

Diseño e implementación en arquitecturas reconfigurables de etapa de codificación de canal para comunicaciones satelitales en radiofrecuencias

Mario Axel Alva Díaz, Dr. Raúl Castillo Pérez, M. en C. Gabriela Aurora Yáñez Casas y Dr. Jorge Javier Hernández Gómez.
ESIME Zacatenco, ESIME Zacatenco, Centro de Desarrollo Aeroespacial del IPN y Centro de Desarrollo Aeroespacial del IPN.

Email: malvad1400@alumno.ipn.mx, rcastillo@ipn.mx, gyanezca@ipn.mx, y jjhernandezgo@ipn.mx

URL ORCID: RCP: 0000-0003-4131-4842

Resumen— Los satélites artificiales y sus servicios han evolucionado a lo largo de los años; la reducción de sus dimensiones es uno de los cambios principales. Dicha reducción ha hecho necesario desarrollar tecnología nueva que cumpla con la normatividad aplicable a la prestación de servicios satelitales. Aquí se plantea el desarrollo de un sistema codificador de canal (con cinco métodos para una señal de 32 bits), enfocado para satélites tipo CubeSat, mediante el uso de un dispositivo FPGA (Field Programmable Gate Array); con ello la etapa de comunicación, codificador de canal, en satélites CubeSat no es eliminada (debido a limitaciones físicas) y se reduce la probabilidad de error en la comunicación. Para la solución del problema se han seleccionado cinco métodos diferentes de codificación a emplear en el FPGA: NRZ (Non-Return-to-Zero), AMI (Alternate Mark Inversion), Manchester, Miller y HDB3 (High Density Bipolar of order 3). Cada codificación ofrece distintas ventajas y en general todas son relativamente simples (circuito sencillo de pocos elementos). En primer lugar, se ha escrito un programa en Matlab que permite visualizar la codificación de una señal de 32 bits, para cada uno de los cinco métodos seleccionados. A continuación, se describe con el lenguaje VHDL (Very High Speed Integrated Circuits) y usando el programa ISE Design Suite 14.7 cada codificación seleccionada, por separado, para más adelante unir cada descripción en un solo programa; cada programa es revisado mediante simulación. Los diagramas de tiempo generados permiten determinar que las codificaciones han sido descritas correctamente.

Palabras Clave — Codificador digital, diseño de hardware, lenguaje VHDL, satélites CubeSat, señal de 32 bits.

Abstract— Artificial satellites and their services have evolved over the years; The reduction of its dimensions is one of the main changes. Such reduction has made it necessary to develop new technology that complies with the regulations applicable to the provision of satellite services. Here the development of a channel encoder system (with five methods for a 32-bit signal), focused on CubeSat-type satellites, using an FPGA device (Field Programmable Gate Array) is stated; With this, the communication stage, channel encoder, in CubeSat satellites is not eliminated (due to physical limitations) and the probability of error in communication is reduced. To solve the problem, five different coding methods have been selected to be used in the FPGA: NRZ (Non-Return-to-Zero),

AMI (Alternate Mark Inversion), Manchester, Miller y HDB3 (High Density Bipolar of order 3). Each coding offers different advantages and in general all are relatively simple (simple circuit with few elements). Firstly, a program has been written in Matlab that allows viewing the encoding of a 32-bit signal, for each of the five selected methods. Next, each selected encoding is described with the VHDL (Very High Speed Integrated Circuits) language and using the ISE Design Suite 14.7 program, separately, to later join each description into a single program; each program is reviewed through simulation. The generated timing diagrams allow us to determine that the encodings have been described correctly.

Key Words — 32-Bit signal, CubeSat satellites, digital encoder, hardware design, VHDL language.

I. INTRODUCCIÓN

Las comunicaciones satelitales ofrecen una gran variedad de servicios: meteorológico, de investigación espacial, satelital de exploración de la Tierra, satelital fijo, satelital móvil, entre otros servicios [1]. Desde el lanzamiento del primer satélite artificial, el Sputnik-1, la tecnología satelital ha sufrido grandes evoluciones. Uno de dichos cambios son las dimensiones con las que se diseñan los satélites, como se muestra en la **Tabla 1** [1].

