

**ANÁLISIS DEL DESEMPEÑO DE UN SISTEMA DE  
COMUNICACIÓN DIGITAL BASADO EN HARDWARE  
RECONFIGURABLE QUE IMPLEMENTA CODIFICACIÓN DE LÍNEA**



Oscar Iván Velásquez Andrade  
Javier Alexis Urresty Sánchez

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telecomunicaciones  
GRIAL - Grupo de Radio e InALámbricas  
GNTT - Grupo I+D Nuevas Tecnologías en Telecomunicaciones  
Señales y Sistemas de Acceso y Difusión Basados en Radio  
Señales y Sistemas de Telecomunicaciones  
Popayán 2016**



**ANÁLISIS DEL DESEMPEÑO DE UN SISTEMA DE  
COMUNICACIÓN DIGITAL BASADO EN HARDWARE  
RECONFIGURABLE QUE IMPLEMENTA CODIFICACIÓN DE LÍNEA**



Trabajo de Grado presentado como requisito para obtener el título de  
Ingeniero en Electrónica y Telecomunicaciones

Oscar Iván Velásquez Andrade  
Javier Alexis Urresty Sánchez

Director  
Mag. Víctor Fabián Miramá Pérez

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telecomunicaciones  
GRIAL - Grupo de Radio e InALámbricas  
GNTT - Grupo I+D Nuevas Tecnologías en Telecomunicaciones  
Señales y Sistemas de Acceso y Difusión Basados en Radio  
Señales y Sistemas de Telecomunicaciones  
Popayán 2016**



## AGRADECIMIENTOS

Siempre a Dios y la Virgen por iluminar mi camino y acompañarme todo el tiempo.  
A mis padres, Isabel A. y Raúl V. por todo su apoyo, esfuerzo, sacrificio y sobre todo por confiar en mí... Gracias Mamá, gracias Papá.

A mi hermano Olmer por sacrificar su tiempo y bienestar en tiempos de crisis a fin de cumplir nuestros sueños en familia. ¡A mi hermanita Isabella que tanta felicidad nos ha traído con su ternura y cariño! A mis abuelos Rosa, Alicia, Vicente V. y Vicente A. (q.e.p.d.) por todas sus bendiciones y afables deseos.

Y finalmente, pero no menos importante, a mi novia Adriana P. por todo el apoyo, comprensión, paciencia y amor incondicional. :\*

¡A todas aquellas personas que hicieron posible la realización de este gran sueño,  
muchas gracias!

Oscar Velásquez

Gracias Dios por permitirme llegar a cumplir este sueño y acompañarme en todo momento. A mis padres María del Socorro Sánchez Sánchez y Celio Urresty Mesa por brindarme su apoyo incondicional, aconsejarme en cada momento de dificultad y esforzarse por darme lo mejor. A mi hermano Andrés por su compañía y ayuda cuando lo necesitaba.

A mis abuelos Nícomedes Urresty y Servio Sánchez (q.e.p.d), Hermila Mesa y Margarita Sánchez, que siempre estuvieron pendientes de mí y me desearon siempre lo mejor.

A mi novia Roxsana que durante el tiempo que hemos compartido siempre me ha apoyado y ha dado alegría a mi vida.

Gracias familia, amigos y a todas aquellas personas que de una u otra forma me apoyaron.

Javier Urresty Sánchez





## TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1 .....	3
1.1. SISTEMAS DE COMUNICACIÓN DIGITAL .....	3
1.2. TRANSMISIÓN DIGITAL EN BANDA BASE .....	3
1.3. CODIFICACIÓN DE LÍNEA.....	6
1.4. CÓDIGO DE LÍNEA.....	7
1.5. CLASIFICACIÓN CÓDIGOS DE LÍNEA.....	8
1.5.1.Según la polaridad.....	8
1.5.2.Según el tiempo de duración del pulso.....	10
1.5.3.Códigos diferenciales .....	11
1.5.4.Códigos bifase .....	11
1.5.5.Códigos pseudoternarios.....	12
1.6. TIPOS DE CÓDIGOS DE LÍNEA .....	12
1.6.1.No Retorno a Cero Unipolar (UNRZ) .....	13
1.6.2.Retorno a Cero Unipolar (URZ) .....	17
1.6.3.No Retorno a Cero Polar (PNRZ).....	19
1.6.4.Retorno a Cero Polar (PRZ).....	22
1.6.5.No Retorno a Cero Invertido (NRZI) .....	24
1.6.6.Manchester o Bifase L .....	25
1.6.7.Código Manchester Diferencial.....	28
1.6.8.Inversión de Marcas Alternadas sin Retorno a Cero (AMI-NRZ) .....	30
1.6.9.Inversión de Marcas Alternadas con Retorno a Cero (AMI-RZ).....	32
CAPÍTULO 2 .....	37
2.1. METODOLOGÍA DE TRABAJO .....	37
2.2. FASE 0: RECOLECCIÓN DE INFORMACIÓN Y ESPECIFICACIONES.....	38
2.2.1.Requisitos de los elementos software y hardware .....	38
2.2.2.Búsqueda de herramientas de desarrollo .....	38
2.3. FASE 1: SELECCIÓN DE HERRAMIENTAS.....	43
2.4. FASE 2. DISEÑO DEL SISTEMA.....	44



2.4.1. Modelo de referencia.....	45
2.4.2. Diseño del modelo de referencia.....	45
2.5. FASE 3. SIMULACIÓN E IMPLEMENTACIÓN FÍSICA DEL SISTEMA .....	67
2.5.1 Simulación .....	67
2.5.2 Implementación Hardware Del Sistema .....	73
2.6. FASE 4. VALIDACIÓN Y PRUEBAS .....	74
2.6.1. Validación .....	74
CAPÍTULO 3 .....	83
3.1. PLAN DE PRUEBAS.....	83
3.2. RESULTADOS Y ANÁLISIS .....	83
3.2.1. Escenario 1.....	84
3.2.2. Escenario 2.....	90
3.2.3. Análisis de desempeño y Eficiencia Espectral .....	96
CAPÍTULO 4 .....	99
4.1. CONCLUSIONES.....	99
4.2. RECOMENDACIONES.....	100
4.3. TRABAJOS FUTUROS.....	101
BIBLIOGRAFÍA.....	103
ANEXOS.....	107
ANEXO A. ENERGÍA Y POTENCIA .....	107
ANEXO B. TASA DE ERROR DE BIT DE LOS CÓDIGOS DE LÍNEA.....	113



## LISTA DE FIGURAS

Figura 1.1. Representación del pulso rectangular. ....	4
Figura 1.2. Sistema de comunicación digital implementando codificación de línea. ....	6
Figura 1.3. Diagrama de bloques sistema de comunicación digital con codificación de línea. ....	7
Figura 1.4. Código de Línea Unipolar en el dominio del tiempo. ....	9
Figura 1.5. Código Polar en el dominio del tiempo. ....	9
Figura 1.6. Código Bipolar en el dominio del tiempo. ....	10
Figura 1.7. Código con retorno a cero en el dominio del tiempo. ....	10
Figura 1.8. Código sin retorno a cero en el dominio del tiempo. ....	10
Figura 1.9. Código diferencial en el dominio del tiempo. ....	11
Figura 1.10. Código bifase en el dominio del tiempo. ....	11
Figura 1.11. Código pseudoternario en el dominio del tiempo. ....	12
Figura 1.12. Clasificación de los códigos de línea a analizar. ....	12
Figura 1.13. Codificación UNRZ en el dominio del tiempo. ....	13
Figura 1.14. Forma de pulso para el código UNRZ. ....	15
Figura 1.15. Espectro de potencia código UNRZ. ....	16
Figura 1.16. Codificación URZ en el dominio del tiempo. ....	17
Figura 1.17. Forma de pulso para el código URZ. ....	18
Figura 1.18. Espectro de potencia código URZ. ....	19
Figura 1.19. Codificación PNRZ en el dominio del tiempo. ....	20
Figura 1.20. Espectro de Potencia del código PNRZ. ....	21
Figura 1.21. Codificación PRZ en el dominio del tiempo. ....	22
Figura 1.22. Espectro de potencia código PRZ. ....	23
Figura 1.23. Codificación NRZI en el dominio del tiempo. ....	24
Figura 1.24. Espectro de Potencia código NRZI. ....	25
Figura 1.25. Codificación Manchester. ....	26
Figura 1.26. Forma de onda para la codificación Manchester. ....	27
Figura 1.27. Espectro de potencia código Manchester. ....	27
Figura 1.28. Codificación Manchester Diferencial. ....	29
Figura 1.29. Espectro de potencia código Manchester Diferencial. ....	30
Figura 1.30. Codificación AMI-NRZ. ....	31
Figura 1.31. Espectro de potencia código AMI-NRZ. ....	32



Figura 1.32. Codificación AMI-RZ.....	33
Figura 1.33. Espectro de potencia código AMI-RZ.....	34
Figura 1.34. Comparación de la PSD para los diferentes códigos de línea. ....	35
Figura 2.1. Metodología de trabajo. ....	37
Figura 2.2. FPGA <i>Spartan 3A XC3S700A</i> . ....	43
Figura 2.3. Modelo de referencia del sistema de comunicaciones. ....	45
Figura 2.4. Bloque Fuente de información.....	45
Figura 2.5. Bloque <i>AWGN Channel</i> . ....	46
Figura 2.6. Bloque que registra la Tasa de Error de Bit. ....	47
Figura 2.7. Bloque Control UART. ....	48
Figura 2.8. Estructura del bloque <i>System Control UART</i> .....	49
Figura 2.9. Estructura del bloque <i>Frame ASCII Encoder</i> . ....	50
Figura 2.10. Estructura de la trama de transmisión serial. ....	50
Figura 2.11. Estructura del subsistema <i>UART Transmitter</i> . ....	51
Figura 2.12. Estructura del bloque <i>Serial TX Control</i> . ....	51
Figura 2.13. Sistema de Control Principal. ....	52
Figura 2.14. Configuraciones del bloque <i>Global System Control</i> . ....	54
Figura 2.15. Estructura del bloque <i>Global System Control</i> . ....	55
Figura 2.16. Configuración para el cálculo de <i>Sigma</i> ( $\sigma$ ) en el bloque <i>Sigma Control</i> . ....	57
Figura 2.17. Normalizador y Desnormalizador de Potencia.....	57
Figura 2.18. Bloque sincronizador. ....	58
Figura 2.19. Bloque de muestreo.....	59
Figura 2.20. Bloque de decisión de símbolos.....	59
Figura 2.21. Codificador y Decodificador NRZI. ....	61
Figura 2.22. Codificador y Decodificador PNRZ.....	61
Figura 2.23. Codificador y Decodificador URZ. ....	62
Figura 2.24. Codificador y Decodificador PRZ. ....	62
Figura 2.25. Codificador y Decodificador AMI-NRZ. ....	63
Figura 2.26. Codificador y Decodificador AMI-NRZ. ....	63
Figura 2.27. Codificador y Decodificador MANCHESTER OR BIPHASE – L. ....	64
Figura 2.28. Codificador y Decodificador MANCHESTER DIFERENTIAL. ....	64
Figura 2.29. Sistema de comunicación digital con codificación UNRZ.....	65
Figura 2.30. Sistema de comunicación digital para diferentes codificaciones de línea.....	66
Figura 2.31. Metodología de simulación [23].....	67



