



Desempeño de un Sistema de Comunicación Digital Empleando Codificación de Línea

Digital Communication System Performance Using Line Codes

Oscar Velásquez

Facultad de Ingeniería Electrónica y
Telecomunicaciones
Universidad del Cauca
oveland@unicauca.edu.co

Javier Urresty

Facultad de Ingeniería Electrónica y
Telecomunicaciones
Universidad del Cauca
jaus10@unicauca.edu.co

Victor Miramá

Facultad de Ingeniería Electrónica y
Telecomunicaciones
Universidad del Cauca
vmirama@unicauca.edu.co

Claudia Hernández

Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
claudiah@unicauca.edu.co

Pablo Jojoa

Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
pjojoa@unicauca.edu.co

(Tipo de Artículo: Investigación Científica y Tecnológica. Recibido el 26/11/2017. Aprobado el 06/02/2018)

Resumen. En el área de las telecomunicaciones es importante analizar el impacto del tipo de codificación de canal sobre el sistema de comunicación digital, estos sistemas emplean técnicas de procesamiento digital de señales que incorporan elementos hardware para crear sistemas de comunicación fiables y de alto rendimiento. Con ayuda de los criterios y la teoría de las comunicaciones, los dispositivos hardware ayudan a estudiar y analizar el comportamiento físico de las señales. Existe una gran variedad de herramientas tanto software como hardware a nivel académico, que permiten validar los conceptos de un sistema de comunicación digital y su funcionamiento, en este artículo se expone el análisis de desempeño de la implementación sobre hardware reconfigurable de diferentes esquemas básicos de codificación de línea, mediante el uso de las herramientas Simulink, System Generator; las curvas de desempeño a nivel de BER y la Eficiencia Espectral. Los esquemas de codificación presentaron una BER similar al comportamiento teórico, y los códigos de línea Polares, Unipolares, Bipolares y Diferenciales, tienen igual desempeño en todas las pruebas, sin embargo, la eficiencia espectral varía para cada código analizado de forma independiente, el uso de la energía en cada tipo de codificación también varía de acuerdo con su complejidad.

Abstract. On the telecommunications field, it is important to analyze the impact from the channel encoding type on a digital communication system. These systems use digital signal processing techniques, which involve hardware elements to create reliable and high performance communication systems. With the Support of criteria and the telecommunication theory, hardware devices help to study and analyze the physical behavior of signals. There is a wide variety of tools, both, software and hardware at an academic level that allows to validate the digital communication system concepts and its performance. In this article the performance analysis upon reconfigurable hardware implementation from the different basic line encoding schemes, deployed in tools such as simulink and system Generator, both BER and spectral efficiency performance curves, are introduced. The encoding schemes showed a close BER to the theoretical behavior, and the polar, unipolar, bipolar and differential line codes have the same performance in all the tests, however the specific efficiency changes for each code analyzed independently. The use of energy in each type also varies due its complexity.

Palabras clave. Codificación de Línea; comunicación digital; desempeño; Hardware reconfigurable; FPGA.

Keywords. Digital communication; line codes; performance; reconfigurable hardware; FPGA

DOI 10.21500/20275846.3323

1. Introducción

En la actualidad existe una gran variedad de herramientas tanto software como hardware que permiten, para entornos académicos, validar los conceptos de un sistema de comunicación digital y la teoría detrás de su funcionamiento. Generalmente el desarrollo e implementación de este tipo de sistemas tiene un elevado costo, por lo que existe la necesidad de encontrar tecnologías y métodos que permitan

implementarlos de forma flexible y reconfigurable para aprovechar los recursos disponibles y así obtener una disminución significativa en los costos de adquisición de herramientas útiles en el aula de clase. De esta manera existen dispositivos hardware reconfigurables, como el Arreglo de Compuertas Programable en Campo (FPGA, *Field Programmable Gate Array*) que por su capacidad, flexibilidad y economía son muy utilizados en diversas

aplicaciones que involucran el procesamiento digital de señales [1].

En este artículo, se presenta el desempeño de un sistema de comunicación digital que emplea códigos de línea, el cual fue implementado sobre la herramienta *System Generator* y la tarjeta de desarrollo Spartan 3A de Xilinx basada en FPGA.

Inicialmente se realizó la definición teórica para los códigos de línea UNRZ, NRZI, URZ, PNRZ, PRZ, AMI-NRZ, AMI-RZ Manchester y Manchester Diferencial, seguido por la presentación del diseño del sistema de comunicación digital y los módulos de codificación y decodificación de los códigos de línea. Posteriormente se presenta la implementación sobre FPGA del sistema digital, compuesto por transmisor, canal y receptor. Finalmente se presentan los resultados del desempeño en las conclusiones.

2. Codificación de línea

La codificación de línea es una técnica utilizada para la transmisión digital en banda base mediante la conversión de datos digitales en señales digitales que se acoplan al canal por donde son transmitidas, el proceso suele conocerse también como formato de señalización y es conocido por sus aplicaciones iniciales sobre sistemas de telefonía alambrados o líneas telefónicas, de ahí el nombre código de línea [2], sin embargo, en los últimos años su implementación va desde transmisión de voz hasta protocolos de Redes de Área Local (LAN, *Local Area Network*) como Ethernet y redes de fibra óptica [3].

