Propuesta de diseño para la codificación de canal de la televisión digital

Por ing. Luis Giraldo Raymond Rodríguez, Especialista Superior, Grupo de Proyectos de la Dirección Técnica, Radiocuba, e Ing. Addis Rey Domínguez, Investigadora, FAR Radiocuba
luis@radiocuba.cu; cid3@reduim.cu

Introducción

El estándar *DTMB—Digital Terrestrial Multimedia Broadcasting—* define la estructura de trama, codificación de canal y modulación para la radiodifusión de televisión digital terrestre en las bandas de VHF—*Very High Frequency*—y UHF —*Ultra High Frequency*—. Según la norma, se establece a 8 MHz de ancho de banda; no obstante, se ha vuelto a presentar contemplando la variante de 6 MHz. Durante 2009, se verificaron las posibilidades reales de cada una de las alternativas tecnológicas por las cuales puede optar nuestro país. Los mejores resultados técnicos han favorecido a la norma china DTMB y a la japonesa-brasileña ISDB-T —*Integrated Services Digital Broadcasting Terrestrial*—.

Entre las características del estándar DTMB se encuentran el uso de los códigos BCH —*Bose-Chadhui-Hocquenghen*— como codificador externo y LDPC —*Low Density Parity Check*— como decodificador interno para la codificación FEC —*Forward Error Correction*— con tres razones de código. Este bloque se encarga de agregar los bits de chequeo necesarios para proteger los datos de información de errores provocados por diversos factores durante la transmisión. El codificador externo BCH es común para las tres razones del codificador interno y es capaz de corregir un error simple o detectar uno doble, su objetivo es corregir errores esporádicos del codificador interno. Un codificador LDPC corresponde al FEC interno de la norma DTMB. Las tres razones de código que utiliza el FEC son 0.4, 0.6 y 0.8 y están dadas por la cantidad de grupos de bits codificados BCH que tienen que concatenarse para entrar al codificador LDPC.

El código BCH es una generalización del código de Hamming, el cual permite la corrección de errores múltiples. Es una clase poderosa de código cíclico que posee una larga selección de longitudes de bloques, razones de código, tamaño de alfabeto así como capacidad de corregir errores. En tanto, los códigos de chequeo de paridad de baja densidad son una clase de códigos de bloques lineales con decodificación alcanzable, los cuales logran un desempeño cercano al límite teórico máximo o límite de Shannon para bloques de transmisión extensos.

Este trabajo tiene como objetivo principal el estudio del estándar DTMB, específicamente, del bloque corrector de errores FEC con el propósito de desarrollar un módulo de propiedad intelectual (IP) sobre tecnología FPGA —*Field Programmable Gate Array*— como parte del proyecto de diseño de un modulador DTMB cubano.

Propuesta de diseño para la codificación de canal

El bloque de codificación FEC del estándar DTMB se encarga de agregar los bits de cheque se compone de un codificador externo BCH y un codificador interno LDPC como se mencionó anteriormente. El codificador externo es del tipo BCH (762,752) correspondiente a 752 bits de información y 10 de chequeo. Este se obtiene del código BCH (1023,1013), el cual se crea rellenando los bits de información con 261 ceros hasta llegar a 1013 bits (Figura 1). Una vez obtenida la palabra de código, se extraen los 261 ceros (formando palabras de código de 762 bits) ubicados en la cabecera de la misma, esto es posible pues el código es sistemático y coloca los bits de información por delante de los de chequeo.

Estos grupos de palabras codificadas BCH son concatenadas hasta formar palabras de cuatro, seis u ocho grupos BCH, en dependencia de la razón de código empleada: 0.4, 0.6 o 0.8. La longitud de estos grupos será de 3048, 4572 o 6096 bits y se le añaden los bits redundantes del codificador LDPC para formar palabras de longitud fija e igual a 7493 bits. Finalmente, se retiran los primeros cinco bits de chequeo de la palabra codificada. La extracción se justifica con el objetivo de transmitir en una trama señal (unidad básica de transmisión del estándar DTMB) un número fijo de tramas de transporte de 188 bytes.

