



Diseño de un receptor DVB-S en VHDL utilizando las herramientas del entorno MATLAB/Simulink

Alejandro González García, Reinier Díaz Hernández

RESUMEN

Los sistemas de radiodifusión satelital son una alternativa económica para el despliegue de la televisión y otros servicios en áreas poco pobladas y de difícil acceso. En el presente trabajo se presenta la modelación e implementación hardware de un receptor DVB-S según las especificaciones contenidas en el estándar EN 300 421. El receptor es modelado y validado dentro del entorno Simulink. A partir del modelo validado se genera un código VHDL equivalente utilizando las herramientas HDL Coder y HDL Verifier. En su diseño se utilizan tanto módulos HDL sintetizables de Simulink, como módulos IP del Xilinx Core Generator para su implementación y validación sobre un entorno hardware, se utiliza la herramienta FPGA in the Loop. Se realiza una comparación entre la implementación con bloques HDL nativos del entorno Simulink y la utilización de módulos de la plataforma Xilinx.

Palabras claves: DVB-S, HDL Coder, FPGA-in-the-Loop, Simulink, Xilinx Core Generator.

ABSTRACT

Satellite broadcasting systems are an economical alternative for the deployment of television and other services in sparsely populated and difficult to access areas. This paper presents the modeling and hardware implementation of a DVB-S receiver according to the specifications contained in the EN 300 421 standard. The receiver is modeled and validated within the Simulink environment. From the validated model, an equivalent VHDL code is generated using the HDL Coder and HDL Verifier tools. Both Simulink synthesizable HDL modules and Xilinx Core Generator IP modules are used in the design for implementation and validation in a hardware environment, using the FPGA in the Loop tool. A comparison is made between implementation with native HDL blocks from the Simulink environment and the use of modules from the Xilinx platform.

Keywords: DVB-S, HDL Coder, FPGA-in-the-Loop, Simulink, Xilinx Core Generator

1.- INTRODUCCION

La televisión satelital, por sus características intrínsecas, es la plataforma de radiodifusión con mayor presencia a nivel internacional. Desde su lanzamiento en 1995, la versión para satélites de la plataforma DVB, DVB-S [1] ha ocupado un lugar preponderante entre las opciones de radiodifusión satelital. A pesar del lanzamiento de la segunda generación de este sistema con el nombre de DVB-S2 [2], la simplicidad de su diseño le ha permitido seguir contando con un importante segmento del mercado, incluyendo Free-to-Air (FTA) y ampliarse a otros segmentos como la transmisión de datos. Además, su prácticamente omnipresencia en todos los entornos geográficos ha generado propuestas de uso como iluminador en sistemas de radares [3].

En *LACETEL*, Instituto de Investigación y Desarrollo de Telecomunicaciones, se han realizado estudios mediante modelos en Simulink sobre el efecto de la interferencia en un receptor DVB-S [4]. Sin embargo, la no disponibilidad de un receptor

profesional con información sobre el estado de la señal en cada una de las etapas dentro del sistema, imposibilita realizar un estudio más detallado sobre el comportamiento de esta interferencia y como afecta a cada una de las partes. Por esta razón se hace necesario contar con un modelo implementado en hardware que nos permita acceder a esta información en condiciones reales y prácticas. Una alternativa para la implementación de este tipo de sistemas de pruebas es la utilización de un FPGA y código HDL por las opciones de reconfiguración que ofrecen.

El entorno Simulink cuenta con varias herramientas para la implementación de los algoritmos y el modelado de sistemas. Dentro de ellas destacan dos alternativas para el desarrollo en FPGAs de Xilinx. Una de ellas es Xilinx System Generator for DSP, que es una herramienta propia de Xilinx dentro del entorno Simulink para la implementación de algoritmos DSP en los productos de esta compañía. Su principal ventaja es que cuenta con más de 100 bloques IP optimizados para estos FPGAs y parametrizables por el usuario. Sin embargo, una de sus mayores desventajas es que no existe un mecanismo implícito de sincronismo por lo que es responsabilidad del diseñador garantizar esto de forma explícita [5].

Por otro lado, se encuentran una serie de herramientas brindadas dentro del entorno MATLAB/Simulink para la generación y validación de código HDL sintetizable [6]. Este conjunto de herramientas lo constituyen:

- Fixed Point Designer: para la conversión y optimización de algoritmos y modelos en punto flotante en punto fijo
- HDL Coder: para la generación de código HDL sintetizable desde funciones de MATLAB, modelos de Simulink y charts de Stateflow. Permite el control de la arquitectura e implementación, así como un esquema de trazabilidad entre el HDL generado y los componentes del sistema.
- HDL Verifier; para la generación de bancos de pruebas en la verificación de los diseños. Además, genera componentes para la validación con simuladores HDL externos y en FPGA.
- FPGA-in-the-Loop (FIL): para la validación de código HDL sobre un FPGA dentro del flujo de la simulación mediante un vínculo entre el simulador y la tarjeta.
- HDL Workflow Advisor: para la integración de las herramientas precedentes en una interfaz gráfica.

Sin embargo, el código HDL generado es genérico y no optimizado por lo que es de esperar que los diseños elaborados con la plataforma, tiendan a ser menos eficientes que su contraparte diseñada en Xilinx System Generator con módulos optimizados para los FPGAs de Xilinx.

Una variante intermedia entre ambas propuestas es explorada en la literatura [7] y plantea la inclusión de un sistema implementado en Xilinx System Generator dentro de un bloque en una capa inferior a la del bloque a traducir mediante HDL Coder. Esto permite al flujo de HDL Coder llamar al token de System Generator para generar los netlist correspondientes a este bloque. La implementación mediante el uso de esta plataforma, en cualquiera de sus dos variantes, es descartada al no contar con licencia para su uso.

En años previos se presentaron varios trabajos referentes a la utilización de las herramientas de Mathwork para el trabajo con HDL en múltiples campos de aplicación. Järviluoma [8] demuestra la viabilidad para el prototipado rápido de algoritmos sobre FPGA en comparación con la opción clásica de tipo manual y Gongpei [9] optimiza un sistema de asistencia al conductor mediante aceleración hardware. Hai [10] lo utiliza en una solución de procesamiento de videos para la extracción de características, en un FPGA de Altera, mientras Besbes [11] amplía su uso al campo de los motores de inducción para el modelado de un observador de alta ganancia. En 2017 Cai [12] presenta un sistema SDR utilizando una metodología basada en introducida en [6] y en [13] Pasha implementa un sistema DVB-T sobre FPGA para el estudio del BER. En 2018 Youngmi [14], valida este esquema mediante el diseño de un ecualizador polinomial no lineal para la implementación de la transformada rápida de Fourier(FFT).

En el presente trabajo, se presenta la modelación en punto fijo de un modelo de receptor DVB-S y su validación. Además, se realiza la conversión a HDL equivalente en VHDL mediante la utilización de HDL Coder y su validación mediante las herramientas descritas. Para el diseño se utilizan dos alternativas. En la primera, se utilizarán bloques HDL nativos sintetizables del entorno Simulink. En la segunda, se modificarán los principales bloques del sistema para utilizar bloques IP optimizados del Xilinx Core Generator para los FPGA de Xilinx.