Figura 2.32. Modelo de simulación para el código UNRZ.....	69
Figura 2.33. Bloque para el cálculo de potencia media de una señal aleatoria. ....	70
Figura 2.34. Formas de onda de la simulación del código UNRZ.....	72
Figura 2.35. Proceso de implementación Hardware. ....	73
Figura 2.36. Captura de la tasa de error de bit en la aplicación de escritorio. ....	75
Figura 2.37. Curva de desempeño codificación UNRZ. ....	77
Figura 2.38. Curva de desempeño codificación NRZ-I.....	77
Figura 2.39. Curva de desempeño codificación PNRZ. ....	78
Figura 2.40. Curva de desempeño codificación URZ.....	78
Figura 2.41. Curva de desempeño codificación PRZ.....	79
Figura 2.42. Curva de desempeño codificación AMI-NRZ. ....	79
Figura 2.43. Curva de desempeño codificación AMI-RZ.....	80
Figura 2.44. Curva de desempeño codificación Manchester or Biphase-L. ....	80
Figura 2.45. Curva de desempeño codificación Differential Manchester.....	81
Figura 3.1. Curvas de desempeño para la codificación UNRZ. Transmisión de 1536 bits.	84
Figura 3.2. Curvas de desempeño para la codificación NRZ-I. Transmisión de 1536 bits.	85
Figura 3.3. Curvas de desempeño para la codificación PNRZ. Transmisión de 1536 bits.	86
Figura 3.4. Curvas de desempeño para la codificación URZ. Transmisión de 1536 bits. ..	86
Figura 3.5. Curvas de desempeño para la codificación PRZ. Transmisión de 1536 bits. ..	87
Figura 3.6. Curvas de desempeño para la codificación AMI-NRZ. Transmisión de 1536 bits. ....	88
Figura 3.7. Curvas de desempeño para la codificación AMI-RZ. Transmisión de 1536 bits. ....	88
Figura 3.8. Curvas de desempeño para la codificación MANCHESTER OR BIPHASE L. Transmisión de 1536 bits. ....	89
Figura 3.9. Curvas de desempeño para la codificación DIFFERENTIAL MANCHESTER. Transmisión de 1536 bits. ....	89
Figura 3.10. Curvas de desempeño para la codificación UNRZ. Transmisión de 1048576 bits. ....	91
Figura 3.11. Curvas de desempeño para la codificación NRZ-I. Transmisión de 1048576 bits. ....	91
Figura 3.12. Curvas de desempeño para la codificación PNRZ. Transmisión de 1048576 bits. ....	92



Figura 3.13. Curvas de desempeño para la codificación URZ. Transmisión de 1048576 bits. ....	92
Figura 3.14. Curvas de desempeño para la codificación PRZ. Transmisión de 1048576 bits. ....	93
Figura 3.15. Curvas de desempeño para la codificación AMI-NRZ. Transmisión de 1048576 bits. ....	93
Figura 3.16. Curvas de desempeño para la codificación AMI-RZ. Transmisión de 1048576 bits. ....	94
Figura 3.17. Curvas de desempeño para la codificación MANCHESTER OR BIPHASE L. Transmisión de 1048576 bits. ....	94
Figura 3.18. Curvas de desempeño para la codificación DIFFERENTIAL MANCHESTER. Transmisión de 1048576 bits. ....	95
Figura 3.19. Comparación de curvas de desempeño para los diferentes esquemas de codificación implementados sobre hardware reconfigurable. ....	98



## LISTA DE TABLAS

Tabla 1.1. Códigos de línea considerados en el presente trabajo de grado.....	8
Tabla 1.2. Transiciones de la codificación Manchester.....	26
Tabla 1.3. Transiciones codificación Manchester Diferencial. ....	29
Tabla 2.1. Valores de $A$ para potencia normalizada.....	58
Tabla 2.2. Umbrales de decisión para cada código de línea. ....	59
Tabla 2.3. Probabilidad de Error .....	76
Tabla 3.1. Escenarios de plan de pruebas. ....	83
Tabla 3.2. Eficiencia espectral de los códigos de línea.....	96



ANÁLISIS DEL DESEMPEÑO DE UN SISTEMA DE COMUNICACIÓN DIGITAL BASADO EN HARDWARE  
RECONFIGURABLE QUE IMPLEMENTA CODIFICACIÓN DE LÍNEA

---



## ACRÓNIMOS

<b>AMI</b>	<i>Alternate Mark Inversion.</i> (Inversión de Marcas Alternadas).
<b>ASCII</b>	<i>American Standard Code for Information Interchange.</i> (Código Estándar Americano para el Intercambio de Información).
<b>ASIC</b>	<i>Application-Specific Integrated Circuit.</i> (Circuitos Integrados de Aplicación Específica)
<b>AWGN</b>	<i>Additive White Gaussian Noise.</i> (Ruido Blanco Gaussiano Aditivo).
<b>BER</b>	<i>Bit Error Rate.</i> (Tasa de Error de Bit).
<b>BRZ</b>	<i>Bipolar Return to Zero.</i> (Retorno a Cero Unipolar).
<b>CMI</b>	<i>Code Mark Inversion.</i> (Código por Inversión de Marcas).
<b>CPLD</b>	<i>Complex Programmable Logic Device.</i> (Dispositivo Lógico Programable Complejo).
<b>DC</b>	<i>Direct Current.</i> (Corriente Directa)
<b>DSP</b>	<i>Digital Signal Processing.</i> (Procesamiento Digital de Señales)
<b>FIET</b>	Facultad de Ingeniería Electrónica y Telecomunicaciones.
<b>FPGA</b>	<i>Field Programmable Gate Array.</i> (Arreglo de Compuertas Programables en Campo).
<b>GCC</b>	<i>GNU Compiler Collection.</i> (Colección de Compiladores de GNU)
<b>GNTT</b>	Grupo de Nuevas Tecnologías en Telecomunicaciones.
<b>GRIAL</b>	Grupo de Radio e InALámbricas.
<b>GUI</b>	<i>Graphical User Interface.</i> (Interfaz Gráfica de Usuario)
<b>HDB</b>	<i>High Density Bipolar.</i> (Alta Densidad Bipolar).
<b>HDB3</b>	<i>High Density Bipolar-3 Zeros.</i> (Alta Densidad Bipolar de 3 Ceros).
<b>HDL</b>	<i>Hardware Description Language.</i> (Lenguaje de Descripción de Hardware)
<b>IDE</b>	<i>Integrated development environment.</i> (Ambiente de Diseño Integrado)
<b>IEEE</b>	<i>Institute of Electrical and Electronics Engineers.</i> (Instituto de Ingenieros en Electricidad y Electrónica).
<b>IF</b>	<i>Intermediate Frequency.</i> (Frecuencia Intermedia)



<b>ISE</b>	<i>Integrated Software Environment.</i> (Ambiente de Software Integrado)
<b>ISI</b>	<i>Intersymbol Interference.</i> (Interferencia Intersimbólica)
<b>LAN</b>	<i>Local Area Network.</i> (Redes de Área Local)
<b>LFSR</b>	<i>Linear Feedback Shift Register.</i> (Registro de Desplazamiento con Realimentación Lineal)
<b>MLT</b>	<i>Multi-Level Transmit.</i> (Transmisión Multinivel).
<b>NRZ</b>	<i>Non Return to Zero.</i> (No Retorno a Cero).
<b>NRZI</b>	<i>Non Return to Zero Inverted.</i> (No Retorno a Cero Invertido).
<b>PDF</b>	<i>Probability Density Function.</i> (Función de Densidad de Probabilidad)
<b>PNRZ</b>	<i>Polar Non Return to Zero.</i> (No Retorno a Cero Polar).
<b>PPG</b>	<i>Programmable Pulse Generator.</i> (Generador de pulsos programable).
<b>PROM</b>	<i>Programmable Read-only Memory.</i> (Memoria Programable de solo Lectura)
<b>PRZ</b>	<i>Polar Return to Zero.</i> (Retorno a Cero Polar).
<b>PSD</b>	<i>Power Spectral Density.</i> (Densidad Espectral de Potencia)
<b>RAM</b>	<i>Random Access Memory.</i> (Memoria de Acceso Aleatorio)
<b>RF</b>	<i>Radio Frequency.</i> (Radio Frecuencia)
<b>RZ</b>	<i>Return to Zero.</i> (Retorno a Cero).
<b>SDCC</b>	<i>Small Device C Compiler.</i> (Pequeño Dispositivo Compilador de C)
<b>SNR</b>	<i>Signal to Noise Ratio.</i> (Relación Señal a Ruido)
<b>TTL</b>	<i>Transistor-Transistor Logic.</i> (Lógica Transistor-Transistor)
<b>UART</b>	<i>Universal Asynchronous Receiver-Transmitter.</i> (Transmisor-Receptor Asíncrono Universal)
<b>UDM</b>	<i>Universal Design Methodology.</i> (Metodología Universal de Diseño)
<b>UNRZ</b>	<i>Unipolar Non Return to Zero.</i> (No Retorno a Cero Unipolar).
<b>URZ</b>	<i>Unipolar Return to Zero.</i> (Retorno a Cero Unipolar).
<b>USB</b>	<i>Universal Serial Bus.</i> (Bus Serial Universal)



<b>USRP</b>	<i>Universal Software Radio Peripheral.</i> (Software Universal de Radio Periférico)
<b>VHDL</b>	<i>VHSIC Hardware Description Language.</i> (Lenguaje de Descripción Hardware VHSIC).
<b>VHSIC</b>	<i>Very High Speed Integrated Circuit.</i> (Circuito Integrado de Alta Velocidad).
<b>VLSI</b>	<i>Very Large Scale Integration.</i> (Integración en escala muy grande).
<b>WGNG</b>	<i>White Gaussian Noise Generator.</i> (Generador de Ruido Blanco Gaussiano)
<b>XGS</b>	<i>System Generator Xilinx.</i> (Sistema Generador de Xilinx)



ANÁLISIS DEL DESEMPEÑO DE UN SISTEMA DE COMUNICACIÓN DIGITAL BASADO EN HARDWARE  
RECONFIGURABLE QUE IMPLEMENTA CODIFICACIÓN DE LÍNEA

---



# INTRODUCCIÓN

Los sistemas digitales gobiernan hoy en día la mayor parte del mundo tecnológico, haciendo que nuestras actividades cotidianas sean cada vez más ágiles y precisas. En los últimos tiempos se ha venido trabajando por mejorar el aprovechamiento de los recursos utilizados para el intercambio de información a través de los sistemas de comunicación digital, logrando que millones de personas puedan interconectarse cada día más.