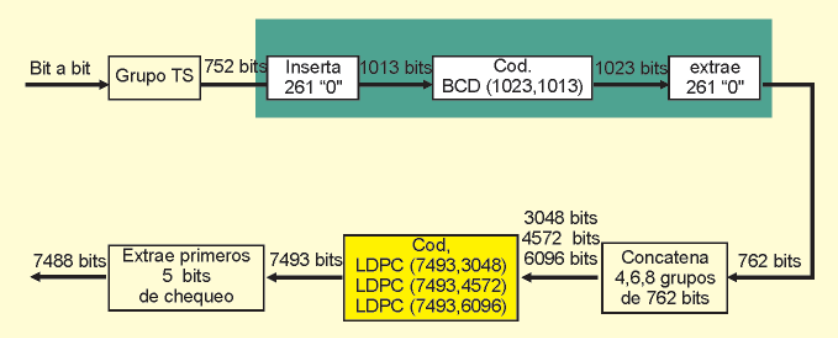

Figura 1 Codificación FEC de razones 0.4, 0.6 y 0.8 (Fuente: elaboración propia).

En las secciones siguientes se abordan los aspectos teóricos de ambos codificadores y se ofrecen soluciones para su implementación sobre la tecnología FPGA.

Codificador externo BCH

El código BCH se describe por los elementos enteros n, m y t, donde se deben cumplir las ecuaciones (1 -5) [2].

$$n = 2^{m-1} \quad (1)$$

$$t < 2^{n-1} \quad (2)$$

$$n - k = mt \quad (3)$$

$$m \geq 3 \quad (4)$$

$$dmín \leq 2t + 1 \quad (5)$$

Donde:

n: número de bits de cada palaba de código

k: número de bits de mensaje

t: cantidad de errores que puede corregir

m: cantidad de bits de chequeo mínima para corregir t errores

dmín: distancia mínima entre dos palabras de código

Una trama de transporte MPEG-2 contiene 188 Bytes o lo que es lo mismo 1504 bits. La codificación BCH se efectuará sobre 752 bits correspondientes a la mitad de dicha Trama de Transporte. En las ecuaciones 1-5 justifican la necesidad del relleno efectuado sobre los bits de información pues el código BCH (762,752) no cumple con estas características. El codificador externo BCH es común para las tres razones del codificador interno. La norma describe el polinomio generador que establece las palabras válidas del código según la ecuación 6. El mismo es una subclase de los códigos cíclicos y se puede implementar con registros de desplazamientos realimentados, lo cual disminuye sustancialmente la complejidad del hardware. Además, el cálculo del síndrome en el decodificador es igualmente realizable con esta estructura cuya arquitectura algebraica brinda técnicas eficientes de decodificación.

$$GBCH(x) = 1 + x^3 + x^{10} \quad (6)$$

Al igual que el polinomio generador, los mensajes a codificar se pueden representar mediante polinomios, y los bits de chequeo se pueden obtener como se muestra en la ecuación 7.

$$p(X) = X^{n-k}\, m(X) \text{ módulo } g(X) \quad (7)$$

Donde:

p(X): bits de paridad

m(X): bits de mensaje

g(X): polinomio generador

Como se observa de la ecuación 7, antes de realizar la división se debe multiplicar el mensaje por Xn-k lo cual equivale a crear el espacio para insertar los n-k bits de chequeo de forma sistemática. Por lo explicado anteriormente, se debe diseñar un circuito que efectúe la división de polinomios a base de registros de desplazamiento realimentados.

Diseño del codificador externo BCH

Para realizar el diseño del BCH se debe efectuar la división de polinomios a base de registros de desplazamiento realimentados.

Para formar la palabra de código de manera sistemática se requiere de un circuito conmutador (multiplexor) que alterne entre bits de mensaje y de chequeo. Para procesar los bits de mensaje no es necesario ningún circuito adicional, basta con presentarlos en la salida al mismo tiempo que son introducidos al circuito divisor para calcular los bits de chequeo, que constituyen el resto de la división. Una vez rotados en el circuito divisor y presentados a la salida del conmutador los 1013 bits de mensaje, el conmutador cambia y muestra los diez bits de chequeo que quedaron almacenados en las etapas del divisor (Figura 2).