Los principales aportes del trabajo son el diseño y validación de un receptor DVB-S utilizando las herramientas de generación y validación del entorno MATLAB/Simulink. Además, se presenta la utilización de módulos IP el Xilinx Core Generator dentro del flujo del HDL Workflow Advisor. Por último, se realiza una comparación de los resultados obtenidos utilizando ambas variantes en cuanto a utilización de recursos y frecuencia máxima de reloj permitida.

El presente trabajo está dividido en 6 secciones. En la sección 2 se presenta la metodología empleada y el diseño de las pruebas de validación. A continuación, se realiza un bosquejo del estándar DVB-S, mientras que en la sección 4 se presenta el diseño propuesto y las especificidades de los bloques diseñados. En la quinta se muestran los resultados de la modelación en cuanto a recursos empleados y los resultados de las pruebas de validación. Así mismo, se realiza una comparación entre los resultados obtenidos para cada alternativa.

2.- METODOLOGÍA

2.1.- METODOLOGÍA DE DISEÑO

Las particularidades de diseño e implementación de un sistema inalámbrico sometido a pruebas en tiempo real, requieren el uso de una metodología bien estructurada que permita un desarrollo robusto y modular del mismo. Una metodología efectiva en el desarrollo de estos sistemas es la estrategia de Diseño Basado en Modelos (MBD, por sus siglas en inglés) [6]. Partiendo de un análisis de una modelación ideal del sistema, el cual es evaluado primero mediante simulaciones iterativas, se realizan las modificaciones necesarias para su adaptación a condiciones reales. El sistema es modular, lo cual facilita la sustitución, modificación y reutilización de sus diferentes componentes.



Figura 1

Esquema de la metodología empleada.

En el presente trabajo se emplea una metodología similar a la planteada por Shah [15] (Figura 1), pues es utilizado ampliamente en la literatura descrita. A excepción de la primera etapa de la metodología, correspondiente al diseño en punto fijo del modelo, el resto de las etapas pueden ser acometidas dentro del flujo de trabajo del HDL Workflow Advisor. Para la implementación de la variante con bloques HDL nativos de Xilinx no son necesarias las modificaciones en el flujo de diseño pues el código HDL generado es sintetizable.

En el caso de los bloques IP de Xilinx Core Generator, se emplea la variante de utilizar bloques Blackbox. Esta interfaz permite la inclusión de módulos IP de Xilinx mediante la modelación de bloques con interfaces equivalentes y un comportamiento similar. Aunque se ha abordado en la literatura [16] la utilización de los Blackbox dentro del entorno HDL Coder para la inclusión de código HDL de terceros, no se ha encontrado evidencia de su utilización para la inclusión de los módulos IP de Xilinx, excluyendo lo presentado en [7] que aborda la cuestión de forma distinta.

Para el modelado con los módulos del Core, se utilizan dos variantes. Para la validación funcional con Cosimulación HDL, se utiliza el código HDL funcional, generado por el Core, insertado mediante la funcionalidad DocBlock de Simulink dentro de cada Blackbox. En el caso de la simulación hardware mediante FPGA-in-the-Loop, se inhabilitan los DocBlocks y se añaden los netlist (.ngc) generados por el Core dentro del ciclo del HDL Workflow Advisor previo a la creación de los archivos requeridos para el FIL.

La explicación para este proceder reside en que el código HDL proporcionado para efectos de simulación por el Core e incluido en los Blackbox, no es sintetizable por lo que solo puede ser utilizado para la validación funcional. Además, la opción DocBlock no soporta la inclusión de otro tipo de archivo de descripción. Una variante pudiera ser la de no utilizar la opción de DocBlock e insertar los netlist de cada IP en la carpeta de trabajo manualmente, pero la variante de DocBlock permite un proceso automático de inclusión y validación dentro del flujo del HDL Workflow Advisor.

Para el desarrollo del sistema se cuenta como referencia con un modelo en punto flotante de un sistema Transmisor DVB-S en banda base, validado en condiciones reales según la metodología propuesta por Díaz [17]. Para la validación del código HDL generado mediante Cosimulación se utiliza el simulador ModelSim mediante la interfaz generada automáticamente por la herramienta HDL Verifier.

Para la temporización del sistema modelado se utiliza un único reloj con habilitaciones para implementar un entono con múltiples requerimientos de temporización. Esta alternativa posee como ventaja, respecto al modelo que emplea varias señales

de reloj, que solamente es necesario generar un único reloj. Esto contribuye a reducir la complejidad del sistema y del esquema de sincronización entre bloques.

2.2.- ESQUEMA DE VALIDACIÓN DEL MODELO

Para la validación del modelo se utilizaron dos esquemas. El primero está basado en un modelo de referencia validado, mientras el segundo utiliza equipamiento especializado para la elaboración de una señal real basada en el estándar.

Por una parte, se utiliza un modelo de transmisor comentado en la metodología y basado en la descripción recomendada por el estándar DVB-S, para generar los datos de entrada al modelo del receptor. Este modelo fue elaborado sobre la plataforma Simulink y validado como modelo de referencia utilizando una metodología similar a la propuesta por Díaz [17]. Esta metodología ha sido reelaborada para adaptarla a las características del estándar de radiodifusión satelital. Este esquema de diseño mediante una referencia validada permite verificar cada una de las etapas del sistema antes de proceder a la siguiente, lo cual disminuye el grado de incertidumbre. Este es el esquema utilizado para la elaboración del modelo y para la elaboración de las gráficas de BER (Razón de bit erróneo) vs Eb/No (Energía de bit sobre el nivel de ruido) a la salida del decodificador Viterbi.

El otro esquema de validación utiliza equipamiento especializado para generar una señal de referencia real. El flujo de trabajo empleado se presenta en Figura 2. Este esquema utiliza la herramienta Digital Video del fabricante Keysight para generar un fichero con las salidas en fase (I) y cuadratura (Q) basada en el estándar DVB-S a partir de una trama de transporte compatible con ISO 13818-1 [18]. Esta información es grabada en la memoria interna del generador de patrones PSG E8267D. Esto permite generar una señal DVB-S real en una frecuencia predeterminada por el operador, así como obtener una señal en banda base de las componentes IQ. A continuación, mediante el uso de un osciloscopio, se almacena esta información en formato CVS (Valores separados por coma). Este fichero se modifica para obtener un arreglo compatible en un fichero '.mat', que es la extensión para archivos de datos de MATLAB. Esta información debe ser modificada para ser compatible con la entrada en punto fijo del modelo del receptor implementado. El modelo a validar se alimenta con esta señal y se compara la salida obtenida con la información de la trama de transporte utilizada para la validación. En caso de coincidir, se comprueba la validez del modelo para la recepción de este tipo de señales.