Tabla 1. Clasificación y características de satélites de pequeño tamaño.

Denominación	Masa (kg)	Dimensiones máximas (m)	Tiempo de desarrollo (años)
Minisatélite	100-500	3-10	3-10
Microsatélite	10-100	1-5	2-5
Nanosatélite	1-10	0.1-1	1-3
Picosatélite	0.1-1	0.05-0.1	1-3
Femto-satélite	<0.1	0.01-0.1	1

Las dimensiones del satélite implican limitantes físicas que deben de ser resueltas para evitar afectaciones en la operación o servicio del satélite. La reducción del tamaño tiene ventajas

como lo son costos y tiempos de desarrollo menores. Sin embargo, se tienen también desventajas. Por ejemplo, la cantidad de potencia disponible es en general menor en cuanto sea menor el tamaño del satélite, y en general se cuenta con menos espacio para la carga útil [1]. Debido al menor espacio, una solución comúnmente empleada es la eliminación de ciertas etapas o elementos de un sistema, como puede ser en el sistema de comunicaciones. Para los satélites de tipo CubeSat (normalmente con dimensiones de $10 \times 10 \times 10 \text{ cm}^3$) el sistema de transmisión se ve afectado al no incluirse un codificador de canal que permita la detección y/o corrección de errores (por lo tanto, se tiene una mayor probabilidad de error [2]). Hoy en día no se puede permitir brindar servicios que estén por debajo de ciertos estándares, ya que instituciones como, por ejemplo, la ITU se encargan de regular toda una serie de normas que se deben de cumplir [3]. Por ello se debe de encontrar una solución como que, para este caso, no se elimine la etapa de codificación de canal en los transmisores de los satélites CubeSat, y así no se vea afectado el sistema de comunicaciones.

Por otra parte, la modulación codificada Trellis, también llamada codificación Trellis, es un ejemplo del uso de la codificación de canal. Mediante dicho método la información se codifica con una técnica convolucional que añade redundancia a la señal. Posteriormente se modula (usando modulación ASK, FSK, PSK, QAM, u otra técnica de modulación digital) y tal procedimiento tiene como resultado disminuir la probabilidad de error y mantener el ancho de banda. La codificación Trellis ha sido usada en módems telefónicos para aumentar la tasa de bits y disminuir los errores en la comunicación [2].

Dentro de la investigación realizada, prácticamente no se ha encontrado material donde se apliquen las técnicas de codificación de canal que aquí se proponen para satélites CubeSat, usando dispositivos FPGA. Por lo tanto, es una gran oportunidad el estudiar las ventajas de dicha etapa de comunicación. Un dispositivo FPGA es relativamente pequeño y consume poca potencia, características que los hacen adecuados para las especificaciones que deben de cumplir los satélites tipo CubeSat. Se propone el uso de diferentes métodos de codificación de canal ya que cada uno tiene sus ventajas y desventajas, pero en general se usan técnicas sencillas que se espera usen pocos recursos de la FPGA. Además, un sistema con más de un método de codificación permite una mayor flexibilidad de uso y se podrá usar para diferentes misiones en diferentes satélites o dentro de un mismo satélite que cambie su misión.

Al desarrollar el proyecto en un dispositivo FPGA se obtiene un programa en lenguaje VHDL, fácilmente modificable, para añadir o eliminar métodos de codificación, que puede ser usado en futuras investigaciones que necesiten dicho dispositivo para sus sistemas satelitales.

De tal forma que el objetivo del trabajo es desarrollar un codificador de canal, que sea capaz de usar cinco distintos tipos de codificación (NRZ, AMI, Manchester, Miller y HDB3) para ser empleado en satélites CubeSat mediante un dispositivo FPGA.

II. METODOLOGÍA/DESARROLLO

En primer lugar, se ha escrito un código en Matlab que permite de manera visual observar una señal de 32 bits y su codificación para cada uno de los cinco métodos seleccionados. Más adelante se escribe el código en VHDL. Es esencial entender el funcionamiento o el algoritmo que debe de seguir cada método de codificación para poder describirlo en Matlab y VHDL.