Con el origen de los dispositivos digitales se da paso a las técnicas de procesamiento digital de señales, que incorporan elementos hardware para crear sistemas de comunicación digital fiables y de alto rendimiento. Con ayuda de los criterios y la teoría de las comunicaciones, los dispositivos hardware ayudan a analizar y estudiar el comportamiento físico de las señales eléctricas. En la actualidad existe una gran variedad de herramientas tanto software como hardware sofisticadas que permiten para entornos académicos validar los conceptos de un sistema de comunicación digital y la teoría detrás de su funcionamiento. Generalmente este tipo de sistemas no son asequibles a bajo costo por lo que existe la necesidad de encontrar tecnologías y métodos económicos que permitan implementar estos sistemas de forma flexible y reconfigurable para aprovechar los recursos disponibles.

Actualmente el uso de dispositivos hardware reconfigurables está siendo implementado en muchas áreas de la comunicación, con el propósito de disminuir costos, obteniendo resultados comparables con sistemas robustos y eficientes y de alto costo. Dentro de estos dispositivos se destaca los Arreglos de Compuertas Programables en Campo (FPGA, *Field Programmable Gate Array*) que por su capacidad, flexibilidad y economía son muy utilizados en varios entornos.

El presente trabajo de grado tiene como objetivo analizar el comportamiento de un sistema de comunicación digital basado en FPGA como hardware reconfigurable, el cual implementa diferentes esquemas de codificación de línea. Esto como un aporte al proyecto “Diseño e Implementación de un Prototipo de Comunicación de Datos Basado en Hardware Reconfigurable” [1] en su segunda fase de desarrollo.



La estructura del trabajo está conformada por cuatro capítulos. El primer capítulo expone las bases teóricas en las que se fundamentan cada uno de los códigos de línea y su correspondiente proceso de codificación.

El segundo, está dedicado al diseño, simulación, implementación y validación de un sistema de comunicación digital con codificación línea basado en hardware reconfigurable FPGA.

El tercer capítulo expone las pruebas realizadas con el fin de evaluar el desempeño de los códigos de línea a nivel de la Tasa de Error de Bit (BER, *Bit Error Rate*) dada en función de la relación Energía de bit a Densidad Espectral de Potencia de Ruido  $E_s/N_0$ .

Finalmente, el cuarto capítulo resalta las conclusiones a partir del análisis de los resultados obtenidos en el capítulo 3, culminando con una serie de recomendaciones y propuestas de futuros trabajos.



# CAPÍTULO 1

## GENERALIDADES

### 1.1. SISTEMAS DE COMUNICACIÓN DIGITAL

Un sistema de comunicación es aquel capaz de transmitir información de un lugar a otro en el tiempo y en el espacio. La comunicación se entabla mediante un emisor y un receptor separados en el espacio.

En un sistema de comunicación digital, la información transmitida es representada en datos binarios. Si bien la información es considerada digital, esta debe pasar por un proceso de codificación que convierta sus datos binarios en señales apropiadas para ser transmitidas a través de un canal físico o medio de transmisión. El proceso puede considerarse como una conversión digital a digital o codificación de los datos digitales dentro de una señal digital [2].

El canal de comunicación se constituye por un medio físico, a través del cual se pueden transmitir las señales de datos originadas en el emisor, una de sus principales características es el ancho de banda del canal que define la capacidad de permitir el paso de una gama de frecuencias, a través del medio por el cual se entabla la comunicación [8]. En otras palabras, el ancho de banda del canal define la cantidad de información que puede ser transmitida en un intervalo de tiempo, concepto fundamentado en el teorema de la capacidad de canal formulado por Claude Shannon [11].

Por otra parte, las señales digitales se pueden transmitir por el canal ya sea mediante un proceso de modulación o bien en banda base [7], siendo esta última forma de transmisión la que se tiene en cuenta en el presente trabajo de grado.

### 1.2. TRANSMISIÓN DIGITAL EN BANDA BASE

Las señales que no sufren ningún proceso de modulación ni desplazamiento en frecuencia, se denominan señales en banda base [4], [5]. En el caso de las señales digitales se denominan códigos en banda base o códigos de línea, los cuales se analizarán en la sección 1.3 del presente capítulo.

Un dígito binario puede ser representado por un pulso rectangular de duración  $T$  como el mostrado en la figura 1.1.a y su transformada de Fourier se representa por la figura 1.1.b.

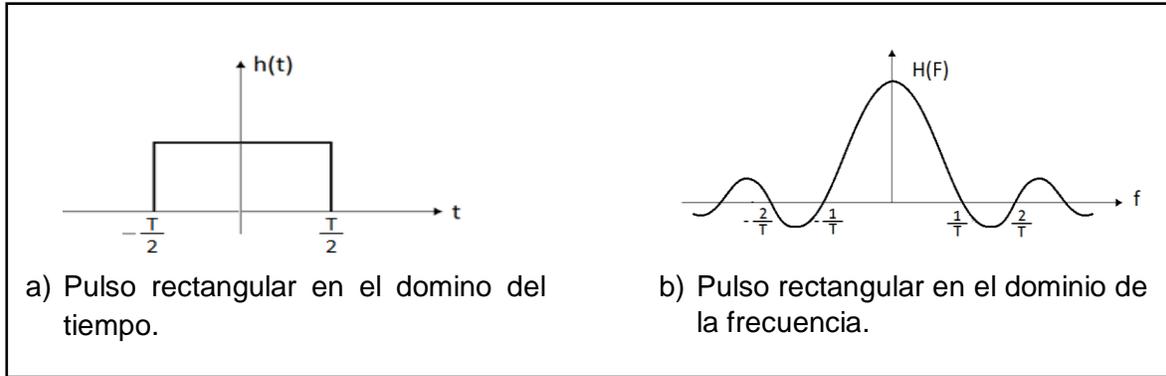


Figura 1.1. Representación del pulso rectangular.

Matemáticamente la función  $h(t)$  está representada por la función rectangular [9]

$$h(t) = \Pi\left(\frac{t}{T}\right) \quad (1.1)$$

y su transformada de Fourier mediante [9], [18]:

$$H(f) = \int_{-T/2}^{T/2} h(t)e^{-j2\pi ft} dt \quad (1.2)$$

$$= \frac{\sin(\pi fT)}{\pi f}$$

$$= Tsinc(Tf). \quad (1.3)$$

En la gráfica 1.1.b se observa que el pulso está compuesto por un número infinito de frecuencias [10]. En la práctica no es posible transmitir completamente una señal de este tipo, ya que no se cuenta con canales de transmisión con un ancho de banda infinito. Sin embargo, es posible transmitir cierta parte de las componentes de frecuencia del pulso, sin afectar en gran medida su reconstrucción en el receptor [12]. Además se observa que el pulso posee una componente de corriente directa o continua (DC, *Direct Current*)<sup>1</sup> muy alta, tal como lo muestra la figura 1.1.b con una gran cantidad de frecuencias cercanas a cero [2], [15].

<sup>1</sup> Corriente Directa: Flujo de corriente en un solo sentido o con una misma polaridad durante tiempos prolongados [13].



La transmisión digital en banda base, involucra inconvenientes como el gran ancho de banda requerido y una componente de corriente continua muy alta que suelen tener las señales de información, lo cual no es deseable para sistemas que no pueden gestionar este tipo de problemas, por ejemplo, una línea telefónica no puede pasar frecuencias por debajo de los 300 Hz, o en un sistema que se acopla con transformadores las componentes continuas no se transmiten [2], [15]. Adicionalmente una señal (analógica o digital) que atraviesa el canal de comunicaciones sufre tres tipos de fenómenos que hacen variar su forma de onda original: atenuación, distorsión y ruido [14].

La atenuación es una disminución en la amplitud de la señal a medida que ésta va recorriendo el medio de transmisión. La atenuación sufrida por la señal es proporcional a la distancia recorrida, por lo que generalmente suele especificarse en decibeles por metro (dB/m) [14].

En tanto que la distorsión provoca una deformación de la señal original al atravesar el canal, esto debido a las características inductivas y capacitivas de los diferentes medios de transmisión [14].

El ruido es toda perturbación o interferencia no deseada que se introduce en el canal de comunicaciones y se suma a la señal transmitida. Existen múltiples fuentes de ruido, unas externas y otras internas al sistema de comunicaciones [14].

Además, si se tiene un canal con un ancho de banda finito, algunas componentes de frecuencia de la señal digital no llegan a su destino, pues el canal lo impide, ya que se presenta un efecto de filtraje sobre la señal, que a su vez ocasiona la dispersión de los pulsos rectangulares, lo cual hace que parte de la energía del símbolo se solape con los símbolos vecinos, causando así lo que se conoce como interferencia entre símbolos (ISI, *Inter Symbol Interference*). El efecto de la ISI degrada la capacidad del receptor para diferenciar un símbolo real a partir de la energía que se ha solapado entre símbolos adyacentes, inclusive sin haber presencia de ruido puede haber errores de detección [2], [3].