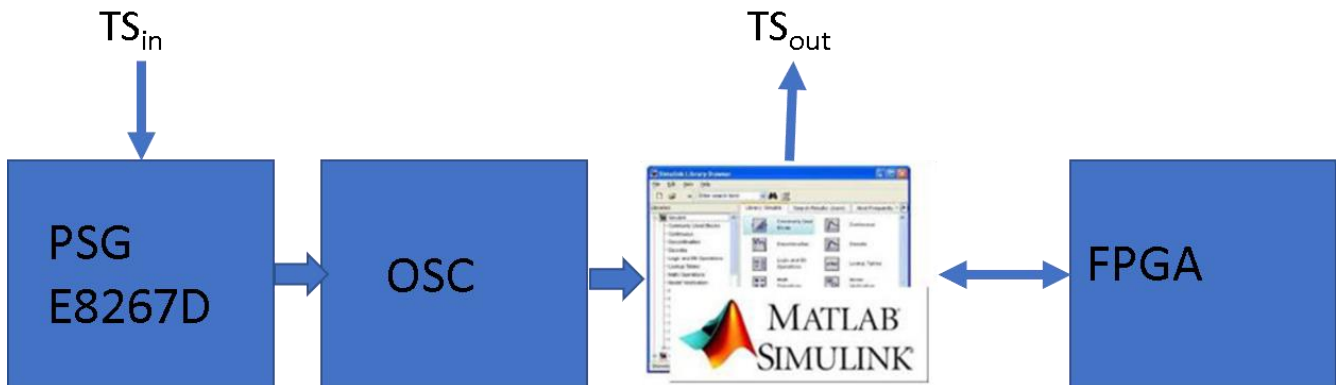


Figura 2

Esquema para el trabajo con hardware especializado.

Para realizar las comprobaciones se utilizan dos tramas de transporte (Figura 3). La primera, llamada 'nulo.ts' contiene solamente paquetes nulos, según se define en ISO 13818-1 [18]. La segunda, llamada 'football.ts' contiene tres programas multiplexados con toda la información de video, audio y datos (tomado de <http://www.dododge.net/roku/ts-samples.html>).

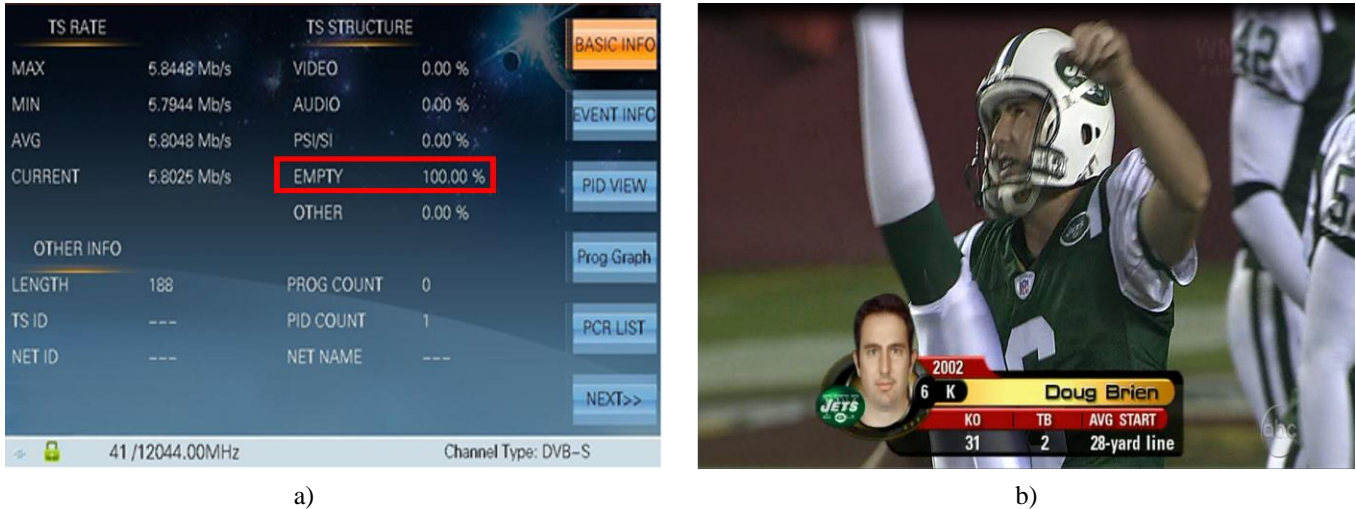


Figura 3

Tramas de transporte utilizadas en la validación del sistema. a) nulo.ts, b) football.ts.

3.- ANÁLISIS DEL ESTÁNDAR DVB-S

El estándar para DVB-S[1] provee un diseño de referencia para la implementación del receptor (Figura 4).

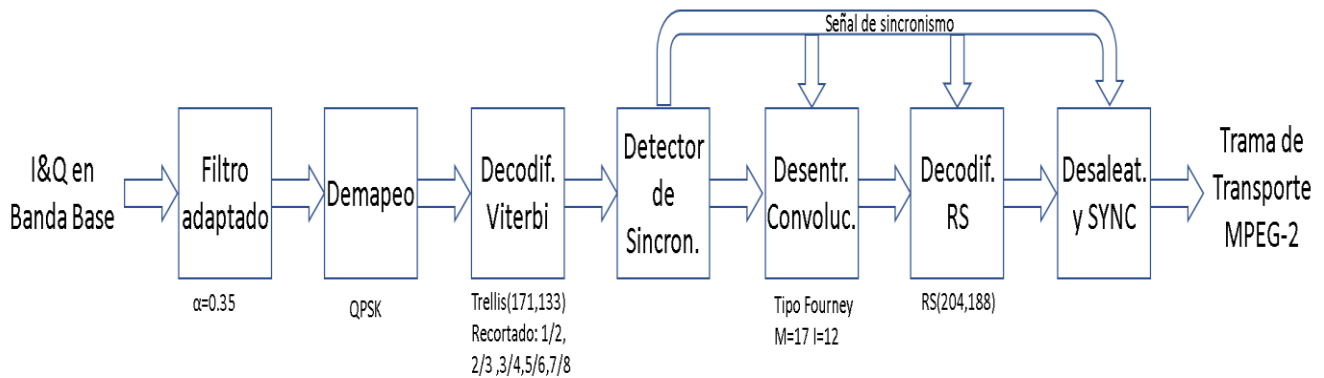


Figura 4

Estructura del receptor DVB-S

A la entrada del sistema receptor, se encuentra la etapa de procesamiento de la señal en la frecuencia intermedia. En esta etapa se produce el traslado de la primera FI, que se encuentra entre los 950 y 2160 MHz a una segunda FI en 70 o 140 MHz. Las señales I y Q analógicas son digitalizadas mediante un Conversor Análogo Digital (ADC) y suministradas a un filtro adaptado. Este filtro está basado en un filtro de raíz de coseno alzado similar al utilizado en el transmisor con un factor de roll-off de 0.35. Su función es disminuir la interferencia inter símbolo (ISI) ya que su respuesta al impulso es cero para todos los tiempos de muestreo, excepto el deseado. En condiciones de un muestreo ideal, la señal es recuperada en su totalidad

Luego de la etapa de filtrado la señal es demapeada a valores binarios teniendo en cuenta las características de una constelación QPSK con codificación Gray. En dependencia del método de demapeo utilizado, la salida del sistema puede ser booleana o estar basada en valores de razón de verosimilitud (LLR) que calculan la probabilidad de cada bit de la dupla IQ, en cada punto de la constelación, recibida sea '1' o '0' de acuerdo a su posición con respecto a la constelación de referencia.