Código NRZ. Mediante este método se representa el símbolo 1 con un pulso cuadrado de amplitud positiva y longitud T , mientras que el símbolo 0 se representa mediante un pulso cuadrado de amplitud negativa y longitud T .

Código AMI. El primer 1 se representa por una transición del nivel $+A$ al nivel 0 (cada uno con una duración de $T/2$ del bit). Los siguientes 1s empiezan con una inversión de polaridad de la amplitud A del 1 anterior. El 0 se representa por el nivel 0.

Código Manchester. La forma de onda es descrita por la representación del símbolo 1 mediante un pulso cuadrado de amplitud positiva durante $T/2$ segundos seguido de un pulso cuadrado de amplitud negativa durante $T/2$. El símbolo 0 se representa de manera inversa, durante los primeros $T/2$ segundos ocurre un pulso cuadrado de amplitud negativa y durante los siguientes $T/2$ de tiempo ocurre un pulso cuadrado de amplitud positiva.

Código Miller. Un 1 es representado por una transición en el punto medio del bit. Un 0 no representa una transición, a menos que siga a un 0 anterior. En este caso la transición empieza en el final del primer cero.

Código HDB3. Limita el número de ceros continuos que se pueden transmitir con una sustitución en cadenas extensivas de 0's, reemplazando el $(n + 1)$ -ésimo cero por una violación bipolar.

Teniendo los conceptos claros y, más específicamente, el algoritmo de cada método de codificación de canal, se escribe un programa en Matlab que grafica una señal de 32 bits (propuesta por el usuario) y sus correspondientes cinco codificaciones.

Para el desarrollo del codificador en lenguaje VHDL se escribió el código en el programa ISE Design Suite 14.7, con la ayuda de la **Figura 1**:

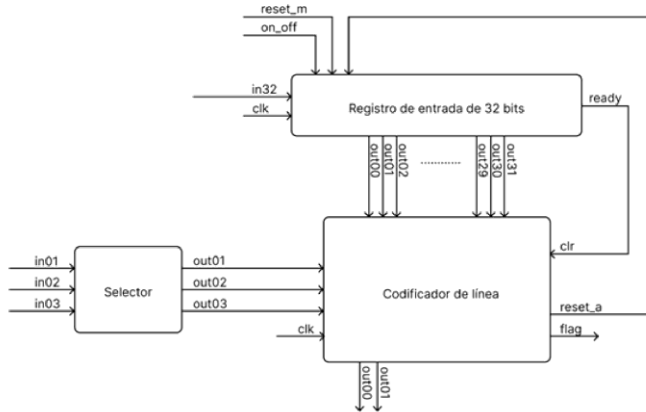


Figura 1.-Diagrama a bloques de codificador de canal.

Se describe el Selector, el Registro de entrada de 32 bits y cada método de codificación como un elemento independiente y al final se interconecta cada componente como se aprecia en la **Figura 1**.

El circuito selector funciona en base a la **Tabla 2** que se muestra a continuación.

Tabla 2.- Tabla de verdad del circuito selector.

Entradas (in01, in02, in03)	Salidas (out01, out02, out03)	Codificación
000	Estado anterior	Ultima codificación seleccionada
001	001	AMI
010	010	HDB3
011	011	Manchester
100	100	Miller
101	101	NRZ
110	Estado anterior	Ultima codificación seleccionada
111	Estado anterior	Ultima codificación seleccionada

Una vez seleccionada la codificación, se pone en alto la señal “on_off” para activar el codificador y se envía la señal de 32 bits de forma serial por “in32”, señal que debe de estar sincronizada con “clk” (señal de reloj). El registro irá guardando los 32 bits de forma serial. Una vez que el registro ha guardado los 32 bits “ready” envía un pulso cuadrado que limpia la entrada del codificador de línea. Luego el mismo lee los 32 bits del registro y los codifica de acuerdo con la codificación seleccionada, a continuación, el codificador transmite los 32 bits codificados de

manera serial en sincronía con la señal de reloj, mediante “out00” y “out01”. Se usan dos bits ya que hay codificaciones que usan tres niveles de amplitud (00 representa 0, 01 representa +A y 10 la amplitud -A). Mientras se transmiten los 32 bits codificados, “flag” se encuentra en alto para indicar la transmisión de los bits. Cuando se terminan de enviar los datos “reset_a” se pone en alto para limpiar el registro. Con “reset_m” el usuario puede limpiar el registro de así desearlo.