Los siguientes pasos describen el proceso de codificación antes expuesto:

1. El conmutador 1 permanece cerrado durante las primeras 1013 rotaciones, permitiendo la transmisión de los bits de mensaje dentro del registro de desplazamiento.
2. El conmutador 2 permanece en su posición baja durante las primeras 1013 rotaciones permitiendo la salida de los bits de mensaje.
3. Después de las primeras 1013 rotaciones, el conmutador 1 se abre y el conmutador 2 pasa a la posición alta.

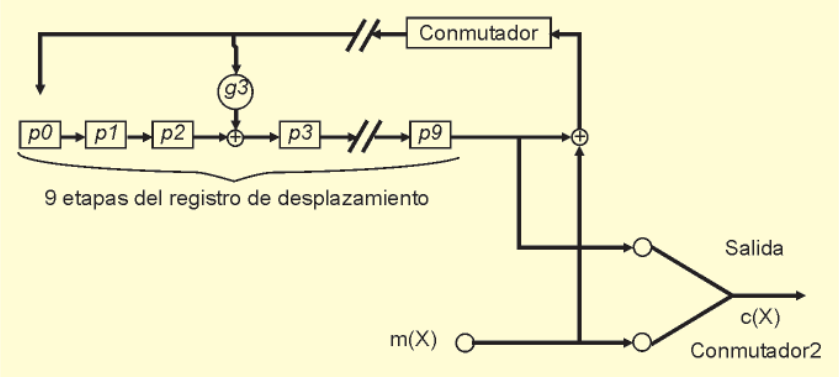

Figura 2 Diagrama del codificador BCH (1023,1013) (Fuente: elaboración propia).

4. Las restantes 10 rotaciones limpian al registro extrayendo los bits de chequeo por la salida.
5. El total de rotaciones es igual a 1023 y lo mostrado a la salida es exactamente la palabra de código.

La figura 3 muestra el esquema simplificado del bloque funcional BCH. Los datos de entrada son almacenados en un FIFO, a la frecuencia de CLK_MPEG (reloj de los datos de entrada encapsulados en tramas de transporte MPEG-2) que emite la señal de aviso SX_GRUPO al bloque Control cuando dispone de un grupo de 752 bits. Con la aparición de esta señal comienza la codificación que incorpora diez bits de chequeo y ocurre de forma serie, comenzando con los 261 ceros de relleno y luego los 752 bits almacenados en el FIFO.

El bloque de control es el encargado de conducir este proceso manejando la lectura del FIFO —*First In First Out*—, el multiplexor y el codificador. Este proceso se realiza con el reloj CLK_BCH (reloj usado para la codificación) que tiene el doble de la frecuencia de CLK_MPEG generando tiempos de espera entre los procesos de codificación. El bloque de control brinda la salida HAB_LDPC encargada de habilitar el funcionamiento del codificador LDPC, con esta señal se eliminan los 261 ceros, agregados anteriormente, del grupo codificado de 1023 bits (Figura 3). También provee la señal LECT_LDPC ya que lleva el conteo del número de grupos de 752 bits necesarios (5, 7, 9) para lograr el flujo continuo de los datos a la salida del sistema.

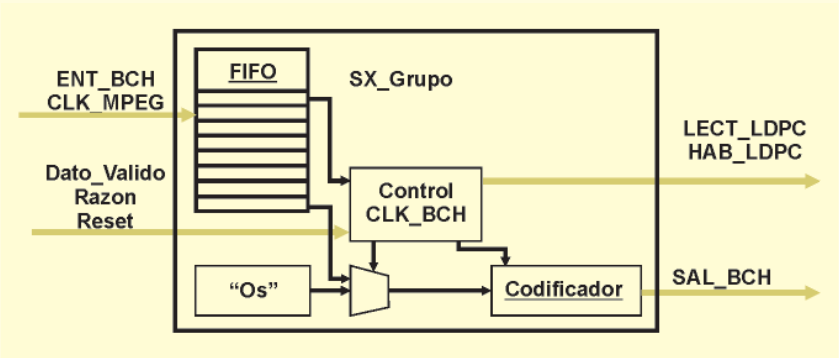

Figura 3 Esquema simplificado del codificador BCH (Fuente: elaboración propia).