A continuación, la señal es decodificada mediante el decodificador interno basado en el algoritmo de Viterbi. A la salida del codificador interno o convolucional en el transmisor suele aplicarse una técnica de recortado, eliminando cíclicamente bits específicos, con el objetivo de disminuir la razón de símbolos a costa de un peor desempeño de la Razón de Bit Erróneo en el receptor. El decodificador en el receptor es capaz de asimilar esta información faltante en tanto posea en su entrada un flujo igual a la salida del transmisor, con los bits recortados sustituidos por '0' e información sobre su posición. Esto hace necesario contar con la información sobre la razón de código transmitida para rellenar esta información faltante. Sin embargo, existen métodos para prescindir de esta información y obtenerla mediante un proceso iterativo [19]. El estándar recomienda un BER máximo igual a $2 \cdot 10^{-4}$ para un E_b/N_0 igual a 5.5dB para garantizar una tasa cuasilibre de error (QEF) menor a $1 \cdot 10^{-10}$.

Luego de efectuada la decodificación convolucional, ocurre la sincronización de trama utilizando el patrón, establecido por el aleatorizador en el transmisor, de un byte de sincronismo invertido cada 8 paquetes de la trama. Además, en esta etapa se detecta la ambigüedad de π [1], producida por una inversión de la fase en el oscilador local con respecto a la del transmisor y que no puede ser corregida por el decodificador Viterbi.

A continuación, la información es desentrelazada, lo cual distribuye las ráfagas de errores que pudiera haber entre los paquetes siguientes. El esquema de desentrelazado es de tipo Fourney y está basado en una FIFO con 12 ramas de registros y una profundidad de 17 bytes. La entrada y salida se encuentran sincronizadas y el byte de sincronismo de cada paquete siempre debe ser transportado por la rama 0 que no introduce demoras.

La trama desentrelazada es corregida a nivel de bytes mediante el decodificador Reed-Solomon (204,188), retornando los paquetes de la trama a su longitud inicial de 188 bytes correspondiente a un paquete del flujo de trama MPEG-2 [18]. Luego de la decodificación Reed Solomon, la dispersión de energía y la inversión del byte de sincronismo a su valor son removidas utilizando como referencia para el inicio de cada bloque la sincronización recuperada en la etapa de sincronización de trama.

4.- DISEÑO PROPUESTO

En la elaboración del sistema se han seguido dos pautas. La primera consiste en la elaboración del modelo utilizando únicamente los bloques proporcionados por el entorno HDL de Simulink. En la segunda, los bloques con mayores requerimientos de recursos, son sustituidos por módulos IP equivalentes del Xilinx Core Generator empleando Blackbox. La frecuencia de reloj requerida es de 252 MHz lo cual garantiza, para un diseño basado en un único reloj, una razón de símbolo de 21 MSymb/s y una razón de bit a su salida de 29 Mb/s. Esta razón de bit garantiza el trabajo con hasta 7 flujos de videos en definición estándar suponiendo 4 Mb/s para cada uno.

El diseño propuesto (Figura 5) está basado en [1]. Los colores se corresponden con los diferentes requerimientos de temporización del sistema.

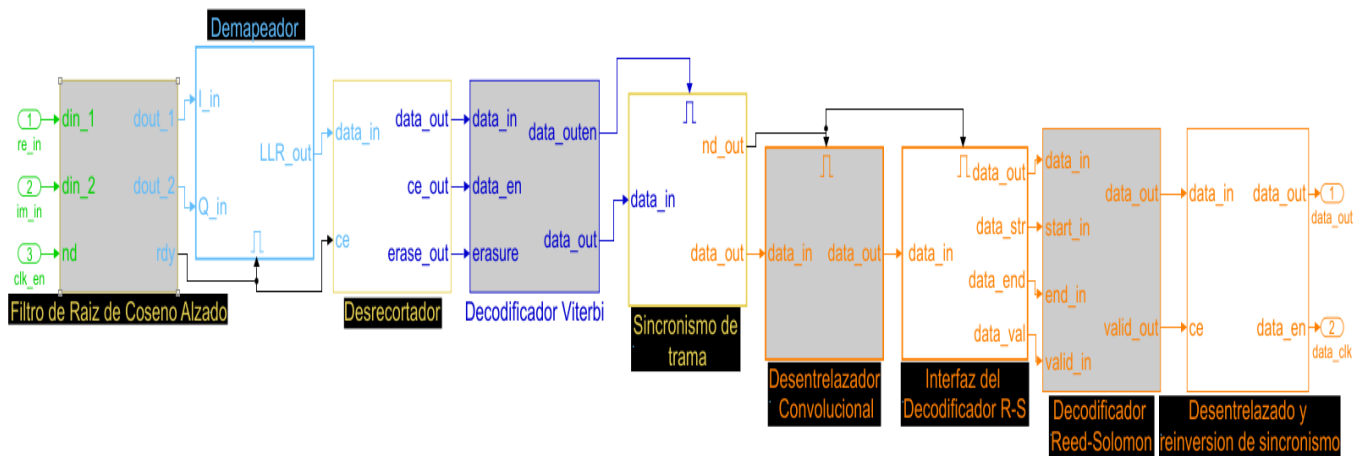


Figura 5

Diagrama en bloques del diseño propuesto del receptor DVB-S con bloques nativos de Simulink.

En gris se representan los bloques que poseen uno similar dentro de los bloques proporcionados por Simulink o Xilinx. Los bloques utilizados, de cada una de las plataformas, se presentan en la siguiente tabla.

Tabla 1
Bloques equivalentes utilizados en el diseño de algunas etapas del receptor

Nombre	HDL Nativo Simulink	Xilinx Core Generator
Filtro de raíz de coseno alzado	FIR Rate Conversion HDL Optimized	FIR Compiler v.5.0
Decodificador Viterbi	Viterbi Decoder	Viterbi Decoder v.7.0
Desentrelazador Convolucional	Deconvolutional Interleaver	Interleaver/Deinterleaver v.6.0
Decodificar Reed-Solomon	Integer RS Decoder	Reed Solomon Decoder v.7.1

En el caso de los restantes bloques del sistema, no se cuenta con un bloque en particular que desarrolle la característica inherente por lo que deben ser diseñados a partir de cero partiendo de bloques discretos proporcionados por Simulink. Además, los bloques nativos y de Xilinx usualmente poseen diferentes requerimientos de entrada y salida, por lo que necesario que los bloques adyacentes deben ser diseñados con estas particularidades. Se debe garantizar que los valores de entrada a cada uno de los bloques del sistema sean iguales, con el propósito de realizar comparaciones. Estos bloques han sido implementados de la siguiente manera.

- Bloque de demapeado

Para la etapa de demapeo se utilizaron las características de la modulación empleada en el estándar DVB-S: QPSK con codificación Gray. Existen dos tipos principales de demapeo: *hard*, el cual utiliza un sistema de umbrales que no tiene en cuenta la varianza del canal, y *soft*, el cual utiliza el cálculo de la varianza de la señal de entrada para calcular el LLR para cada una de las ramas. En el presente artículo se utiliza el algoritmo simplificado para el cálculo del LLR [20]. Este algoritmo evita el uso de cálculos exponenciales que usualmente requieren de mayor complejidad computacional.

La herramienta HDL para la versión de MATLAB no soporta la división por valores distinto a potencias de 2. Por esta razón, se utiliza el cálculo del recíproco del divisor mediante el método de Newton Raphson y a continuación se multiplica por el dividendo, lográndose una operación equivalente a la división.