Para mitigar los efectos y problemas presentados anteriormente, la señal digital debe de alguna manera acoplarse al medio por el cual será transmitida, de tal forma que los

efectos del canal y de otros factores puedan ser tolerables. Para acoplar la señal al canal se lleva a cabo un proceso que da un formato físico a los datos binarios. El proceso es considerado como un tipo de modulación y se denomina codificación de línea, el cual es el tema central de este trabajo de grado.

### 1.3. CODIFICACIÓN DE LÍNEA

La codificación de línea suele conocerse también como formato de señalización, el nombre es atribuido por sus aplicaciones iniciales sobre sistemas de telefonía alambrados o líneas telefónicas, aunque hoy en día 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 [2].

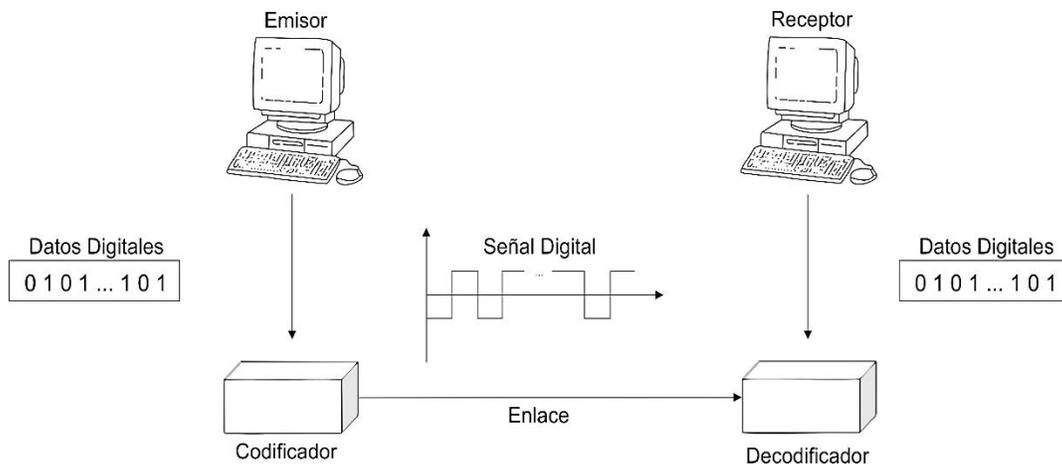


Figura 1.2. Sistema de comunicación digital implementando codificación de línea. Adaptado de [2].

Situando los códigos de línea en el contexto de un sistema de comunicaciones digital como el de la figura 1.2, se tiene como propósito transmitir información desde un emisor a un receptor a través de un enlace o canal. A partir de esto se formula el diagrama de bloques mostrado en la figura 1.3.

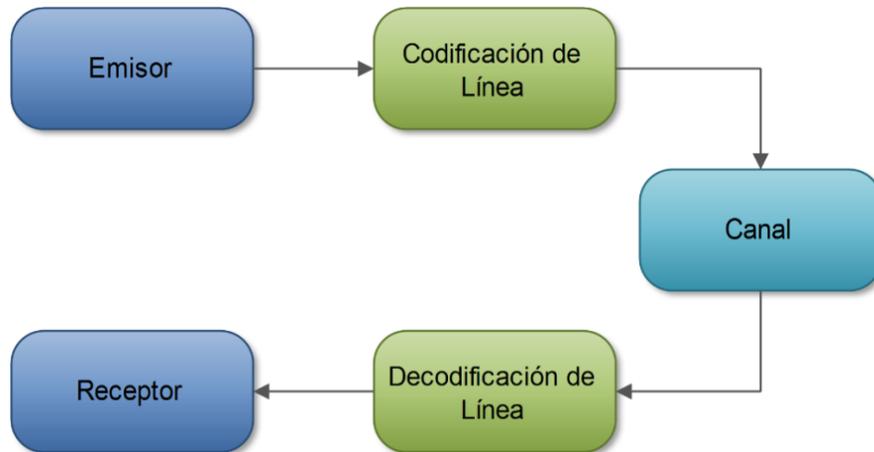


Figura 1.3. Diagrama de bloques sistema de comunicación digital con codificación de línea.

En el bloque emisor se generan las señales discretas aleatorias y se emiten los diferentes bits que se serán transmitidos. Los bits emitidos llegan al bloque codificador de línea donde son transformados mediante un formato de señalización en símbolos o señales más adecuados para ser transmitidas a través del canal el cual les adiciona ruido.

El formato de señalización depende de diferentes factores como los recursos disponibles, requerimientos del sistema y características del canal de transmisión.

Ya en el receptor se realiza el proceso inverso, el decodificador de línea convierte la señal recibida en señales discretas y se entregan al destino como datos binarios.

#### 1.4. CÓDIGO DE LÍNEA

Un código de línea es utilizado en sistemas de comunicación para el envío de información 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.

En el presente trabajo de grado, se estudiaron los siguientes códigos de línea, clasificados según su polaridad y relacionados en la tabla 1.1.



Tabla 1.1. Códigos de línea considerados en el presente trabajo de grado.

Códigos Unipolares	<ul style="list-style-type: none"><li>• UNRZ (<i>Unipolar Non Return to Zero</i>. No Retorno a Cero Unipolar).</li><li>• URZ (<i>Unipolar Return to Zero</i>. Retorno a Cero Unipolar).</li></ul>
Códigos Polares	<ul style="list-style-type: none"><li>• PNRZ (<i>Polar Non Return to Zero</i>. No Retorno a Cero Polar).</li><li>• PRZ (<i>Polar Return to Zero</i>. Retorno a Cero Polar).</li><li>• NRZI (<i>Non Return to Zero Inverted</i>. No Retorno a Zero Invertido).</li><li>• Manchester. En sus variantes Manchester o Bifase L y Manchester Diferencial.</li></ul>
Códigos Bipolares	<ul style="list-style-type: none"><li>• AMI (<i>Alternate Mark Inversion</i>, Inversión de Marcas Alternadas). En sus variantes AMI-RZ y AMI-NRZ.</li></ul>

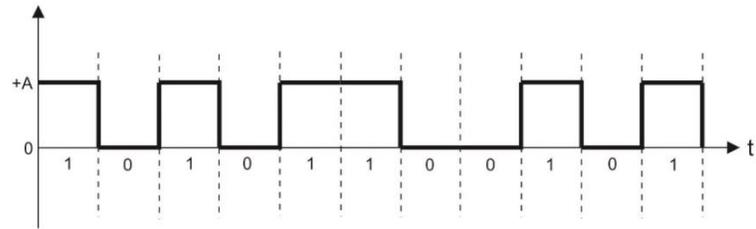
Los anteriores códigos de línea son la base para la conformación de códigos más complejos y de mayor rendimiento como códigos multiniveles. Razón por la cual se tienen en cuenta para su implementación como aporte al proyecto en el cual se enmarca este trabajo de grado, creando una herramienta de uso académico en el estudio de la codificación de línea y validar la teoría detrás de su funcionamiento.

## 1.5. CLASIFICACIÓN CÓDIGOS DE LÍNEA

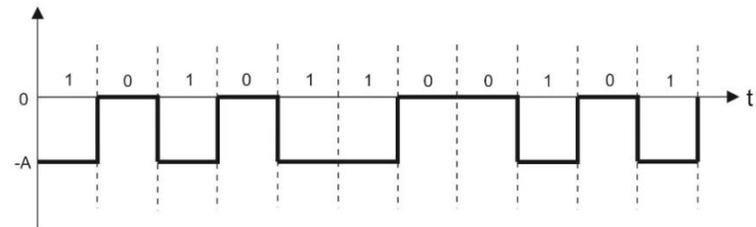
### 1.5.1. Según la polaridad

#### 1.5.1.1. Códigos Unipolares

Un dígito binario (“1” o “0”) se representa mediante un nivel de tensión positivo, mientras que el otro dígito con un nivel de tensión cero. El código es llamado Unipolar Positivo (Figura 1.4.a) siendo este el más utilizado, cuando un “1” binario toma un nivel de tensión alto y llamado código Unipolar Negativo (Figura 1.4.b) en caso contrario [6], [15].



a) Código unipolar positivo.



b) Código unipolar negativo.

Figura 1.4. Código de Línea Unipolar en el dominio del tiempo.

### 1.5.1.2. Códigos Polares

Un dígito binario (“1” o “0”) se representa mediante un nivel de tensión positivo, mientras que el otro con un nivel de tensión negativo. En este tipo de código los niveles opuestos identifican los dos estados lógicos binarios como lo muestra la figura 1.5 [6], [15].

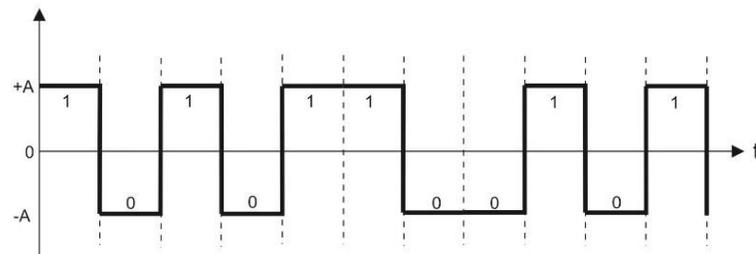


Figura 1.5. Código Polar en el dominio del tiempo.

### 1.5.1.3. Códigos Bipolares

A un dígito binario (“1” o “0”) se le asigna valores de tensión alternados mientras que el otro se mantiene en el nivel cero, es decir, la señal varía entre tres niveles: positivo, cero y negativo, razón por la cual se denomina un código multinivel. Este tipo de código se muestra en la figura 1.6 [6], [15].

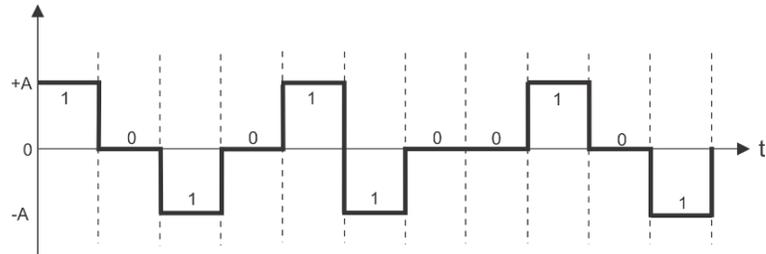


Figura 1.6. Código Bipolar en el dominio del tiempo.