Codificador Interno de la norma DTMB: LDPC cuasi-cíclico

EL codificador LDPC que nos ocupa es de baja densidad pues es un código de bloque lineal representado por una matriz de chequeo de paridad H (m x n) con baja densidad de 1s —se considera baja la densidad si es menor del 1% de todos los elementos de la matriz H—. En la figura 4 aparece la matriz H dada en la norma [1].

Esta matriz está conformada por submatrices Ai,j de (b x b) elementos con b = 127 y pesos de fila 1 cumpliendo con la condición de baja densidad, o sea, 1<< 127. La densidad de la matriz compuesta en su conjunto es del 0.26 % no igual a 1/127 debido a la existencia de submatrices todos ceros, con lo cual el valor es mucho menor que la máxima teórica del 1 %, siendo suficientemente baja como para permitir decodificaciones iterativas efectivas. Esto es en efecto, la clave innovativa detrás de la invención de los códigos LDPC.

$$H_{qc}=\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,t-1} \\ \vdots & \vdots & \ddots & \\ A_{C-1,0} & A_{C-1,1} & \cdots & A_{C-1,t-1} \end{bmatrix}$$

Figura 4 Matriz H de representación del código LDPC (Fuente: elaboración propia).

La matriz H está representada por un arreglo de matrices circulantes, en la cual cada fila es un desplazamiento cíclico de la anterior: la primera fila es un cambio cíclico de la última fila, y así mismo sucede con las columnas [2]. El peso de cada fila es 1, lo que significa que el peso de las columnas será igual, otorgando la propiedad de regulares a las submatrices y de irregular a la matriz compuesta.

Al conformar una estructura de código cuasi-cíclicos (QC) permite que se simplifiquen los diseños del codificador en comparación con los iniciales que eran aleatorios. Adicionalmente, posibilita mayor flexibilidad en el diseño del código, en especial la irregularidad, que logra una mejora en relación con los códigos LDPC cíclicos, ya que reducen la capacidad de almacenamiento necesaria. Esto se debe a que la matriz H de los códigos cíclicos deben ser del tipo n x n, sin importar la razón de codificación, y disminuyen la cantidad de 1s por filas y columnas que poseen los códigos LDPC cíclicos, minimizando la complejidad del codificador y decodificador [2].

La norma presenta la matriz generadora Gqc por lo que no es necesario obtenerla a partir de la matriz H. La misma es de forma cuasi-circular sistemática y se obtienen palabras de códigos de 7493 bits.

Diseño del codificador interno LDPC

Para el diseño del codificador interno LDPC, se utiliza el proceso de codificación de los códigos cuasi-cíclicos [3], donde se demuestra que si u representa los bits de mensajes divididos en secciones de 127 bits y Gqc sys representa la matriz generadora de forma sistemática del código LDPC, se puede obtener los bits de chequeo V del código (Ecuación 8).

$$V = u * G = u_0G_0, u_1G_1, \ldots, u_{c-1}G_{c-1} \qquad (8)$$

La ecuación destaca la necesidad de implementar un circuito que emule la multiplicación de matrices. La sección de chequeo de paridad pj puede ser formada con un Registro CSRAA —*Cyclic Shift-Register–Adder–Accumulator*— cuyo circuito se define en la figura 5.