Situando los códigos de línea en el contexto de un sistema de comunicaciones digital como el de la Figura 1, se tiene como propósito transmitir información desde un emisor a un receptor a través de un enlace o canal.

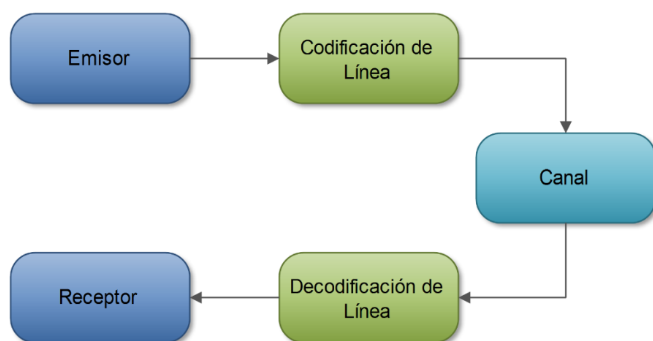


Figura 1. Diagrama de bloques sistema de comunicación digital con codificación de línea

Un código de línea es, por tanto, un tipo de codificación usado en sistemas de comunicación para la transmisión digital en banda base y está conformado por pulsos rectangulares que siguen un determinado patrón de codificación, el cual asigna niveles de tensión a los bits que entrega la fuente [4].

En el presente trabajo se implementaron los códigos de línea mostrados en la Tabla 1, clasificados según su polaridad.

Tabla 1. Códigos de línea considerados

Códigos Unipolares	UNRZ (<i>Unipolar Non Return to Zero. No Retorno a Cero Unipolar</i>). URZ (<i>Unipolar Return to Zero. Retorno a Cero Unipolar</i>).
Códigos Polares	PNRZ (<i>Polar Non Return to Zero. No Retorno a Cero Polar</i>). PRZ (<i>Polar Return to Zero. Retorno a Cero Polar</i>). NRZI (<i>Non Return to Zero Inverted. No Retorno a Zero Invertido</i>). Manchester. En sus variantes Manchester o Bifase L y Manchester Diferencial.
Códigos Bipolares	AMI (<i>Alternate Mark Inversion, Inversión de Marcas Alternadas</i>). En sus variantes AMI-RZ y AMI-NRZ.

La Figura 2 se muestra la densidad espectral de potencia que posee cada codificación en el dominio de la frecuencia.

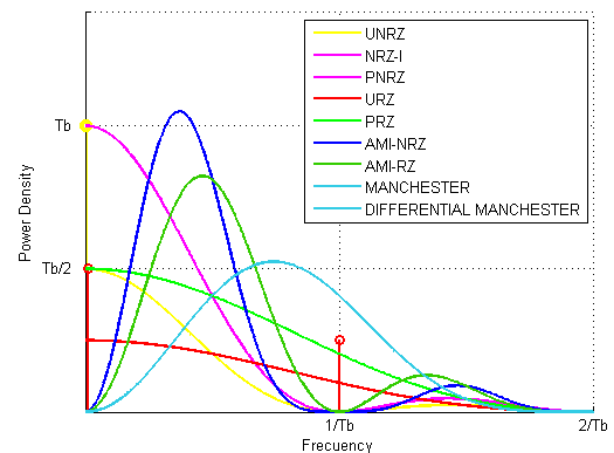


Figura 2. Densidad espectral de potencia para cada código de línea implementado

A partir de la Figura 2, la Tabla 2 muestra la eficiencia espectral de cada código de línea, definida como la relación entre la tasa de transmisión de los datos y el ancho de banda ocupado por la señal transmitida, tal como lo muestra la ecuación 1.

$$\eta = \frac{R_b \text{ bps}}{B \text{ Hz}} \quad (1)$$

La eficiencia espectral indica el número de bits por segundo que se pueden transmitir por cada hercio ocupado en el canal de transmisión [5].

Tabla 1. Eficiencia espectral de los códigos de línea

Código de Línea	Eficiencia Espectral (bps/Hz)
UNRZ	1
NRZ-I	1
PNRZ	1
URZ	0.5
PRZ	0.5
AMI-NRZ	1
AMI-RZ	1
MANCHESTER	0.5
DIFF. MANCHESTER	0.5

3. Diseño del sistema de comunicación con codificación de línea

Para el desarrollo del sistema de comunicaciones con codificación de línea, se elaboró el diagrama de bloques mostrado en la Figura 3, en donde puede apreciarse un Emisor o fuente que genera la información a transmitir; un codificador de línea; un canal de comunicaciones que adiciona ruido AWGN; un decodificador de línea y finalmente un receptor donde se encuentra el componente encargado de evaluar la Tasa de Error de Bit (Bit Error Rate).