## 1.5.2. Según el tiempo de duración del pulso

### 1.5.2.1. Códigos RZ (*Return to Zero*)

En este caso la forma de onda del código regresa al nivel cero por una fracción del periodo de bit<sup>2</sup> denotado como  $T_b$ , [6], [15]. Un ejemplo de este tipo de código se muestra en la figura 1.7.

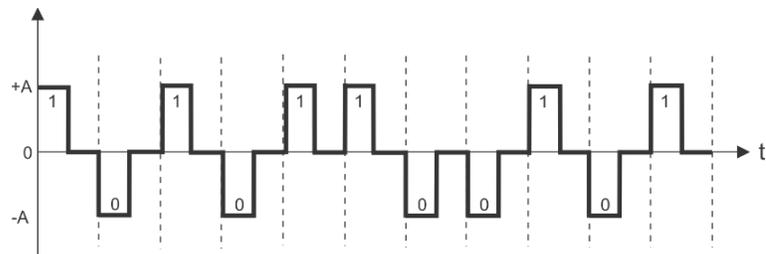


Figura 1.7. Código con retorno a cero en el dominio del tiempo.

### 1.5.2.2. Códigos NRZ (*Non Return to Zero*)

La forma de onda asignada para la transmisión no regresa al nivel cero, durante todo el periodo de bit de la señal [6], [15], tal como se muestra en la figura 1.8.

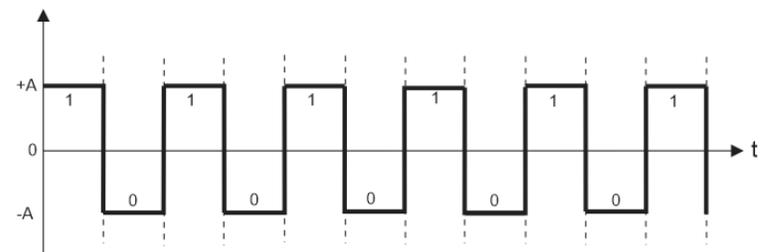
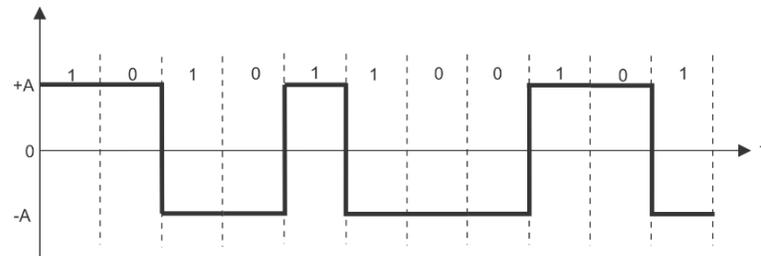


Figura 1.8. Código sin retorno a cero en el dominio del tiempo.

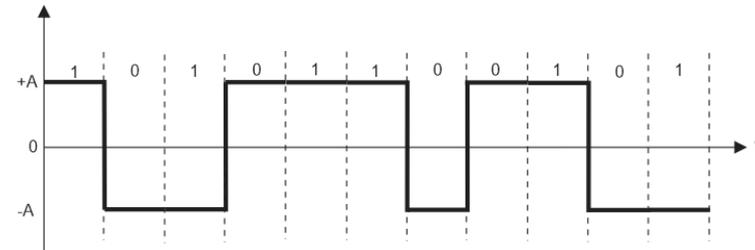
<sup>2</sup> Duración o tiempo de un bit.

### 1.5.3. Códigos diferenciales

El nivel de la señal se invierte con respecto al nivel anterior, según el tipo de lógica implementada se tendrá los códigos tipo M o S. Cuando el código es diferencial tipo M (Figura 1.9.a) el nivel de la señal se invierte al codificar un “1” binario. Para el código diferencial tipo S (Figura 1.9.b) la señal se invierte al codificar un “0” binario [15], [16].



a) Código diferencial tipo M.



b) Código diferencial tipo S.

Figura 1.9. Código diferencial en el dominio del tiempo.

### 1.5.4. Códigos bifase

Son códigos que producen transiciones entre dos niveles, uno positivo y otro negativo, para representar un solo bit [15], [16]. Este código se aprecia en la figura 1.10.

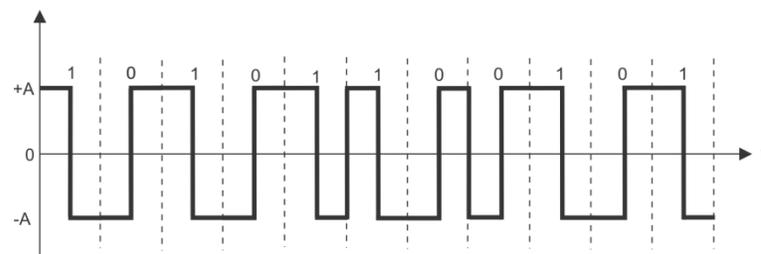


Figura 1.10. Código bifase en el dominio del tiempo.

### 1.5.5. Códigos pseudoternarios

Dos niveles de tensión se utilizan alternadamente para representar un solo dígito binario (“1” o “0”) [15], [16], este código se observa en la figura 1.11.

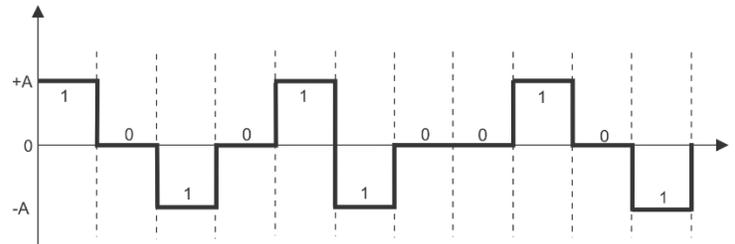


Figura 1.11. Código pseudoternario en el dominio del tiempo.

### 1.6. TIPOS DE CÓDIGOS DE LÍNEA

Teniendo en cuenta la clasificación de la sección 1.5. La figura 1.12 muestra un esquema que clasifica los códigos de línea según la polaridad.

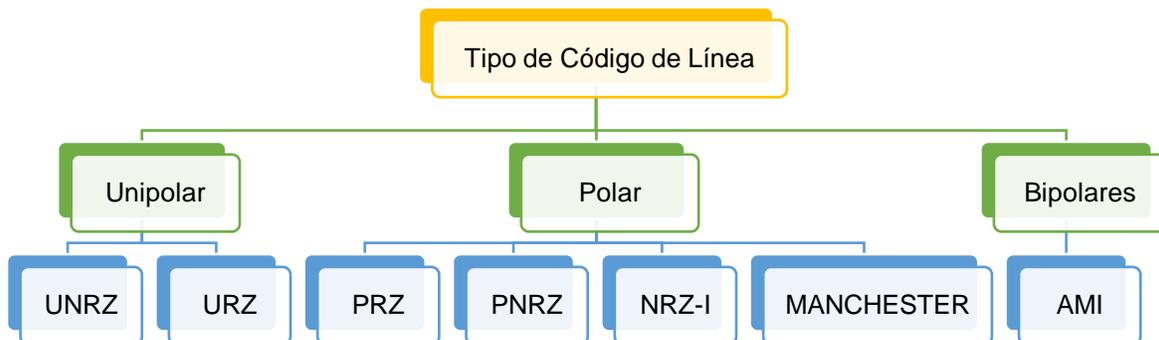


Figura 1.12. Clasificación de los códigos de línea a analizar.

Cada código tiene características que le dan un formato a la señal que se va a transmitir. En las siguientes subsecciones se describe el algoritmo de codificación que indica el tratamiento de los bits según sea el tipo de codificación.

La ecuación que representa una señal dada por la codificación de línea es:

$$\begin{aligned}
 s(t) &= \sum_i a_i \delta(t - iT_b) * h(t) \\
 &= \sum_i a_i h(t - iT_b).
 \end{aligned}
 \tag{1.4}$$

Siendo  $h(t)$  la función que describe la forma de pulso utilizada para codificar los datos binarios y  $a_i$  un símbolo que depende del dígito binario (“1” o “0”) de la secuencia de bits



$\{b_n\}$  a transmitir y de la señalización del esquema de codificación de línea, como se describe a continuación.

### 1.6.1. No Retorno a Cero Unipolar (UNRZ)

En este tipo de codificación cada dígito binario  $b_i$  es representado por un nivel de tensión constante durante todo el periodo de bit [19].

#### Algoritmo de codificación

La señal codificada es una señal con dos niveles que dependen del bit codificado:

1: Nivel de tensión positivo (+A), sin transiciones o cambios de tensión durante el período de bit.

0: Nivel cero (0), sin transiciones durante el período de bit.

De esta manera el valor de  $a_i$  toma los valores:

$$a_i = \begin{cases} 0 & b_i = 0 \\ A & b_i = 1 \end{cases} \quad (1.5)$$

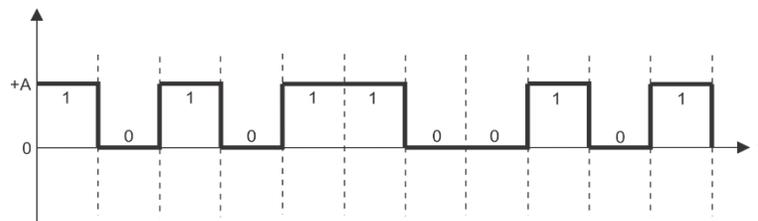


Figura 1.13. Codificación UNRZ en el dominio del tiempo.

Una de las ventajas del código UNRZ ilustrado en la figura 1.13, es su fácil implementación por la sencillez del algoritmo de codificación, lo cual no implica recursos complejos ni costos elevados para su desarrollo, además por ser un código de tipo no retorno a cero la frecuencia de la señal codificada y de la señal de datos binarios son iguales, por lo que se requiere menor ancho de banda que para otros sistemas de codificación que emplean pulsos más cortos como los códigos de tipo retorno a cero [17]. Por otra parte, este código presenta una componente alta de corriente continua DC, característica dada por ser un código de tipo unipolar, siendo una desventaja para sistemas acoplados mediante transformadores y que no permiten el paso de señales con niveles de tensión constantes [19].