- Bloque desrecortador

Como se comentó, la salida del bloque demapeador puede encontrarse recortada por lo que es necesario rellenar esta información, previo al bloque de decodificación Viterbi. En el presente artículo, se utiliza una razón de código de 3/4 ya que actualmente es la utilizada por el transponder donde se encuentra insertado Cubavisión Internacional en el satélite HISPASAT 1C. Se emplea la siguiente estructura para el proceso de rellenado con '0' teniendo en cuenta el patrón de recortado correspondiente.

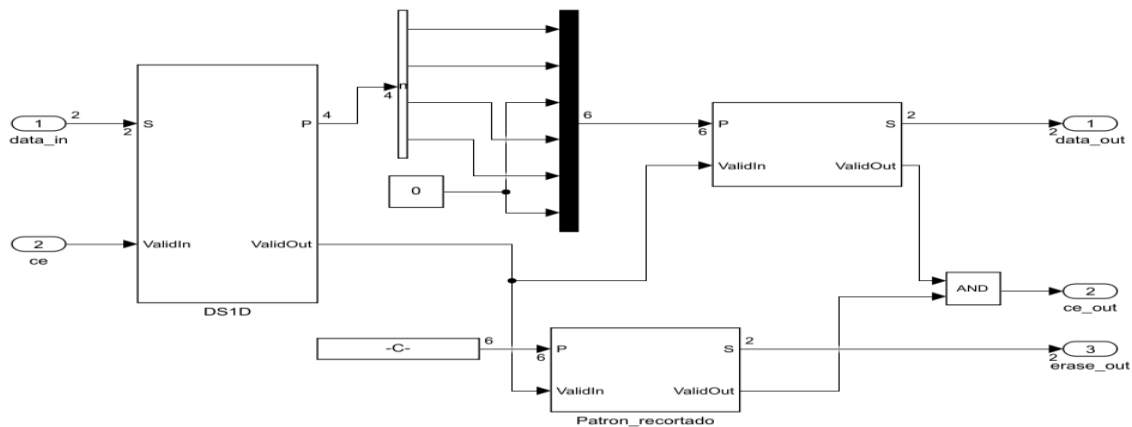


Figura 6

Diagrama del diseño propuesto de desrecortador para una razón de código de 3/4.

- Bloque de obtención de sincronismo de trama.

Como se explicó anteriormente, el sistema DVB-S no cuenta con un sistema de cabecera que le permita sincronizarse con las tramas MPEG. El byte de sincronismo al inicio de cada paquete es utilizado con fines de sincronización dentro del sistema receptor, pero este solo puede ser obtenido a la salida del decodificador Viterbi. Además, la falta de una cabecera, no le permite al sistema detectar errores en la fase del oscilador local ante una ambigüedad π o 180° , ya que el decodificador Viterbi entiende a ambas fases como válidas. La inversión de fase, genera que los datos a la salida del decodificador sean los mismo que en caso de una correcta sincronización, pero con su valor negado o invertido. Estas dos cuestiones son solucionadas por el detector de sincronismo de trama, aunque en [21] Fu, propone un sistema para la detección de esta ambigüedad en una etapa anterior al decodificador. Para la detección del sincronismo se utiliza el patrón proporcionado por la inversión del byte de sincronismo cada 8 paquetes MPEG-2. La longitud de un paquete es de 1632 bits (204 bytes).

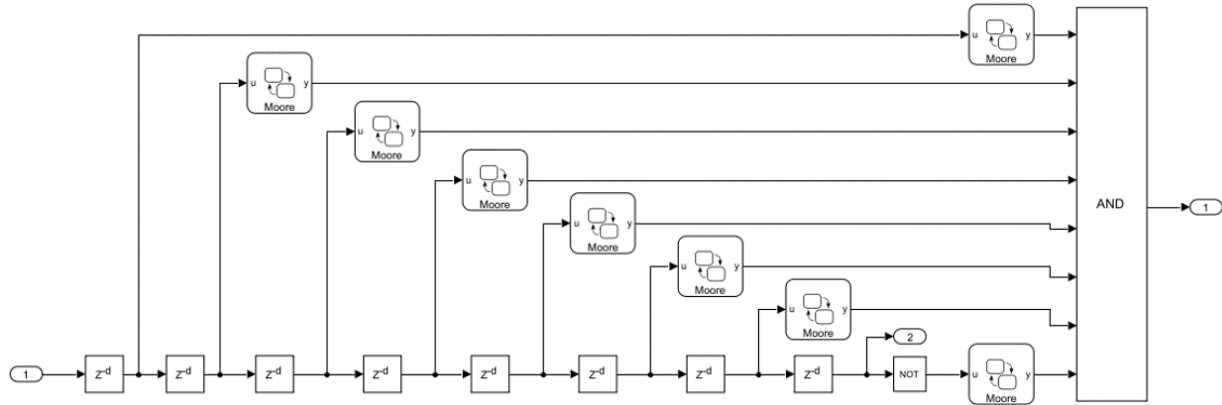


Figura 7

Diagrama del diseño propuesto con máquinas de estado para una de las ramas.

Para el diseño, se utiliza el modelo propuesto por Zhiqiang [22] basado en máquinas de estado. El modelo está constituido por dos ramas. Una de ellas se encarga de la detección del sincronismo suponiendo un patrón normal con un byte de sincronismo invertido (0xB8h) seguido de siete normales (0x47h). La otra detecta la ambigüedad π , para un patrón inverso de uno normal y siete negados. Basado en la simetría de ambos patrones, ambas ramas son idénticas. Cada máquina de estado está diseñada para detectar el byte 0x47h, teniendo la encargada de detectar la ambigüedad su entrada invertida mediante una compuerta NOT. En ambos casos se comparte las demoras entre bytes de sincronismo. En la Figura 7, se observa el diseño de la rama suponiendo que la fase del oscilador está sincronizada con la del receptor. En este caso las primeras 7 máquinas de estado detectan el byte de sincronismo 0x47h, mientras la última detecta el byte invertido 0xB8h. Los bloques de demora, garantizan la distancia de 1632 bits entre los bytes sincronismo. En caso de detección del patrón, se genera un pulso a la salida de la compuerta lógica AND para indicar al resto del sistema el inicio del flujo sincronizado en la salida 2.

Para el ajuste de la señal en caso de detección de la inversión de fase del local se invierte la salida 2 de este bloque, en vez de ajustar la fase del oscilador. Con esta solución se evitan complicaciones derivadas de un cambio brusco en los sistemas de sincronización previos al receptor y los errores correspondientes. Las demoras entre cada byte de sincronismo correspondiente a registros de desplazamiento de 1632 bits ($204 * 8$ bits), son mapeadas automáticamente por el HDL Coder a una RAM con lo cual se reducen los recursos necesarios para este módulo.

Respecto a la interfaz con el bloque de desentrelazado convolucional, hay una variante para cada una de las versiones de bloque utilizada. En el caso de utilizar un bloque HDL nativo de Simulink, solamente son necesaria la salida de dato de un byte, que se implementa mediante un conversor serie paralelo, y una señal de habilitación que coincide con el inicio del flujo válido. Para el bloque Xilinx, a estas dos señales se adiciona una señal de inicio de trama, la cual es requerida para resetear los registros del bloque de desentrelazado y apuntar a su primera rama. Esta señal es implementada mediante un detector de frente de subida en la señal de habilitación.