A partir del diagrama de bloques de la Figura 3, se diseña en Simulink un modelo de referencia para la implementación del sistema en la FPGA, es importante resaltar que sobre la FPGA se implementó el transmisor,

el canal y el receptor. La Figura 4 presenta el modelo para el sistema de comunicaciones con código de línea UNRZ.

Teniendo en cuenta el modelo de referencia de la Figura 4 y utilizando la herramienta *System Generator* de Xilinx que permite la generación automática de código de Lenguaje de Descripción Hardware VHSIC (VHDL, *VHSIC Hardware Description Language*) para programar la FPGA y que posee elementos de diseño predefinidos, se diseñan uno a uno los bloques que componen el sistema de comunicación digital con codificación de línea.

La Figura 5 muestra el sistema de comunicación digital diseñado y que será implementado en la FPGA, en el cual se integran los bloques codificadores y decodificadores de cada código línea. Estos bloques implementan algoritmos de programación, los cuales describen el proceso de codificación y decodificación, a través de archivos con extensión .m, que siguen un algoritmo como el presentado en la Figura 6 para la codificación UNRZ, los cuales son incluidos en los bloques M-Code de *System Generator*. Además, el sistema contiene bloques de control y comunicación para transmitir los datos obtenidos de la medición de la Tasa de Error de bit al hacer la implementación en hardware.

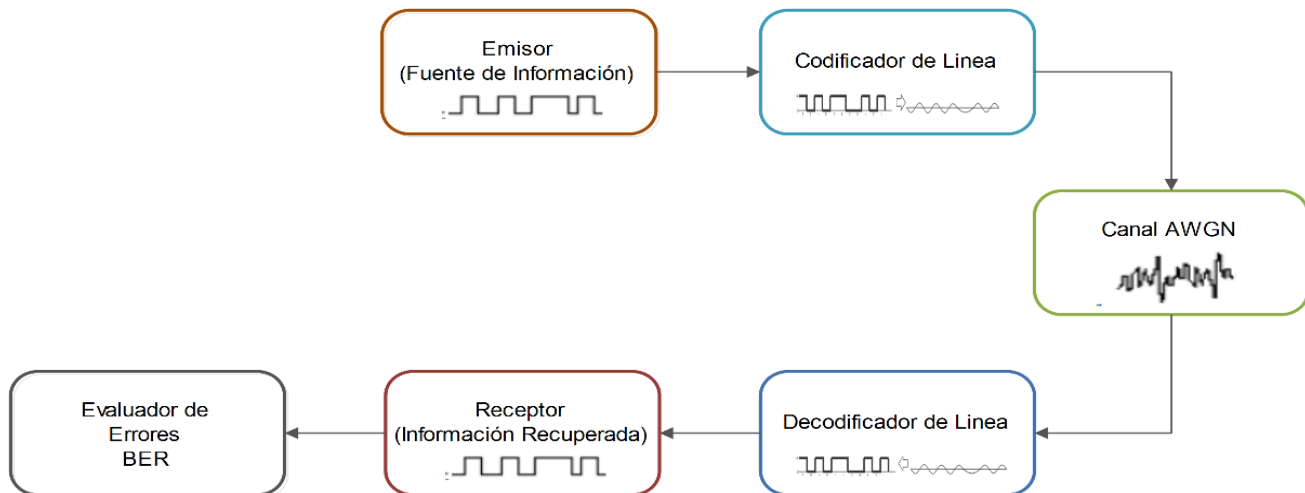


Figura 3. Diagrama de bloques del sistema de comunicaciones

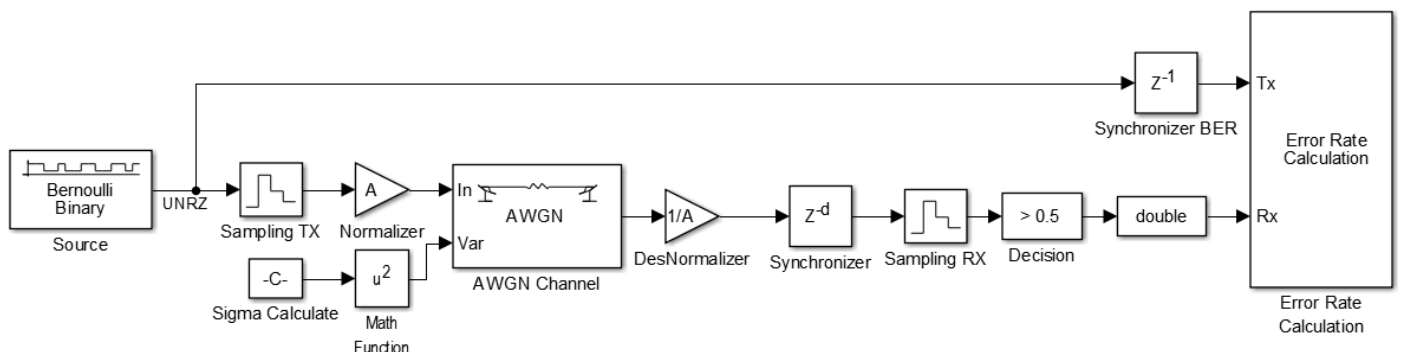


Figura 4. Modelo de referencia del sistema de comunicación con código de línea UNRZ en Simulink