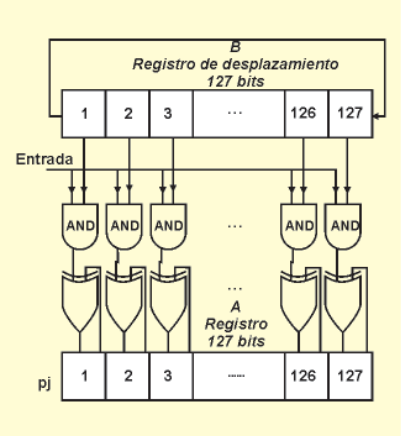

Figura 5 Circuito CSRAA (Fuente: elaboración propia).

La secuencia de información se divide en 24,36 y 48 secciones de 127 bits en dependencia de la razón 0.4, 0.6, 0.8 (4, 6, 8 grupos BCH) a lograr en el FEC, estas secciones coinciden con la cantidad de filas de la matriz generadora. Para obtener 7493 bits de salida, la cantidad de columnas que representan las secciones pj serán 35, 23, 11, respectivamente. A modo de explicación, se utilizará la de c=24 secciones de bits de entrada y 35 secciones de bits de chequeo. Las secciones u=(u1(0,) u1()1,)…u123) son rotadas en el codificador. Para $0 \leq c \leq 23$, la suma acumulada que constituye el aporte de la sección j de los bits de chequeo será según la ecuación 9.

$$S_{23,j} = (u_0G_{0,j} + u_1G_{1j} + \cdots + u_{23}G_{23,j}) \qquad (9)$$

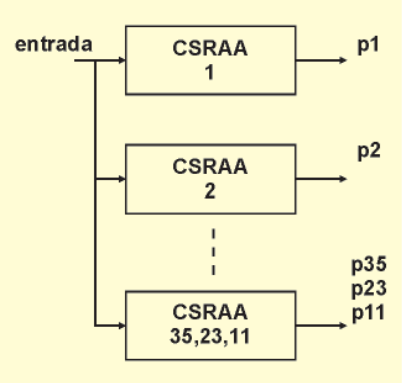

Figura 6 Diagrama en bloque del codificador LDPC (Fuente: elaboración propia).

Para formar 35 secciones de chequeo de paridad se necesitan 35 secciones de circuitos CSRAA (Figura 6).

Todas las secciones de chequeo de paridad se generan al mismo tiempo, en paralelo, y son rotadas en la salida de forma serie. El circuito de codificación consiste en 35, 23, 11 circuitos CSRAA. La codificación se lleva a cabo en un tiempo similar para todos los casos, o sea, para los distintos bits de chequeo de paridad (4445, 2921, 1397) según la razón, pues aunque se procesen más bits en un caso que en otro, la velocidad de entrada de los mismos aumenta a medida que crecen los grupos BCH [3].

La figura 7 expone de manera simplificada las partes componentes del codificador LDPC. La entrada RAZON indica al bloque de control la razón de código seleccionada y posibilita la configuración del Banco de ROM y los CSRAA. El Banco de ROM almacena las matrices generadoras de cada razón de código. La característica cuasi-cíclica de estas matrices reduce significativamente la capacidad de almacenamiento necesaria porque solo es preciso guardar la primera fila de cada submatriz cíclica.

Cuando la entrada HAB_LDPC (brindada por el codificador BCH) está activa, los datos de entrada son procesados por los CSRAA sincronizados a CLK_BCH, al mismo tiempo que son almacenados en un FIFO. Es necesario almacenarlos debido a que la estructura de codificación establece los bits de chequeo antecediendo a los bits de información. Se han diseñado 35 circuitos CSRAA con la razón 0.4 que requiere la mayor cantidad de secciones (35) de bits de chequeo. Este mismo hardware se emplea para codificar con la razón 0.6 y la razón 0.8 que requieren 23 y 11 secciones, respectivamente.