Este código es utilizado en lógica digital para la representación física de los estados lógicos “1” y “0”, mediante la tecnología de Lógica Transistor-Transistor (TTL, *Transistor-Transistor Logic*). Esta codificación es la más usada para la transmisión de datos en sistemas que no presentan un alto rendimiento, es decir, que no requieren de tasas de error de bit muy bajas y que además cuentan con un ancho de banda mínimo en comparación a sistemas que utilizan codificaciones más eficientes y confiables<sup>3</sup>. Por lo tanto, se establece al código UNRZ como la referencia para evaluar el desempeño de las demás codificaciones que usualmente son generadas a partir de esta codificación [15] y que se desarrollan en este trabajo de grado.

### Espectro de Potencia Teórico

Para el análisis espectral de los códigos de línea, se tiene en cuenta que la señal codificada es un proceso aleatorio que además se lo considera estacionario, ya que las características estadísticas no varían en el tiempo y según la teoría de probabilidad y procesos estocásticos el espectro de la señal puede ser descrito por la función de densidad espectral de potencia (PSD, *Power Spectrum Density*) [21], la cual se obtiene mediante la transformada de Fourier de la función de autocorrelación de la señal a transmitir o la señal codificada [17]. Entonces, se tiene que:

$$S_s(f) = \mathcal{F}\{R_{ss}(\tau)\} \quad (1.6)$$

$$S_s(f) = |\mathcal{F}\{h(t)\}|^2 \frac{1}{T_b} \sum_k E\{a_i \cdot a_{i+k}\} e^{j2\pi f k T_b}, \quad (1.7)$$

donde:

- $R_{ss}(\tau)$  es la función de autocorrelación de la señal codificada  $s(t)$ .
- $T_b$  es el período de bit de la señal codificada.
- $h(t)$  es la función que describe la forma del pulso.
- $a_i$  es el símbolo que depende del bit  $b_i$  de la secuencia binaria a transmitir.
- $E\{a_i \cdot a_{i+k}\}$  es la correlación de la secuencia de símbolos codificados [9].

---

<sup>3</sup> Eficientes en cuanto a la transmisión de más bits sobre el mismo o menor ancho de banda con respecto a la codificación UNRZ y confiables en cuanto a la incorporación de algoritmos en códigos para el control de errores presentes en la codificación de canal [19].

Para cualquier código de línea considerado de aquí en adelante la ecuación 1.7 describe su espectro de potencia [15] y las ecuaciones para sus PSD correspondientes se obtienen a partir del análisis matemático documentado en el apéndice A.

Para el código UNRZ, la función de la PSD está dada por:

$$S_{SUNRZ}(f) = \frac{A^2 T_b}{4} \text{sinc}^2(T_b f) \left(1 + \frac{1}{T_b} \delta(f)\right). \quad (1.8)$$

La ecuación 1.8 se obtiene a partir de las siguientes consideraciones:

- La secuencia de bits codificados son variables aleatorias binarias independientes<sup>4</sup> e idénticamente distribuidas<sup>5</sup>.
- Los dígitos binarios de la secuencia de bits son equiprobables.
- La función conformadora de los pulsos  $h(t)$  está dada por la función rectangular:

$$h(t) = \Pi\left(\frac{t - T_b/2}{T_b}\right). \quad (1.9).$$

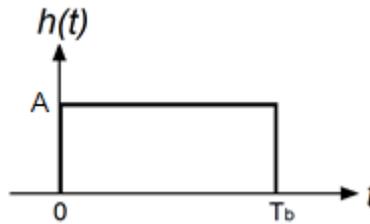


Figura 1.14. Forma de pulso para el código UNRZ.

Por otra parte el valor de la amplitud  $A$  que se muestra en la figura 1.14, se calcula de tal modo que la potencia media de la señal codificada esté normalizada<sup>6</sup>, con el fin de establecer un punto de referencia en todas las codificaciones a razón de comparar su desempeño y eficiencia espectral, por consiguiente es preciso establecer la ecuación para la potencia media de la señal  $s(t)$  a partir de las definiciones de energía y potencia de una señal, mostradas en el anexo A. De esta manera se tiene que:

<sup>4</sup> Cada bit es generado de forma independiente de los demás bits generados por la fuente.

<sup>5</sup> Son variables aleatorias que tienen la misma distribución, es decir, sus funciones de probabilidad son iguales [9].

<sup>6</sup> Es decir que la potencia media de la señal codificada sea igual a  $1W$ .

$$P_s = \frac{1}{T_b} \int_{T_b} |s(t)|^2 dt. \quad (1.10)$$

La ecuación 1.10 queda definida de forma general para cualquier código de línea considerado de ahora en adelante y para el caso del código UNRZ como:

$$P_{S_{UNRZ}} = \frac{A^2}{2}. \quad (1.11)$$

Es claro que para normalizar la potencia el valor de  $A$  es igual a  $\sqrt{2}$  de esta manera la PSD para el código UNRZ se muestra en la figura 1.15 en donde se observa una mayor concentración de energía alrededor de una componente DC muy alta, debido a la presencia de un impulso en la frecuencia 0 Hz, esta es una característica de los códigos unipolares [6], [15].

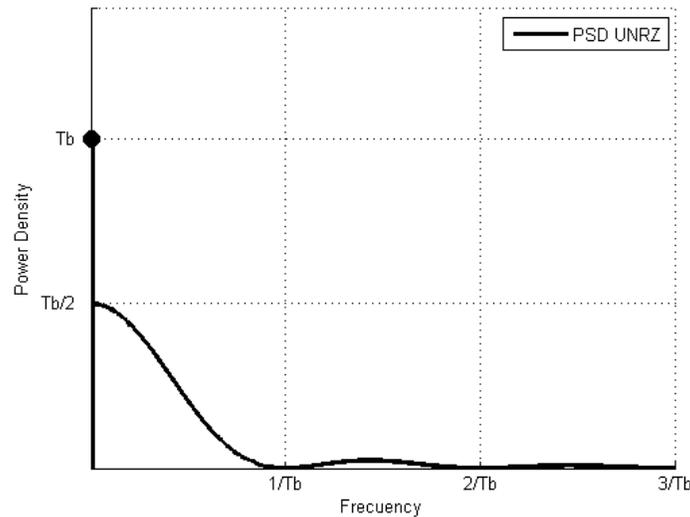


Figura 1.15. Espectro de potencia código UNRZ.

Habiendo considerado al código UNRZ como la codificación de referencia para el análisis de los siguientes códigos de línea, es preciso definir una medida que involucre la velocidad de transmisión de los datos binarios con relación al ancho de banda requerido por la señal codificada, esta medida se define como la eficiencia espectral ( $\eta$ )<sup>7</sup> representada por la ecuación 1.12.

$$\eta = \frac{R_b \text{ bps}}{B \text{ Hz}}. \quad (1.12)$$

<sup>7</sup> Es un parámetro de medida que define el número de bits por segundo que se puede transmitir por cada hercio bajo un determinado esquema de modulación [22].



En donde  $R_b$  es la tasa de transmisión ( $1/T_b$ ) dada en Bits por segundo (bps) y  $B$  es el ancho de banda al primer nulo<sup>8</sup> medido en la figura 1.15. Para la codificación UNRZ la eficiencia espectral es:

$$\eta_{UNRZ} = \frac{1/T_b}{1/T_b} = 1 \frac{bps}{Hz}. \quad (1.13)$$

Esto quiere decir que por cada hercio se transmite información a una velocidad de un bit por segundo.

### 1.6.2. Retorno a Cero Unipolar (URZ)

Es un tipo de codificación en la cual cada bit es representado por una señal que retorna a cero en algún instante de tiempo durante el periodo de bit [15].

#### Algoritmo de codificación

El dígito “1” binario es codificado mediante un nivel positivo y el “0” binario es codificado mediante un nivel cero. Además, la señal codificada es una señal con transiciones entre dos niveles que dependen del bit codificado [19]:

1: Nivel positivo con una transición durante el periodo de bit.

0: Nivel cero, sin transiciones durante el período de bit.

En este caso el valor de  $a_i$  toma los mismos valores que para UNRZ.

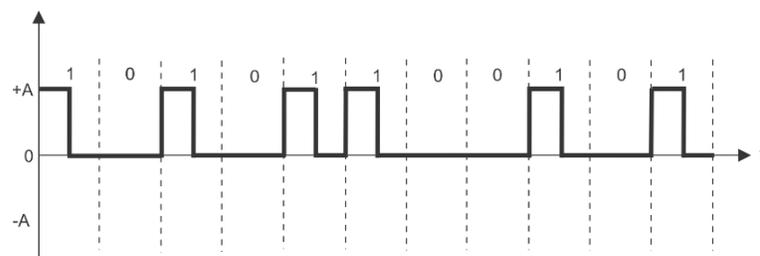


Figura 1.16. Codificación URZ en el dominio del tiempo.

Una ventaja de esta codificación es que le proporciona al receptor la capacidad de detectar la velocidad de bits a la que se transmiten los datos, permitiendo establecer un

<sup>8</sup> Es el rango de frecuencia por debajo del primer cruce por cero de la curva que describe la PSD.



sincronismo en el destino para temporizar apropiadamente el tiempo de muestreo de la señal recibida [15].

El sincronismo es posible gracias a la característica dada por el retorno a cero de la señal codificada en la mitad del período de bit como se ve en la figura 1.16 y se considera de igual manera para todos los códigos de línea del tipo retorno a cero<sup>9</sup>, aunque como es de esperarse, el ancho de banda de la señal transmitida es mayor con respecto a los códigos sin retorno a cero [17].

Para el caso del código URZ es posible establecer y mantener sincronismo entre el transmisor y el receptor para una secuencia larga de “1”s, aunque para una secuencia larga de “0”s puede perderse y finalmente decodificar erróneamente los símbolos recibidos [17].

### **Espectro de Potencia Teórico**

La función conformadora de los pulsos  $h(t)$  para un código de tipo retorno a cero está descrita por la función rectangular:

$$h(t) = \Pi\left(\frac{t-T_b/4}{T_b/2}\right). \quad (1.14)$$

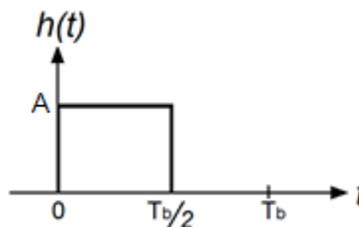


Figura 1.17. Forma de pulso para el código URZ.