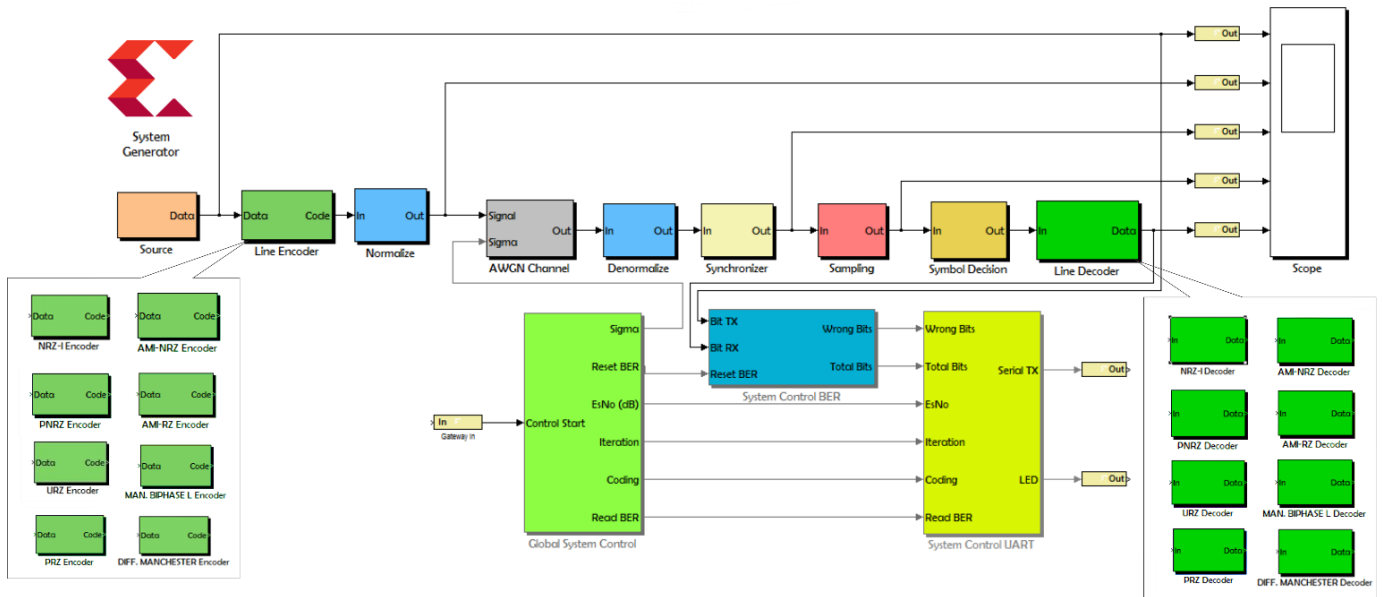


Figura 5. Sistema de comunicación digital con codificación de línea diseñado con System Generator

Algoritmo 1 Pseudocódigo codificación UNRZ

Entrada: E=Muestreo de la señal digital de entrada cada T_b segundos

Salida: S= Señal codificada.

- 1: inicio
- 2: Inicializar variables de entrada y salida E y S respectivamente.
- 3: Si E=0 entonces
S=1
- 4: Si no
S=0
- 5: fin si
- 6: fin

Figura 6. Pseudocódigo para el codificador UNRZ

El emisor del sistema está conformado por un bloque llamado *Bits Generator* mostrado en la Figura 7, el cual implementa un Registro de Desplazamiento con Realimentación Lineal (LFSR, *Linear Feedback Shift Register*), compuesto por una cadena de biestables que puede generar largas secuencias pseudoaleatorias de unos y ceros.



Figura 7. Captura de la tasa de error de bit en la aplicación de escritorio

En la Figura 8 se muestra el bloque Generador de Ruido Blanco Gaussiano (WGNG, *White Gaussian Noise Generator*) propio del paquete de Xilinx,

encargado de generar ruido AWGN usando una combinación del algoritmo de Box-Muller y el teorema del límite central [6].

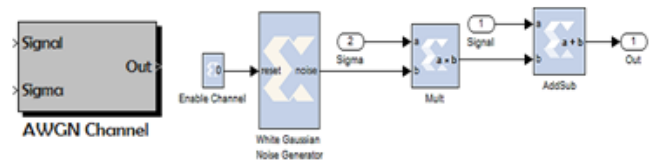


Figura 8. Captura de la tasa de error de bit en la aplicación de escritorio

El algoritmo de Box-Muller genera una variable aleatoria normal estándar de la forma $X \sim N(\mu_x, \sigma_x^2)$ utilizando una transformación de dos variables aleatorias independientes entre sí, que están uniformemente distribuidas entre 0 y 1 [6].