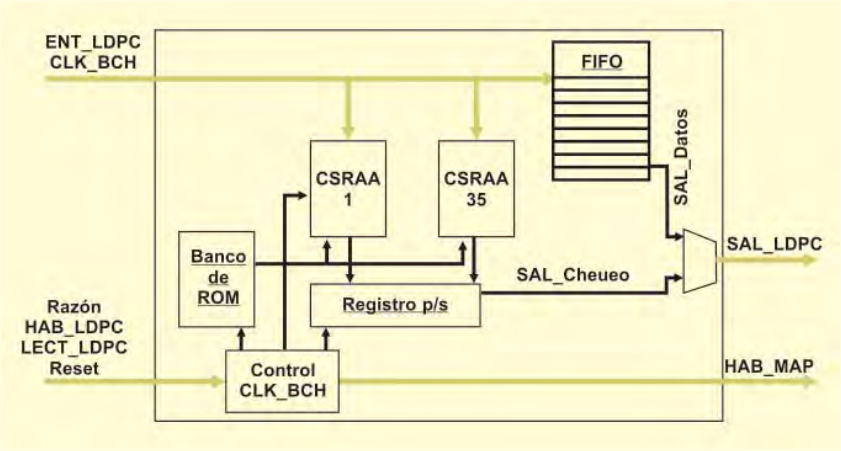

Figura 7 Esquema simplificado LDPC (Fuente: elaboración propia).

Cuando se obtiene el total de bits de chequeo, el control carga de forma paralelo los bits de chequeo en el Registro Paralelo/Serie. Cuando se activa la entrada LECT_LDPC, la palabra de código es transmitida de forma serie sincronizada a CLK_LDPC, el control maneja el multiplexor de salida para alternar los bits de SAL_CHEQUEO y SAL_DATOS, además activa la salida HAB_MAP para indicar que los datos de salida son válidos.

Verificación de los resultados

Después de finalizar el diseño y la programación correspondiente mediante el uso del lenguaje de descripción de hardware VHDL, se realizó la comprobación del código con la herramienta ISim soportada por Xilinx. ISim es un entorno de modelado y simulación que permite la verificación de los diseños generados con código VHDL, a partir de la cual se depuraron y corrigieron los errores cometidos durante el diseño. La veracidad de los resultados obtenidos se comprueba con la herramienta Simulink de Matlab. La figura 8 muestra la estructura del bloque del codificador externo BCH en Simulink. Los datos de entrada provenientes de Aleatorizador (bloque que le precede en la norma DTMB) son almacenados en el Buffer con el objetivo de formar el grupo de 752 bits. El bloque Pad incorpora los 261 ceros de relleno para crear el grupo de 1013 bits necesario para la codificación BCH (1023,1013). El bloque BCH Encoder realiza la codificación sistemática de forma paralelo y entrega a su salida la palabra de código de 1023 bits. Pad1 elimina los 261 ceros insertados anteriormente para generar el grupo de 762 bits, según exige la norma DTMB. La comprobación de los resultados de esta etapa se efectuó a través de To Workspace1 el cual crea en el Workspace de Matlab la variable COD_EXTERNO, donde se almacena el producto de la codificación.

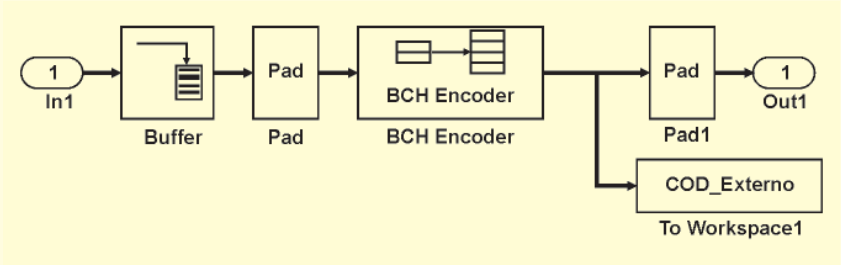

Figura 8 Estructura del codificador externo BCH (Fuente: elaboración propia).

En el campo Codeword length de la interfaz de configuración del bloque BCH Encoder, N especifica la longitud de la palabra codificada mientras que en el campo Message length, K indica la cantidad de bits de mensaje. Aunque este bloque brinda la opción de especificar el polinomio generador del código, también puede inferirlo insertando los parámetros N y K, que fue la opción escogida para el diseño (Figura 9).