- Bloque de desaleatorización y reinversión del byte de sincronismo invertido

En este bloque se realiza la desaleatorización de la salida del decodificador Reed Solomon mediante un producto XOR con un código pseudoaleatorio sincronizado y cíclico. Para lograr la reinversión del primer byte de sincronismo de cada bloque de 8 paquetes, se incorpora el byte 0xFFh al inicio del código pseudoaleatorio.

La generación del código pseudoaleatorio puede ser abordado desde dos perspectivas. Se puede utilizar el bloque Generador de secuencia PN para generar el código pseudoaleatorio. Sin embargo, la salida de estos es de un bit mientras la entrada al sistema es de un byte. Esto podría ser resuelto mediante el uso de un acumulador para realizar la conversión bit a byte, pero también requeriría el uso de dos temporizaciones, lo que hace el sistema más complejo para la sincronización.

Otra perspectiva es el uso de los valores pseudoaleatorios almacenados en memoria en formato byte. De esta forma se puede utilizar la misma temporización de entrada de datos. Esta última, es la opción utilizada en el modelo (Figura 8).

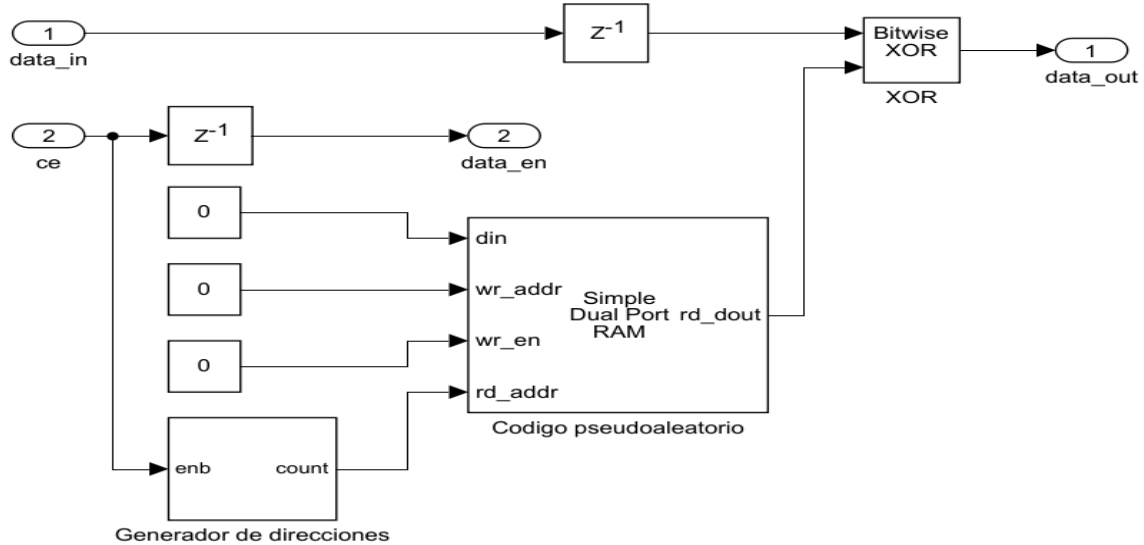


Figura 8

Diagrama del diseño propuesto para el módulo de desaleatorización y reinversión de sincronismo.

La generación de las direcciones se genera mediante un contador habilitado por la señal de validez de los datos de salida del decodificador Reed Solomon.

5.- RESULTADOS

5.1.- RESULTADOS DE LAS PRUEBAS DE VALIDACIÓN

La primera prueba tiene como objetivo evaluar el comportamiento del sistema ante el Ruido Aleatorio Gaussiano (AWGN) El objetivo es obtener un resultado mejor que el descrito como mínimo en el estándar [1] de $2 \cdot 10^{-4}$ para un E_b/N_0 de 5.5 dB.

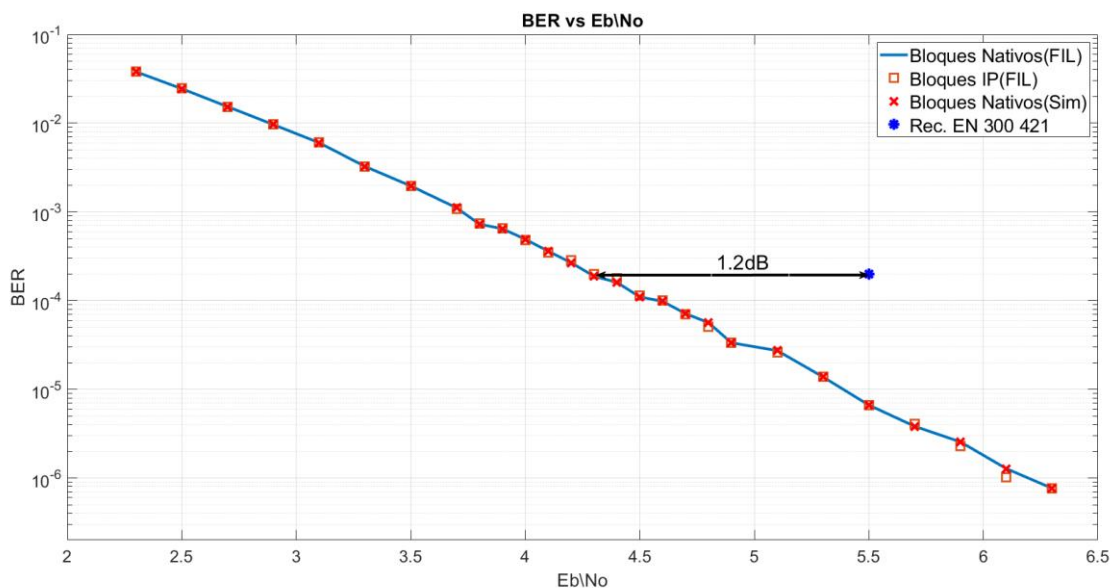


Figura 9

Comportamiento del BER posterior al bloque Viterbi

La Figura 9 muestra el comportamiento ante ruido aleatorio gaussiano (AWGN, por sus siglas en inglés) de los dos modelos desarrollados. Se presentan los resultados de la simulación dentro del entorno Simulink del sistema propuesto y se comparan con las soluciones implementadas en FPGA mediante la herramienta FPGA-in-the-Loop. Se observa que los resultados entre

el modelo en Simulink y el implementado en FPGA con bloques nativos coinciden en su comportamiento. En el caso del bloque implementado con bloques IP de Xilinx, los resultados son similares.

De estos resultados, se puede deducir que no existen grandes diferencias entre el comportamiento de ambas implementaciones. Además, los resultados del sistema modelado presentan un BER post-Viterbi de $2e-4$, para 4.3 dB. Este valor es 1.2dB mejor que el recomendado en [1].

Para la segunda prueba se utilizan los ficheros de prueba generados mediante hardware especializado. El primero fichero utilizado es una trama de transporte con todos sus paquetes configurados como nulos. Esto permite determinar los posibles errores de sincronización que pudiesen ocurrir. Para el segundo caso se utiliza una trama de transporte real. Se espera una correcta decodificación en ambos casos.