La salida de cuatro subsistemas paralelos Box-Muller son promediados para obtener una Función Densidad de Probabilidad (PDF, *Probability Density Function*) gaussiana, generado así ruido normalizado. El ruido generado por un WGNG es modelado a través de una variable aleatoria normal estándar $N(0,1)$, de esta manera al multiplicar un factor constante y real por la señal de salida de un WGNG, este factor toma el lugar de la desviación estándar (σ). Finalmente, es así como el ruido que se adiciona a la señal codificada se modelada como una variable aleatoria normal estándar de media cero ($\mu_x = 0$) y varianza σ_x^2 definida por la ecuación 2 [7].

$$\sigma_x^2 = \frac{N_0}{2} \quad (2)$$

De esta manera, la señal de entrada *Sigma* se utiliza para controlar la potencia de ruido generada por el bloque de la Figura 8. Esto con el fin de evaluar el desempeño de cada codificación bajo un determinado

valor de la relación energía de símbolo a densidad espectral de potencia de ruido (E_s/N_0).

El bloque de control de Errores de la Figura 9, registra la cantidad de bits errados a partir de la comparación de las entradas *Bit TX* y *Bit RX* correspondientes a la señal digital transmitida por el bloque *Source* y la señal a la salida del decodificador de línea, respectivamente. Cuenta con una entrada adicional (*Reset BER*) que reestablece a cero los contadores internos que trasfieren a su salida los bits errados (*Wrong Bits*) sobre un total de bits transmitidos (*Total Bits*).

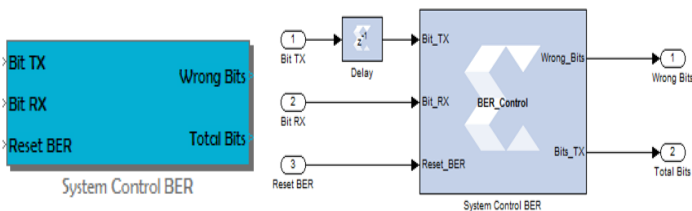


Figura 9. Bloque que registra la Tasa de Error de Bit

El sistema de control para la transmisión serial de la información procesada por la FPGA hacia el computador es mostrado en la Figura 10. El principal objetivo de este bloque es transmitir los datos obtenidos por el Sistema de Control de Errores y el Sistema de Control Global, mediante el modulo Transmisor-Receptor Asíncrono Universal (UART, *Universal Asynchronous Receiver-Transmitter*) integrado en la placa de desarrollo, hacia la aplicación de escritorio desarrollada en Microsoft Visual Studio la cual procesa la información recibida con el fin de obtener la lectura de la tasa de error de Bit del sistema de comunicación.

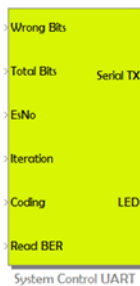


Figura 10. Bloque Control UART

El subsistema visualizado en la Figura 10 consta de dos salidas, siendo Serial TX el puerto de salida del FPGA que se conecta al pin transmisor del puerto serial RS-232 de la placa de desarrollo *Spartan 3A*. Por su parte, la salida LED es una señal binaria y está conectada a un led de la placa que indica visualmente la transmisión serial de los datos.

El funcionamiento de este bloque consiste en generar una trama, mostrada en la Figura 11, la cual brinda información de la tasa de error de bit evaluada bajo cierto valor de E_s/N_0 . Cabe resaltar que la trama generada tiene el formato del Código Estándar Estadounidense para el Intercambio de Información (ASCII, *American Standard Code for Information*

Interchange) [8], así la tarea de decodificar los datos recibidos en la aplicación de escritorio solo se limita a interpretar cadenas de caracteres que facilitan el cálculo de la BER.



Figura 11. Estructura de la trama de transmisión serial

El módulo de control principal mostrado en la Figura 12, genera los valores de desviación estándar (σ) para controlar la potencia de ruido en el canal AWGN del sistema de comunicaciones. Además controla la lectura de la tasa de error de bit para cierto valor de E_s/N_0 .

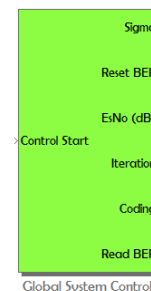


Figura 12. Sistema de Control Principal

La entrada y las salidas del bloque *Global System Control* son:

- *Sigma*: Es una salida de tipo decimal que representa el valor de desviación estándar y controla la potencia de ruido generada por el canal AWGN para determinado valor de E_s/N_0 .
- *Reset BER*: Es una salida de tipo binaria que indica el restablecimiento de la lectura de la tasa de error de bit al módulo *System Control BER* ilustrado en la Figura 8.
- *EsNo (dB)*: Es una salida numérica que indica el valor de E_s/N_0 en un determinado instante de tiempo.
- *Iteration*: Es una salida numérica que indica el número de repeticiones en un determinado instante de tiempo para un valor fijo de E_s/N_0 .
- *Coding*: Es una salida numérica que indica el tipo de codificación con los valores a corresponder:

- 1 → Código UNRZ.
- 2 → Código NRZI.
- 3 → Código PNRZ.
- 4 → Código URZ.
- 5 → Código PRZ.
- 6 → Código AMI-NRZ.
- 7 → Código AMI-RZ.
- 8 → Código MANCHESTER OR BIPHASE-L.
- 9 → Código DIFFERENTIAL-MANCHESTER.