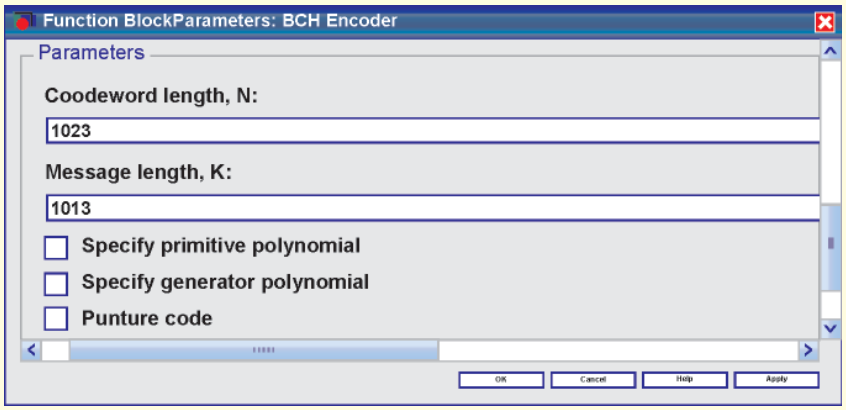

Figura 9 Interfaz de configuración del bloque BCH (Fuente: elaboración propia).

En la figura 10 se muestra un fragmento de las palabras de código BCH obtenidas de la simulación del diseño a través de la herramienta ISim de Xilinx y el resultado de la simulación de la estructura del codificador BCH realizada con los bloques funcionales ofrecidos por la herramienta Simulink de Matlab, donde ambas coinciden.

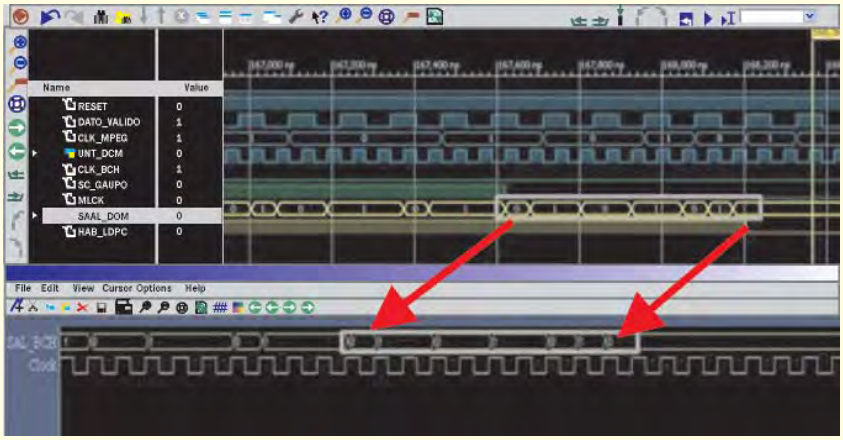

Figura 10 Comprobación de resultados (Fuente: elaboración propia).

Para la validación de los resultados del codificador interno LDPC se utiliza la estructura de diseño de la figura 11 en la herramienta Simulink de Matlab.

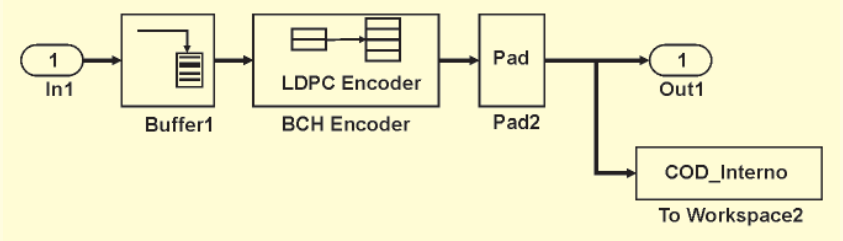

Figura 11 Estructura del codificador interno LDPC (Fuente: elaboración propia).

El bloque Buffer1 almacena los grupos de 762 bits provenientes del codificador externo para formar grupos de 3048, 4572 o 6096 bits de acuerdo con la razón empleada. El bloque LDPC Encoder codifica de forma paralelo los bits de entrada (Figura 12).