Además, se ha desarrollado un circuito que genera señales de 32 bits, con el fin de probar el funcionamiento del codificador. El esquema del generador se muestra en la **Figura 2**.

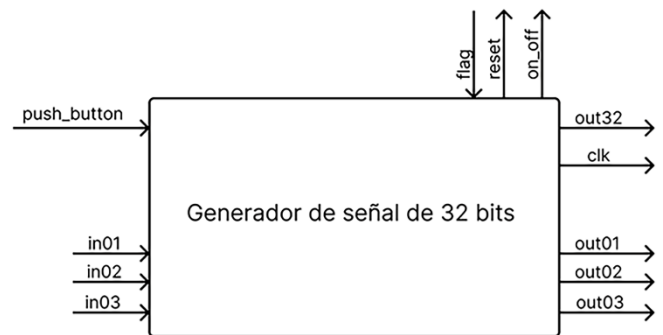


Figura 2.-Diagrama a bloques de generador de señal de 32 bits.

Las entradas “in01”, “in02” y “in03” funcionan igual que en el selector; “reset” se conecta a “reset_m” del codificador (limpia el registro); “on_off” se conecta a la señal del codificador con el mismo nombre y sirve para activar la codificación; “flag” se conecta a la señal del codificador con el mismo nombre, se usa para saber que se están enviando los 32 bits codificados; “out01”, “out02” y “out03” se conectan al selector del codificador. “out32” transmite la señal generada de 32 bits y sincronizada con una señal de reloj que se envía por “clk”. Una vez se detecta un flanco de bajada por “push_button”, el generador empieza su función. Transmitirá todas las posibles señales de 32 bits de forma serial mediante “out32”, comenzando por aquella que tiene un valor de 0 en todos sus bits, hasta aquella que tiene en todos sus bits el valor de 1 (forma ascendente). La señal “flag” indicará en qué momento se envía la siguiente señal (cuando se terminan de transmitir los 32 bits codificados en el codificador) y con la señal “reset” se limpia el registro del codificador antes de enviar la siguiente señal de 32 bits que se codificará.

Los programas escritos en Matlab y VHDL no se proporcionan debido a su gran extensión.

III. RESULTADOS Y DISCUSIÓN

Con ayuda del programa escrito en Matlab es posible determinar el resultado esperado al codificar una señal de 32 bits (en este caso “10111100100001100111010101001101”) para

los cinco métodos de codificación propuestos, como se ejemplifica en la **Figura 3** para la codificación HDB3.

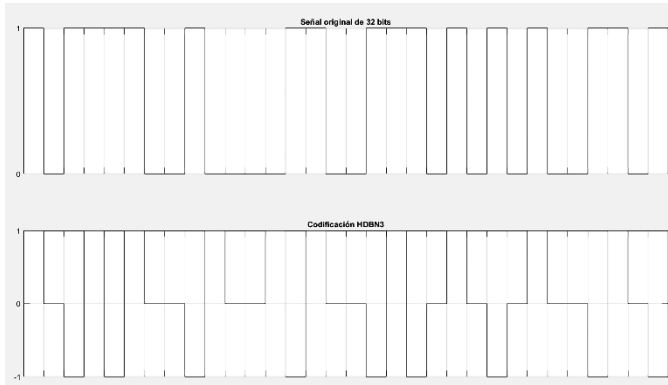


Figura 3.- Gráfica en Matlab de codificación HDB3 de señal de 32 bits.