- *Read BER*: Es una señal binaria que indica mediante un estado lógico “1” la transmisión de sus entradas Wrong bits, Total Bits, Iteration, EsNo y Coding, por el puerto serial.
- *Control Start*: Es un puerto de entrada de tipo lógico que controla el inicio de la configuración de los parámetros del sistema de comunicaciones. Cuando su estado toma un “0” lógico el módulo *Global System Control* se deshabilita y se reinician todos los registros de los demás bloques de Control. Esta entrada está conectada a un interruptor de la tarjeta de desarrollo Spartan 3A, de tal forma que el usuario pueda tener el control sobre el inicio de la lectura de la BER.

Antes de implementar el sistema de comunicación en la FPGA, se deben configurar algunos parámetros del bloque *Global System Control*.

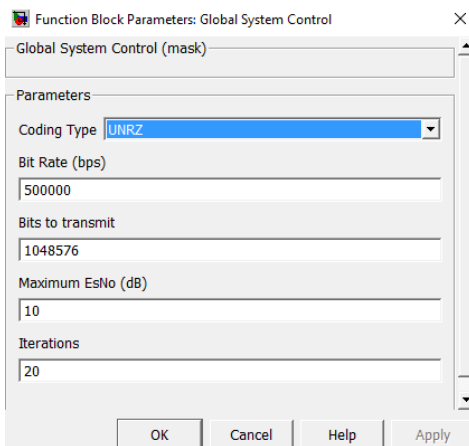


Figura 13. Configuraciones del bloque *Global System Control*

Los parámetros de configuración del bloque mostrado en la Figura 13 son:

- *Coding Type*: Permite seleccionar el tipo de codificación que implementa el sistema de comunicación digital, de tal forma que el sistema pueda normalizar automáticamente la potencia de la señal correspondiente al código de línea escogido.
- *Bit Rate*: Establece la tasa de transmisión de los bits, considerado el valor en unidades de bits por segundo.
- *Bits to Transmit*: Permite establecer el número de bits a transmitir por cada iteración, de tal forma que cuando esta cantidad de bits sean transmitidos la salida *Read BER* del bloque de control toma el estado lógico “1”, iniciando así, el proceso de transmisión por el puerto serial de los valores de BER calculados.
- *Maximum EsNo (dB)*: Permite establecer el rango de valores para E_s/N_0 desde 0 hasta el valor indicado. De esta manera el bloque calcula los valores de Sigma incluidos en dicho rango.
- *Iterations*: Establece el número de veces que el sistema transmite la cantidad de bits configurada, para cada valor de E_s/N_0 . De esta manera se pueden

obtener varias medidas de la tasa de error de bit para un mismo escenario, aumentando la confiabilidad en la evaluación del rendimiento de cada código de línea en el análisis de resultados.

4. Implementación del sistema

Para la implementación física del sistema de comunicación, es necesario generar a partir del modelo en *System Generator*, el archivo de programación *bitstream* (.bit), el cual contiene las instrucciones de cómo deben conectarse los componentes internos de la tarjeta de desarrollo *Spartan 3A* para ejecutar las tareas requeridas.

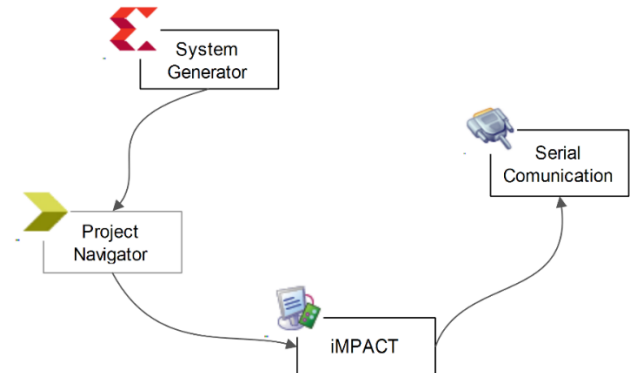


Figura 14. Proceso de implementación Hardware

El proceso de la implementación hardware ilustrado en la Figura 14, inicia con la configuración de *System Generator*, mediante su panel de opciones proporcionado por el entorno de Simulink, en el cual se establecen los parámetros principales como el modo de compilación y la referencia del dispositivo FPGA a programar, con el fin de obtener un archivo .xise, que a su vez es leído por el software *Project Navigator*, que se encarga de analizar todas las restricciones de diseño, calcular los recursos hardware requeridos para la implementación y finalmente realizar la generación del archivo de instalación con extensión .bit y las limitaciones de tiempo (*timing constraints*), los cuales determinan el retardo máximo que el sistema puede soportar. Luego se procede a cargar el código sobre el FPGA seleccionada (XC3S700A), por medio de la herramienta *iMPACT*, que se encarga de realizar la configuración directa del FPGA grabando el código de programación en la memoria flash.