En el campo Parity-check matrix se especifica la matriz de chequeo de paridad del código LDPC la cual varía según la razón de código. Se creó una función en Matlab para la obtención de las matrices de chequeo de paridad [3]. Dicha función retorna la matriz de chequeo de paridad H de forma esparcida según requiere el bloque LDPC Encoder.

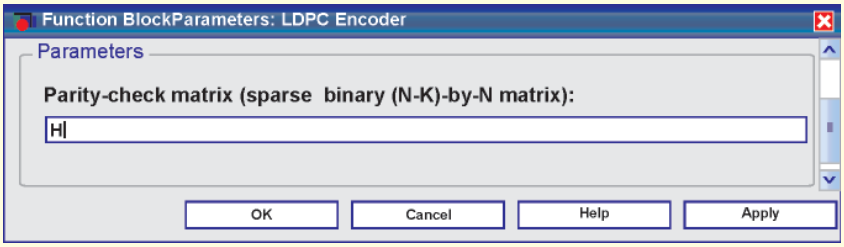

Figura 12 Interfaz de configuración del bloque LDPC (Fuente: elaboración propia).

El bloque Pad2 extrae los primeros cinco bits de chequeo. El codificador interno se verificó con el bloque To Workspace2, mediante el cual se crea la variable COD_INTERNO donde se almacena el resultado. En la figura 12 se expone la comprobación de los resultados en ambas herramientas de diseño.

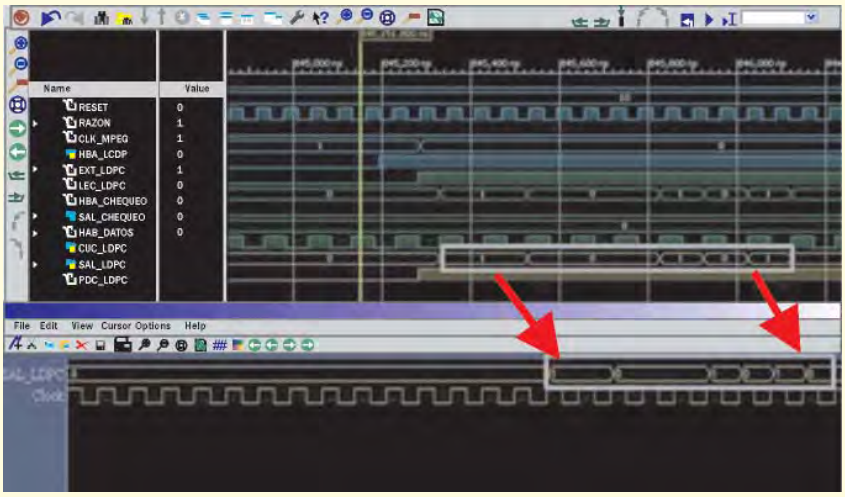

Figura 13 Comprobación de resultados (Fuente: elaboración propia).

Conclusiones

El estudio realizado sobre el estándar DTMB permitió adquirir los conocimientos necesarios para luego diseñar la etapa correspondiente a la codificación de canal en VHDL. El mismo contribuye al enriquecimiento científico con el conocimiento del bloque FEC del estándar de Televisión Digital Terrestre DTMB. La propuesta de diseño de los codificadores externo e interno facilita la realización de un módulo de propiedad intelectual de la norma china con el fin de lograr la soberanía tecnológica.

Referencias bibliográficas

[1] Standard, C.N., Framing Structure, Channel Coding and Modulation For Digital Television Terrestrial Broadcasting System 2006.

[2] Ryan, W.E. and S. Lin, Channel Codes: Classical and Modern. 2009: Cambridge

[3] Domínguez, A.R. and Rodríguez, L.R., Diseño del Aleatorizador, la Codificación de Canal, Mapeo y Entrelazado correspondientes a un modulador DTMB., in Facultad de Ingeniería Eléctrica. 2011, CUJAE: La Habana.