A partir de la ecuación 1.10 la potencia media de la señal URZ es:

$$P_{SURZ} = \frac{A^2}{4}, \quad (1.15)$$

<sup>9</sup> URZ, PRZ, AMI-RZ.

donde  $A$  debe ser igual a 2 para normalizar la potencia. Por su parte la PSD es representada mediante:

$$S_{SURZ}(f) = \frac{A^2 T_b}{16} \text{sinc}^2 \left( \frac{T_b}{2} f \right) \left( 1 + \frac{1}{T_b} \sum_k \delta \left( f - \frac{k}{T_b} \right) \right). \quad (1.16).$$

La figura 1.18 describe la PSD de la señal URZ dada por la ecuación 1.16.

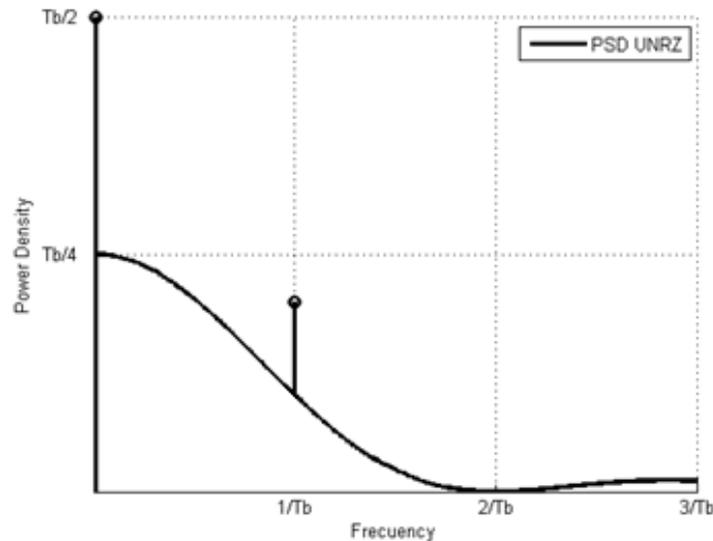


Figura 1.18. Espectro de potencia código URZ.

De igual manera que para el código UNRZ, también se presenta una componente de corriente continua muy alta. Aunque cabe destacar que en este caso la señal codificada requiere del doble de ancho de banda efectivo de la señal de datos debido a la reducción del pulso a la mitad [6]. Así su eficiencia espectral es:

$$\eta_{URZ} = \frac{1/T_b}{2/T_b} = 0.5 \frac{\text{bps}}{\text{Hz}}. \quad (1.17)$$

### 1.6.3. No Retorno a Cero Polar (PNRZ)

La señal codificada es representada mediante niveles de tensión tanto positivos como negativos [6], [17].



### Algoritmo de codificación

Este código presenta una regla de codificación sencilla, un dígito binario “1” es codificado mediante un nivel de tensión positivo y para un “0” binario se asigna un nivel negativo [17]. Cabe destacar que no existen transiciones o cambios de tensión durante el período de bit.

1: Nivel de tensión positivo, sin transiciones durante el período de bit.

0: Nivel de tensión negativo, sin transiciones durante el periodo de bit.

Por ser un código de tipo polar el valor de  $a_i$  toma los siguientes valores:

$$a_i = \begin{cases} -A & b_i = 0 \\ A & b_i = 1 \end{cases} \quad (1.18)$$

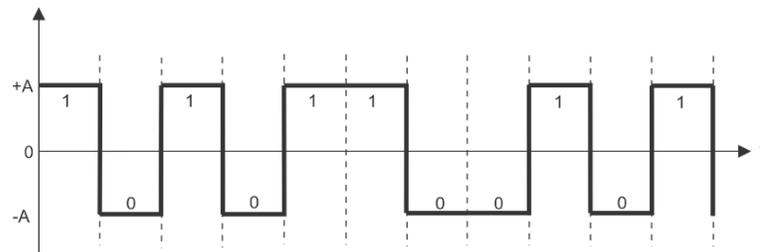


Figura 1.19. Codificación PNRZ en el dominio del tiempo.

Como es notable en la figura 1.19 la distancia entre los niveles de tensión codificados es mayor que para los códigos unipolares, esto aumenta la probabilidad de decodificar correctamente la señal transmitida en el receptor ya que la señal utiliza el doble de energía que utiliza la codificación UNRZ debido a la transmisión de pulsos para ambos dígitos binarios [6], [15]. Además, esta codificación no tiene transiciones entre el intervalo de duración del bit, esto significa que requiere menos ancho de banda que los códigos de tipo retorno a cero, sin embargo, una secuencia larga de “1”s o “0”s puede ocasionar una pérdida total de sincronización en el receptor [17]. Por otra parte, el proceso de codificación requiere el uso de dos fuentes de tensión para generar dos niveles que codifican los bits, lo que implica mayor costo en su implementación y un uso adicional de energía en relación al código UNRZ.

## Espectro de Potencia Teórico

En este caso la forma de pulso  $h(t)$  es la misma que para UNRZ (Ec.1.9), la diferencia radica en el formato de señalización descrito anteriormente.

La potencia media de la señal PNRZ a partir de la ecuación 1.10 es:

$$P_{SPNRZ} = A^2, \quad (1.19)$$

donde  $A$  debe ser igual a 1 para normalizar la potencia y la ecuación que describe la PSD para este código es:

$$S_{SPNRZ}(f) = A^2 T_b \text{sinc}^2(T_b f). \quad (1.20)$$

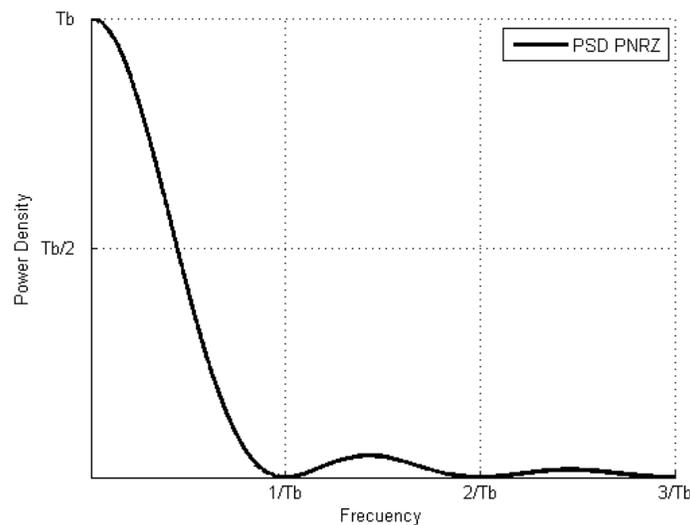


Figura 1.20. Espectro de potencia del código PNRZ.

Según la figura 1.20, la señal PNRZ ocupa el mismo ancho de banda que una señal UNRZ. Aunque en este código no se observa la presencia de impulsos, se presenta una alta componente de DC que como se había mencionado anteriormente involucra un inconveniente para ciertos sistemas. Además, es notable que la energía utilizada por la señal es mayor que para la codificación UNRZ.



La eficiencia espectral de este código es:

$$\eta_{PNRZ} = \frac{1/T_b}{1/T_b} = 1 \frac{bps}{Hz}. \quad (1.21)$$

#### 1.6.4. Retorno a Cero Polar (PRZ)

Es conocido también como un código ternario, ya que tiene tres niveles de tensión distintos, un nivel positivo, uno cero y un negativo, utilizados para codificar los bits de información a transmitir [15]. El código es de tipo retorno a cero.

##### Algoritmo de codificación

El dígito “1” binario es codificado mediante un nivel positivo y el “0” binario es codificado mediante un nivel negativo de tensión, tal como se ve en la figura 1.21.

1: Nivel positivo con una transición durante el periodo de bit.

0: Nivel negativo con una transición durante el periodo de bit.

En este caso el valor de  $a_i$  toma los mismos valores que para PNRZ.

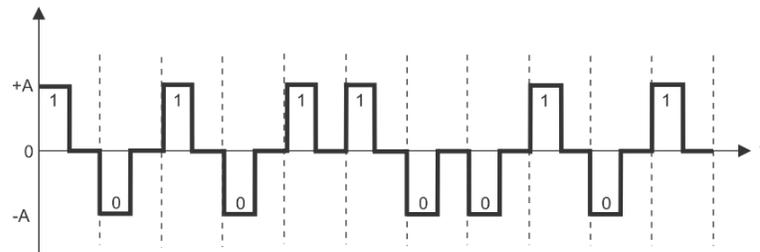


Figura 1.21. Codificación PRZ en el dominio del tiempo.

Una ventaja de esta codificación es que se puede mantener la sincronía en el reloj de muestreo del receptor, independientemente del símbolo transmitido debido a la presencia de transiciones para cualquier dígito [6], [17].

##### Espectro de Potencia Teórico

En este caso  $h(t)$  describe la misma forma de pulso que para URZ (ec. 1.14).

Empleando la ecuación 1.10 la potencia media de la señal PRZ es:

$$P_{SPRZ} = \frac{A^2}{2}, \quad (1.22)$$

donde  $A$  debe ser igual a  $\sqrt{2}$  para normalizar la potencia, y la PSD es:

$$S_{SPRZ}(f) = \frac{A^2 T_b}{4} \text{sinc}^2\left(\frac{T_b f}{2}\right). \quad (1.23)$$

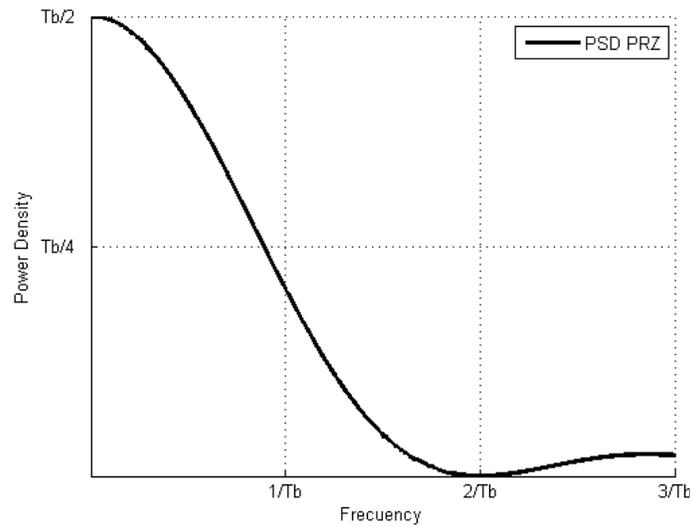


Figura 1.22. Espectro de potencia código PRZ.