Por último, se realiza la captura de los datos generados y procesados por la placa de desarrollo a través del puerto serial que envía la información de la BER del sistema de comunicación con codificación de línea a la aplicación de escritorio mostrada en la Figura 15, la cual se desarrolló en Visual Basic.

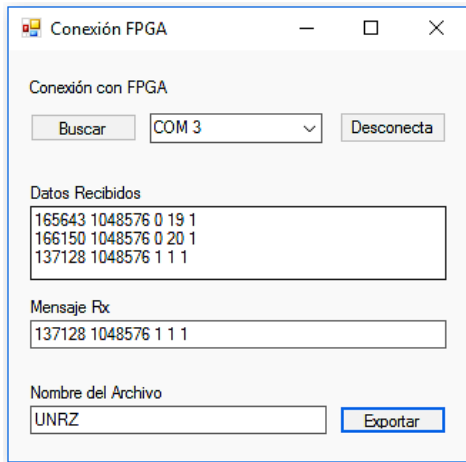


Figura 15. Captura de la tasa de error de bit en la aplicación de escritorio

Los datos recibidos en la aplicación de escritorio se almacenan en una memoria temporal de la aplicación hasta que el FPGA finalice la transmisión. Luego estos son exportados por el usuario a un archivo Excel, para posteriormente leer su contenido mediante scripts desarrollados en Matlab, los cuales extraen los datos obtenidos y los presentan en forma gráfica para su análisis posterior.

5. Resultados

Se comparan los resultados obtenidos al implementar el sistema diseñado sobre FPGA para los escenarios planteados en la Tabla 3. Para efectos de este documento se agregan los resultados más relevantes.

Luego de observar los valores de la BER obtenidos en cada sistema se decidió documentar un total de 20 iteraciones por cada valor de E_s/N_0 , tomando como dato de tabulación, para las gráficas expuestas, la media de los valores medidos, ya que las variaciones en cada una de las pruebas fueron mínimas y suficientes para la obtención de resultados que presentaran una tendencia estadística confiable acorde a los valores teóricos.

Tabla 3. Escenarios de prueba

Escenario	Parámetros de configuración			
	Cantidad de bits transmitidos	bits	Velocidad de transmisión	de
	1536	1048576	10 Kbps	500 Kbps
1	✓		✓	
	✓			✓
2		✓	✓	
		✓		✓

En la Figura 16 se valida el diseño e implementación de la codificación UNRZ con una velocidad de transmisión de 500Kbps y 1048576 bits transmitidos, obtenidas en Simulink, *System Generator* y el FPGA, las cuales se comparan con la curva teórica aproximada. Se

observa que las tres curvas, presentan un comportamiento similar a la curva teórica, esto permite comprobar que el sistema implementado sobre el FPGA es correcto. Para los demás escenarios los resultados fueron consistentes y se observó un comportamiento similar entre las curvas teóricas y las obtenidas con la implementación.

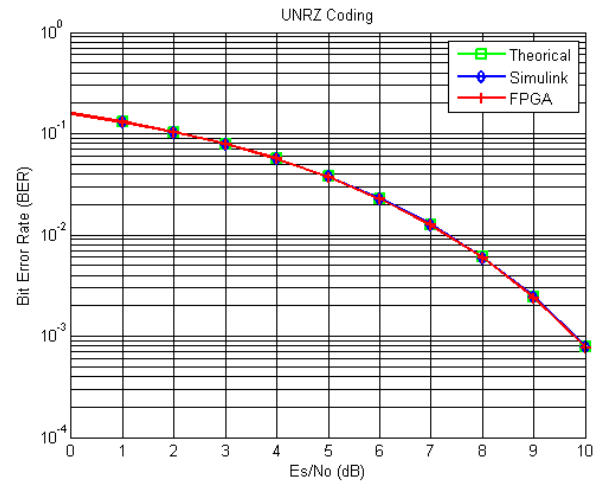


Figura 16. Curvas de desempeño de la BER del código

La Figura 16, evidencia que la implementación de la codificación y del sistema en entorno controlado como el de FPGA, proporciona resultados muy próximos a la teoría, pero sería necesario realizar las pruebas sobre un canal real.

De igual forma la Figura 17, muestra el desempeño de todas las codificaciones consideradas en la Tabla 1, las cuales se implementaron en el sistema de comunicación digital a una velocidad de transmisión de 500 Kbps sobre un total de 1048576 bits transmitidos.