La simulación en VHDL y el uso del programa en Matlab permiten determinar que el codificador funciona de manera correcta. En la **Figura 4** se muestra un ejemplo de una señal de 32 bits codificada en HDB3. Se muestran la señal de entrada, la señal de salida de 2 bits (recordar que HDB3 usa tres niveles de amplitud), la señal de reloj y, además, se añade un gráfico que representa los niveles de amplitud para los dos bits de salida del codificador. Todo ello demuestra que el codificador funciona como se esperaba.

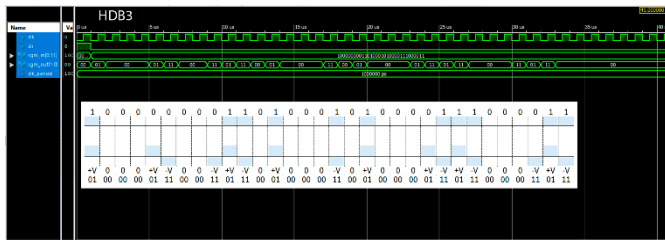


Figura 4.- Simulación en VHDL de codificador usando HDB3 para señal de 32 bits.

Por último, la **Figura 5** muestra solo una parte del circuito sintetizado por VHDL (ya que el circuito es muy extenso para mostrarlo completo), en este caso para la FPGA XC6SLX9 de la familia Spartan 6. Ello demuestra que el programa codificador está listo para ser implementado físicamente en un dispositivo FPGA.

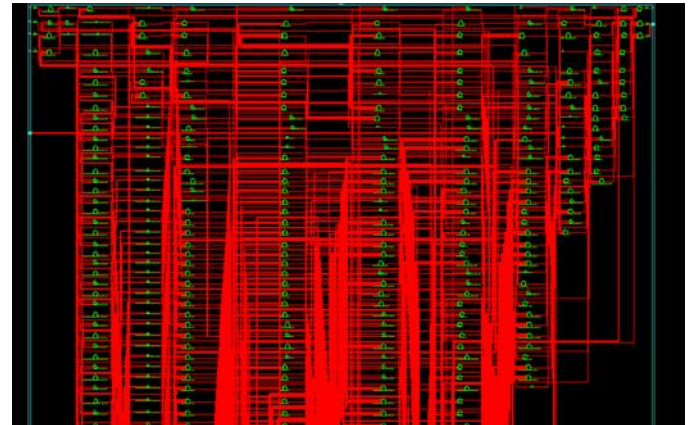


Figura 5.- Parte del circuito codificador sintetizado en VHDL.

IV. CONCLUSIONES

La elaboración del trabajo descrito presenta una alternativa adecuada para el empleo de un sistema de codificación de canal en satélites tipo CubeSat, debido a que los dispositivos por su relativo tamaño pequeño y consumo de potencia son adecuados para dichos satélites. Como se ha demostrado mediante las simulaciones, el código desarrollado en VHDL funciona correctamente para las codificaciones NRZ, AMI, Manchester, Miller y HDB3. Sin embargo, debe de ser probado de manera física el sistema para eliminar cualquier sospecha de funcionamiento en condiciones no ideales. El circuito puede ser empleado en la FPGA XC6SLX9 de la familia Spartan 6. Para otra FPGA se debe de comprobar que es posible sintetizar el sistema codificador: una FPGA con recursos escasos podría no ser capaz de implementarlo.

AGRADECIMIENTOS

Los autores agradecen al Posgrado en Ciencias en Ingeniería de Telecomunicaciones de la Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Zacatenco, y al Centro de Desarrollo Aeroespacial del Instituto Politécnico Nacional por el apoyo brindado. Mario Axel Alva Díaz agradece al CONAHCYT por el apoyo brindado a través de la beca nacional de estudios de maestría. El Dr. Raúl Castillo Pérez agradece el apoyo del CONAHCYT a través del proyecto Frontera 15022, y al Instituto Politécnico Nacional por el apoyo con los estímulos SIBE y EDI.

REFERENCIAS

- [1] ITU Publications, Handbook on Small Satellites, 2023 edition, 2023.
- [2] Wayne Tomasi, Sistemas de Comunicaciones Electrónicas, Cuarta edición, 2003.
- [3] Timothy Pratt y Jeremy Allnutt, Satellite Communications, Tercera edición, 2020.