596 – 1600
- [2] Morlet, C.; Boucheret, M.-L.; Calmettes, V.; Paillassa, B.; Perennou, T. "Towards Generic Satellite Payloads: Software Radio". .Parallel and Distributed Processing International Symposium, 2003. Proceedings. 22-26 April 2003 P 7
- [3] Medina P, Samir; Astaiza H, Evelio and Vera V, Pedro. "Reconfigurable Satellite Payload Model based on Software Radio Technologies". Presented to the Third IEEE International Congress of the Andean Region – Andescon2006.
- [4] Tan L. K., and H. Samueli, "A 200 MHz Quadrature Digital Synthesizer/Mixer in 0.8 μm CMOS," IEEE Solid State Circuits, Vol. 30, No. 3, pp. 193-200, Mar. 1995.
- [5] Haykin, Simon. "Cognitive Radio: Brain-Empowered Wireless Communications". IEEE Journal on Selected Areas in Communications, Vol. 23, No 2. February 2005.
- [6] Harada, H. y Prasad, R. (2002) Simulation and Software Radio for Mobile Communications. Artech House Publishers. Nueva York. 467p.
- [7] J. Mitola, "Cognitive radio for flexible mobile multimedia communications," in Proc. IEEE Int. Workshop Mobile Multimedia Communications, Pp. 3–10, 1999.
- [8] Haiyun Tang, "Some Physical Layer Issues of Wide-Band Cognitive Radio Systems", First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. 8-11 Nov. Pp. 151 – 159.
- [9] L. Le and E. Hossain, "Resource allocation for spectrum underlay in cognitive radio networks," IEEE Transactions on Wireless Communications, vol. 7, Issue 12, Part 2, December 2008.
- [10] J. George, A. Sultan, and M. Nafie, " Distributed admission and power control for cognitive radios in spectrum underlay networks," to appear, GLOBECOM proceedings.
- [11] D. I. Kim, L. Le, and E. Hossain, "Joint rate and power allocation for cognitive radios in dynamic spectrum access environment," IEEE Transactions on Wireless Communications, vol. 7, Issue 12, Part 2, December 2008.
- [12] P. Maille and B. Tuffin. "Optimization of transmission power in competitive wireless networks". Lecture notes in computer science., May 2009.
- [13] W. Yu, W. Rhee, S. Boyd, and J. Cioffi. "Iterative water-filling for Gaussian vector multiple-access channels". IEEE Transactions on Information Theory, 50(1):145–152, Jan. 2004.
- [14] Rodríguez, Virgilio, "An Analytical Foundation for Resource Management in Wireless Communications". IEEE Global Telecommunications Conference, San Francisco, CA, 1-5 December 2003.
- [15] Heeyoung Lee, Seongkwan Kim, Okhwan Lee, Sunghyun Choi, Sung-Ju Lee, "Available Bandwidth-Based Association in IEEE 802.11 Wireless LANs". MSWiM '08 Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems, ACM New York, NY, USA ©2008: 132-139, 2008.
- [16] Li-Hsing Yen; Jia-Jun Li; Che-Ming Lin. "Stability and Fairness of Native AP Selection Games in IEEE 802.11 Access Networks". Seventh International Conference on Wireless and Optical Communications Networks (WOCN) 20 Pp. 1 – 5, 2010.
- [17] Bonald, T.; Ibrahim, A.; Roberts, J. "The Impact of Association on the Capacity of WLANs". 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. WiOPT 2009. Pp. 1 – 10.
- [18] Ghini, V.; Ferretti, S.; Panzieri, F. "A strategy for best access point selection". Wireless Days (WD), 2010 IFIP, Pp. 1 – 5.
- [19] I-Ping Hsieh, Fu-Min Chang, and Shang-Juh Kao. "Adaptive Access Points Selection for 802.11 Wireless Networks". Proceeding AsiaCSN '07 Proceedings of the Fourth IASTED Asian Conference on Communication Systems and Networks. Pp. 218-222. 2007.
- [20] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose and D. Towsley. "Facilitating Access Point Selection in IEEE 802.11 Wireless Networks". IMC '05 Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. Pp. 26-27.