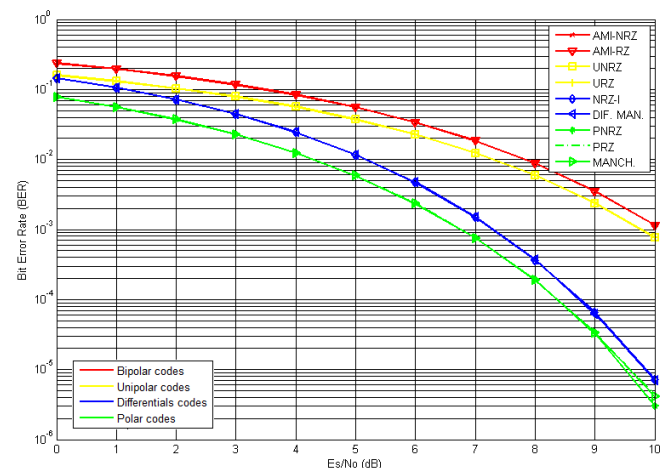


Figura 17. Curvas de desempeño para los códigos de línea implementados

En la Figura 17, se puede apreciar la adecuada implementación de los códigos de línea, donde el desempeño más bajo se obtiene con los códigos AMI-NRZ y AMI-RZ debido al proceso de rectificación de onda, propio de la etapa de decodificación en el

receptor, lo que conlleva a que los niveles de la señal recibida (señal transmitida + señal de ruido) se conviertan en niveles positivos de tensión presentes a la entrada del decisor de símbolos, quien finalmente encuentra mayor probabilidad de decodificar erróneamente la señal muestreada. El mejor desempeño se tiene al implementar los códigos polares, precisamente por su bajo consumo de energía en comparación con los demás códigos implementados.

En la Tabla 4 se resume la eficiencia espectral de cada una de las técnicas de modulación, mostrando independencia con el desempeño de los diferentes esquemas de codificación, evidenciando una contradicción entre la baja eficiencia espectral que tienen los códigos de mejor desempeño, como se muestra en la Figura 8.

6. Conclusiones

La implementación sobre hardware reconfigurable (FPGA), permitió establecer apropiadamente un análisis completo del desempeño del sistema de comunicación digital con codificación de línea.

Los códigos de línea Polares, Unipolares, Bipolares y Diferenciales, tienen igual desempeño en cada clasificación, sin embargo, la eficiencia espectral varía para cada código de línea analizado de forma independiente.

Tabla 2. Eficiencia espectral códigos de línea

Código de Línea	Ancho de banda de nulo a nulo (Hz)	Eficiencia Espectral (bps/Hz)
UNRZ	$1/T_b$	1
NRZ-I	$1/T_b$	1
PNRZ	$1/T_b$	1
URZ	$2/T_b$	0.5
PRZ	$2/T_b$	0.5
AMI-NRZ	$1/T_b$	1
AMI-RZ	$1/T_b$	1
MANCHESTER	$2/T_b$	0.5
DIFF. MANCHESTER	$2/T_b$	0.5

Los sistemas de comunicación digital que implementan los códigos de línea PNRZ, PRZ y

MANCHESTER requieren la mitad de energía que el código UNRZ a fin de lograr una BER objetivo específica. Además, presentan el mejor desempeño con respecto a las demás codificaciones, aunque su eficiencia espectral es menor que la de los códigos sin retorno a cero.

Los sistemas de comunicación digital que implementan los códigos de línea NRZ-I y *Differential Manchester*, tienen un desempeño similar a los códigos UNRZ para un nivel de E_s/N_0 igual a 0 dB; sin embargo, para valores altos de E_s/N_0 tienen un desempeño similar a los códigos polares.

7. Trabajos futuros

Con el desarrollo de este trabajo se visualiza la opción de desarrollar otros trabajos que implementen técnicas de codificación multinivel. Por otro lado, también puede ser interesante el desarrollo empleando un canal real, para lo cual se tiene que implementar la parte de radio frecuencia en el sistema transmisor y receptor.

8. Agradecimientos

Los autores agradecen a la Universidad del Cauca, por su apoyo y soporte en el desarrollo de los proyectos de investigación.

Referencias

- [1] National Instrument, "Introducción a la Tecnología FPGA: Los Cinco Beneficios Principales," Colombia, 2011. [En línea]. Disponible: <http://www.ni.com/white-paper/6984/es/>.
- [2] W. Stallings, *Comunicaciones y Redes de Computadores*. 6th Edición, Prentice Hall, 1999.
- [3] E. Coimbra "Transmisión digital en banda base," 2010. [En línea]. Disponible: http://coimbraweb.com/documentos/digital/4.3_transmision_digital_bbase.pdf
- [4] J. G. Proakis, *Digital Communications*. 4th Edition, McGraw-Hill Companies, 2000.
- [5] G. A. Smillie, *Analogue and Digital Communication Techniques*. Newnes, 2nd ed, 2002. [E-book]. Disponible: ScienceDirect.
- [6] O. M. Ulgen, J. J. Black, B. Johnsonbaugh y R. Klungle, "Simulation Methodology -A Practitioners Perspective," Michigan Simulation User Group, Technical Committee on Simulation Methodology, 2000.
- [7] A. Artés, F. Pérez, "Comunicaciones digitales," Departamento de Teoría de la Señal y Comunicaciones, Madrid 2012.
- [8] J. D. Allen, et al. "The Unicode® Standard", version 8.0, [En línea], Disponible: <http://www.unicode.org/versions/Unicode8.0.0/UnicodeStandard-8.0.pdf>