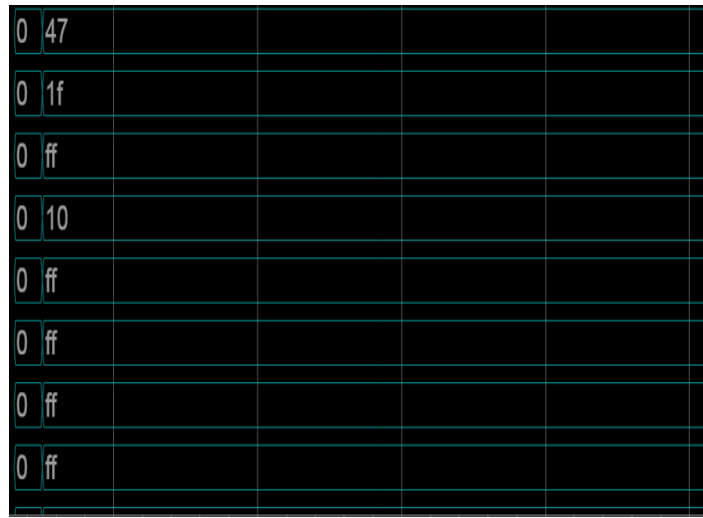


Figura 10

Validación de la correcta codificación de la trama de transporte ‘nulo.ts’ utilizando FIL

En la Figura 10 se observa el resultado de la decodificación de la información generada a partir del fichero ‘nulo.ts’. Los valores obtenidos se corresponden con la información de un paquete nulo según lo estipulado en [18].

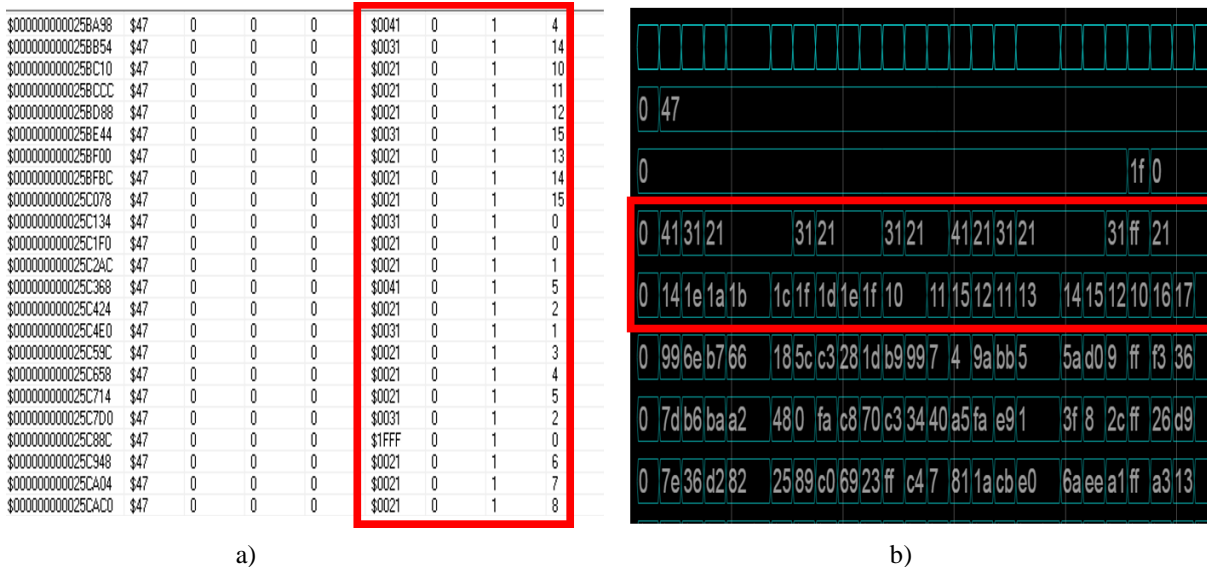


Figura 11

Validación de la correcta codificación de la trama de transporte ‘football.ts’ utilizando FIL. a) Datos a la entrada del transmisor b) Datos a la salida del receptor.

Los resultados para la variante con el fichero 'football.ts' se presentan en la Figura 11. Se muestra la correspondencia entre las cabeceras de los datos transmitidos y los datos decodificados mediante el sistema modelado. Con esto se puede determinar la validez del sistema como un receptor DVB-S según lo estipulado en el estándar [1].

5.2.- RESULTADOS DE LA MODELACIÓN

En la Tabla 2 se muestran los resultados de la modelación de ambas alternativas en un FPGA Virtex-6 XC6VLX240T.

Tabla 2
Recursos utilizados por las implementaciones

Recurso	HDL Nativo Simulink	%	Xilinx Core Generator	%	Disponibles
Registros	14102	4	6959	2	301440
LUT	16641	11	6321	4	150720
LUTRAM	677	1	531	1	58400
FF	4685	12	2560	6	37680
BRAM18	5	1	7	1	832
BRAM36	4	1	3	1	416
DSP	75	9	49	6	768
f_{clkmax}	264 MHz		291 MHz		

Se puede observar que el diseño utilizando los bloques HDL nativos de Simulink utiliza en la mayoría de los casos el doble que su contraparte con bloques del Core de Xilinx. Este es un resultado esperado, pues en el primer caso los bloques están optimizados para la plataforma, mientras en el segundo es código equivalente genérico con el objetivo de ser portable y multiplataforma. En el caso de la frecuencia máxima de reloj el resultado es similar, permitiendo la opción Xilinx mejores resultados.

6.- CONCLUSIONES

En el presente artículo se ha presentado el modelado y la validación de un receptor en punto fijo basado en el estándar DVB-S dentro del entorno MATLAB/Simulink. Además, fue elaborado un código HDL equivalente mediante la herramienta HDL Coder de MATLAB. Este código fue implementado en FPGA y validado utilizando la herramienta FPGA-in-the-Loop. Para el diseño del sistema se utilizan tanto bloques HDL propios de la plataforma Simulink como módulos IP generados con el Xilinx Core Generator en un esquema no descrito en la literatura. Fue realizada una comparación entre los módulos HDL nativos de Simulink y los módulos optimizados del Xilinx Core Generator donde se observa que el esquema propuesto con bloques IP del Xilinx Core Generator, requiere de un menor número de recursos y permite trabajar con una mayor frecuencia de reloj, lo que redundaría en una mayor tasa de símbolos permisibles en su entrada. El modelo diseñado con bloques HDL nativos de Simulink cumple con los objetivos propuestos, por lo que se demuestra su validez para la realización de modelos de altas prestaciones y con altas cargas computacionales con la ventaja de su enfoque multiplataforma, lo cual permite su implementación en otras plataformas.

REFERENCIAS

1. ETSI. Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for 11/12 GHz satellite services. ETSI Standard En 300 421 [Internet]. 1997;1.1.2:1-24. Disponible en: https://www.etsi.org/deliver/etsi_en/300400_300499/300421/01.01.02_60/en_300421v010102p.pdf
2. ETSI. Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation

- systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2). ETSI Standard EN 302 307 [Internet]. 1.3.1. 2012;1.3.1:1-85. Disponible en: https://www.etsi.org/deliver/etsi_en/302300_302399/302307/01.03.01_20/en_302307v010301a.pdf
3. Pisciotto I, Pastina D, Cristallini D. DVB-S based passive radar imaging of ship targets. En: 2019 20th International Radar Symposium (IRS) [Internet]. IEEE; 2019. p. 1-7. Disponible en: <https://ieeexplore.ieee.org/document/8768097/>
 4. Bezanilla AF, Alonso AM. Análisis de interferencia co-canal en sistemas DVB-S. En: 1er Foro Internacional de TVD [Internet]. 2013. Disponible en: http://www.lacel.com/index.php/evento/Gestionar_Trabajo_Evento/1/Trabajos/1
 5. Shah RK. Executable Model Based Design Methodology for Fast Prototyping of Mobile Network Protocol : A Case Study on MIPI LLI. Master's thesis [Internet]. Indraprastha Institute of Information Technology; 2014. Disponible en: <https://repository.iiitd.edu.in/xmlui/handle/123456789/172>
 6. Venkataramani G, Kintali K, Prakash S, van Beek S. Model-based hardware design. En: 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) [Internet]. IEEE; 2013. p. 69-73. Disponible en: <http://ieeexplore.ieee.org/document/6691099/>
 7. Kintali K, Gu Y. Model-Based Design with Simulink, HDL Coder, and Xilinx System Generator for DSP [White paper] [Internet]. Natick, MA, USA; 2015. Disponible en: https://la.mathworks.com/content/dam/mathworks/tag-team/Objects/x/86457_92077_v01_Xilinx_WhitePaper.pdf
 8. Järviuoma J. Rapid prototyping from algorithm to FPGA prototype. Master's Thesis [Internet]. University of Oulu; 2015. Disponible en: <http://jultika.oulu.fi/Record/nbnfioulu-201512052252>
 9. Cui G. A Driving Assistance System with Hardware Acceleration. Master's Thesis [Internet]. Chalmers University of Technology; 2015. Disponible en: <https://www.semanticscholar.org/paper/A-Driving-Assistance-System-with-Hardware-Cui/03a441366bfc26fd9723d13213375b5e93fb7dee>
 10. Hai JCT, Pun OC, Haw TW. Accelerating video and image processing design for FPGA using HDL coder and simulink. En: 2015 IEEE Conference on Sustainable Utilization And Development In Engineering and Technology (CSUDET) [Internet]. IEEE; 2015. p. 1-5. Disponible en: <http://ieeexplore.ieee.org/document/7446221/>
 11. Besbes M, Said SH, M'Sahli F. FPGA implementation of high gain observer for induction machine using Simulink HDL coder. En: 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT) [Internet]. IEEE; 2015. p. 1-6. Disponible en: <http://ieeexplore.ieee.org/document/7233064/>
 12. Cai X, Zhou M, Huang X. Model-Based Design for Software Defined Radio on an FPGA. IEEE Access [Internet]. 2017;5:8276-83. Disponible en: <http://ieeexplore.ieee.org/document/7895173/>
 13. Pasha SG, Kohir V V. OFDM based DVB-T system implementation using MATLAB and HDL coder. En: 2017 International Conference on Computing Methodologies and Communication (ICCMC) [Internet]. IEEE; 2017. p. 256-61. Disponible en: <http://ieeexplore.ieee.org/document/8282686/>
 14. Sarge VY. Evaluating Simulink HDL Coder as a Framework for Flexible and Modular Hardware Description. Master's Thesis [Internet]. Massachusetts Institute of Technology; 2018. Disponible en: <https://dspace.mit.edu/bitstream/handle/1721.1/119717/1078637048-MIT.pdf?sequence=1&isAllowed=y>
 15. Shah RK, Kumar T, Fell A, Dohadwala MS, Malik R. Executable model based design methodology for fast prototyping of mobile network protocol: A case study on MIPI LLI. En: 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN) [Internet]. IEEE; 2017. p. 346-51. Disponible en: <http://ieeexplore.ieee.org/document/8049972/>
 16. Pant H, Bourai H, Rana GS, Yadav SC. Conversion of MATLAB code in VHDL using HDL Coder & Implementation of Code on FPGA. HCTL Open IJTIR [Internet]. 2015;14:1-9. Disponible en: https://www.researchgate.net/post/How_to_generate_VHDL_code_from_Matlab_Code
 17. Díaz R, Fontes E. Modelo de referencia de un modulador de la norma DTMB de Televisión Digital. Rev Telem@tica [Internet]. 2015;14(2):38-48. Disponible en: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/189>
 18. ITU-T H.222 | ISO/IEC. Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems. ISO/IEC Standard ISO/IEC 13818-12019 [Internet]. 2019; Disponible en: <https://www.iso.org/standard/75928.html>
 19. Yi Z, Li Y. An Efficient Convolutional Code Information Recovery Scheme in DVB-S. En: 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing [Internet]. IEEE; 2008. p. 1-4. Disponible en: <http://ieeexplore.ieee.org/document/4678716/>
 20. Jang Woong Park, Chang Duk Ryu, Myung Hoon Sunwoo, Pan Soo Kim, Dae-Ig Chang. Simplified soft-decision

demapping algorithm for DVB-S2. En: 2009 International SoC Design Conference (ISOCC) [Internet]. IEEE; 2009. p. 444-7. Disponible en: <http://ieeexplore.ieee.org/document/5423908/>

21. Fu J, Tang Y. A Parity Check Node Synchronization Based Method for Solving Phase Ambiguity Problem in DVB-S Systems. En: 2009 Fifth International Conference on Information Assurance and Security [Internet]. IEEE; 2009. p. 463-6. Disponible en: <http://ieeexplore.ieee.org/document/5283224/>
22. Zhiqiang Y, Yun L. The Research of Sync-search Scheme in DVB-S Receiver. En: 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing [Internet]. IEEE; 2010. p. 377-80. Disponible en: <http://ieeexplore.ieee.org/document/5480875/>

CONFLICTO DE INTERESES

Ninguno de los autores manifestó la existencia de posibles conflictos de intereses que debieran ser declarados en relación con este artículo.

CONTRIBUCIONES DE LOS AUTORES

Alejandro González García: contribución importante a la idea y diseño del estudio, la revisión bibliográfica, su análisis e interpretación. Redacción del borrador del artículo y de su versión final.

Reinier Díaz Hernández: contribución al diseño del estudio. Revisión del borrador del artículo y su versión final.

AUTOR

Alejandro González García, Ingeniero en Telecomunicaciones y Electrónica desde 2013. Investigador con categoría científica “Aspirante a Investigador” desde septiembre de 2016, trabaja en **LACETEL**, Instituto de Investigación y Desarrollo de Telecomunicaciones. La Habana, Cuba. Actualmente es maestrante de la X Edición de la Maestría en Sistemas Digitales, alejandro@lacetel.cu. ORCID: <http://orcid.org/0000-0002-8517-4848>

Reinier Díaz Hernández, MsC., Ing. en Telecomunicaciones y Electrónica, graduado en 2007 en la Universidad Tecnológica de La Habana (CUJAE). Se desempeña como investigador en LACETEL, La Habana, Cuba, desde octubre 2009; en agosto de 2016 obtiene la categoría de “Investigador Agregado”. En febrero de 2017 concluye su tesis de Maestría y obtiene la categoría que lo acredita como Máster en Ciencias (MsC.). reinier@lacetel.cu ORCID: <http://orcid.org/0000-0001-9439-4714>



Esta revista se publica bajo una [Licencia Creative Commons Atribución-No Comercial-Sin Derivar 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)