El análisis es similar que para el código PNRZ, aunque cabe destacar que la energía utilizada por la señal es menor porque su ancho de banda se ha duplicado con respecto al código UNRZ como se visualiza en la figura 1.22. Si bien esta codificación tiene la capacidad de mantener la sincronización entre transmisor y receptor aún se siguen presentando componentes de corriente continua.

La eficiencia espectral de este código es:

$$\eta_{PRZ} = \frac{1/T_b}{2/T_b} = 0.5 \frac{\text{bps}}{\text{Hz}}. \quad (1.24)$$

### 1.6.5. No Retorno a Cero Invertido (NRZI)

La codificación NRZI también conocida como codificación NRZ tipo M y considerada como una codificación diferencial, es una variante de la codificación NRZ que no presenta transiciones entre el intervalo de duración de cada bit [15].

#### Algoritmo de codificación

El dígito binario “1” se codifica con un nivel de tensión opuesto en polaridad y de igual magnitud que el nivel de tensión codificado para el dígito binario anterior. Mientras que el “0” binario no representa cambio en el nivel de tensión con respecto al estado asignado para el bit precedente [6], [15].

0: No hay transiciones durante el periodo de bit y no hay cambio en el nivel de la señal con respecto al estado del bit anterior.

1: Hay una transición al comienzo del periodo de bit, dependiendo del nivel de tensión asignado al bit precedente.

Por ser un código de tipo diferencial el valor de  $a_i$  toma los siguientes valores:

$$a_i = \begin{cases} \pm A & b_i = 0 \\ \pm A & b_i = 1 \end{cases} \quad (1.25)$$

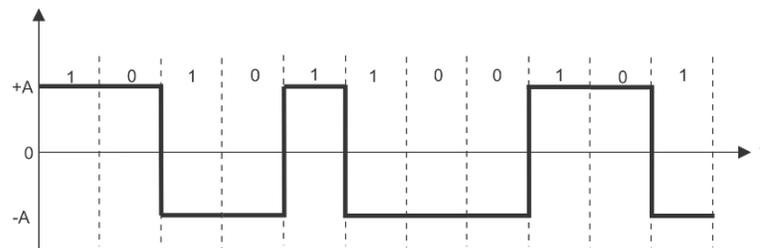


Figura 1.23. Codificación NRZI en el dominio del tiempo.

Como puede apreciarse en la figura 1.23 el código NRZI es un tipo de codificación polar, de esta manera su análisis es similar que para el código PNRZ. La ventaja de esta codificación es que el receptor puede decodificar la señal mediante la detección de transiciones entre los niveles de tensión recibidos, en vez de comparar muestras de la señal con un umbral predefinido como lo hacen los códigos anteriores [15].

## Espectro de Potencia Teórico

La forma de pulso  $h(t)$  es la misma que para UNRZ (Ec.1.9).

La potencia media de la señal NRZI dada por la ecuación 1.10 es:

$$P_{S_{NRZI}} = A^2, \quad (1.26)$$

donde  $A$  debe ser igual a 1 para normalizar la potencia y la función densidad espectral de potencia, dada por la ecuación 1.27, se obtiene de forma similar que para la codificación PNRZ [15] y se ilustra en la figura 1.24.

$$S_{S_{NRZI}} = A^2 T_b \text{sinc}^2(T_b f). \quad (1.27).$$

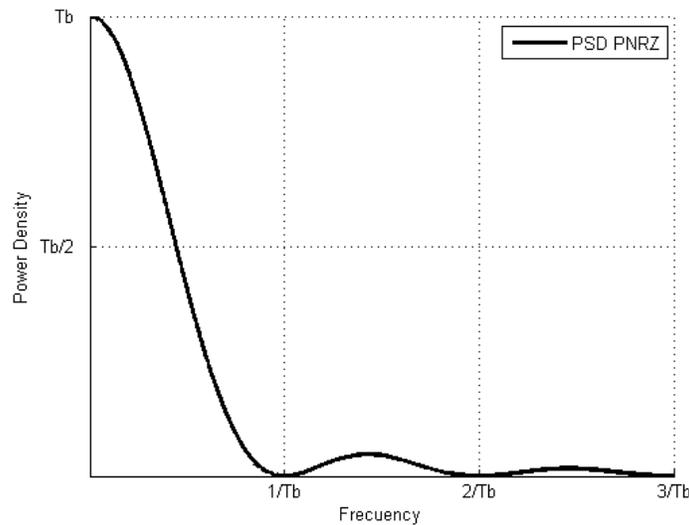


Figura 1.24. Espectro de potencia código NRZI.

La eficiencia espectral de este código es:

$$\eta_{NRZI} = \frac{1/T_b}{1/T_b} = 1 \frac{\text{bps}}{\text{Hz}}. \quad (1.28)$$

### 1.6.6. Manchester o Bifase L

El código Manchester se caracteriza por presentar una transición desde un nivel positivo hasta un nivel negativo o viceversa, justo a la mitad del período de bit [6].



### Algoritmo de codificación

Los dígitos binarios (“1” o “0”) se codifican mediante dos pulsos de polaridad invertida, manteniendo transiciones en los niveles de la señal durante toda la transmisión:

0: Transición desde un nivel negativo a uno positivo en la mitad del período de bit.

1: Transición desde un nivel positivo a uno negativo en la mitad del período de bit.

En la tabla 1.2 se visualizan las posibles transiciones de esta codificación.

Tabla 1.2. Transiciones de la codificación Manchester.

Digito Binario	Transición previa
1	
0	

En este caso el valor de  $a_i$  toma los mismos valores que para PNRZ.

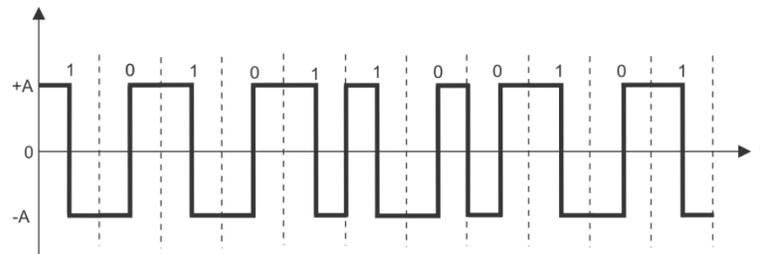


Figura 1.25. Codificación Manchester.

Este código puede establecer una sincronización entre el emisor y receptor por sus transiciones a mitad del período de bit como se ve en la figura 1.25, independientemente del bit transmitido [17].

### Espectro de Potencia Teórico

La función  $h(t)$  que describe la forma del pulso para el código Manchester está compuesta por la suma de dos funciones rectangulares:

$$h(t) = \Pi\left(\frac{t-T_b/4}{T_b/2}\right) - \Pi\left(\frac{t-3T_b/4}{T_b/2}\right). \quad (1.29).$$

La representación gráfica se muestra en la figura 1.26.

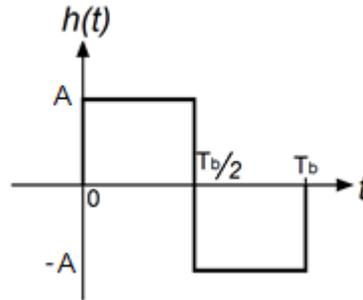


Figura 1.26. Forma de onda para la codificación Manchester.

A partir de la ecuación 1.10 la potencia media del código Manchester es:

$$P_{S_{Manchester}} = A^2, \quad (1.30)$$

donde  $A$  debe ser igual a 1 para normalizar la potencia y la función de la PSD para este código es:

$$S_{S_{Manchester}}(f) = A^2 T_b \text{sinc}^2\left(\frac{T_b f}{2}\right) \text{sen}^2\left(\frac{\pi T_b f}{2}\right). \quad (1.31).$$

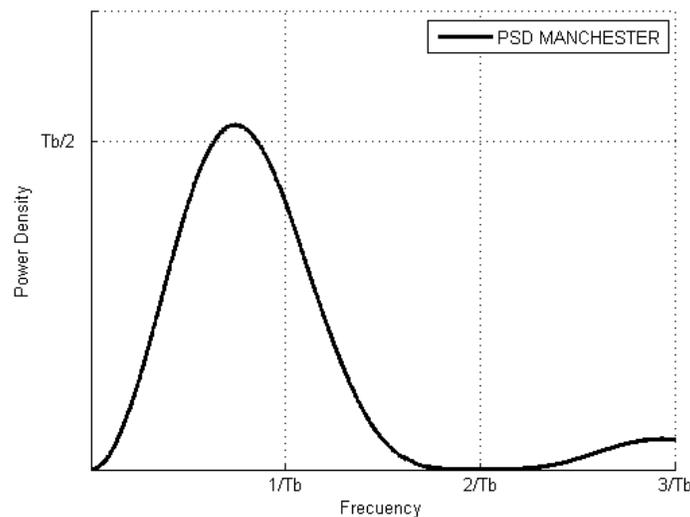


Figura 1.27. Espectro de potencia código Manchester.



A pesar de que esta codificación utiliza el doble de ancho de banda con respecto al código UNRZ, se observa a partir de la figura 1.27 que ya no existe una componente de corriente continua, esto es gracias a las transiciones que se presentan durante toda la transmisión. El formato de señalización para esta codificación facilita al receptor poder establecer una buena sincronización con respecto al transmisor, lo cual es importante para decodificar correctamente la señal recibida [6], [15].

La eficiencia espectral de este código es:

$$\eta_{Manchester} = \frac{1/T_b}{2/T_b} = 0.5 \frac{bps}{Hz}. \quad (1.32)$$

### 1.6.7. Código Manchester Diferencial

Este código es una variante de la codificación Manchester, manteniendo las transiciones a la mitad del periodo de bit. Además, como su nombre lo indica, se trata de un tipo de codificación diferencial que adopta las características de señalización del código NRZI [6], [15], [19].

#### Algoritmo de codificación

Teniendo en cuenta el algoritmo de señalización para el código NRZI, en donde los niveles de tensión solo se invierten al codificar un “1” binario, el código Manchester Diferencial invierte la polaridad de las transiciones de la siguiente manera:

- 0: Transición a la mitad del período de bit, con igual polaridad que la transición previa.
- 1: Transición a la mitad del período de bit, con polaridad opuesta a la transición previa.

La tabla 1.3 muestra las transiciones que toma la señal acorde al bit a codificar.

Tabla 1.3. Transiciones codificación Manchester Diferencial.

Bit a codificar	Transición previa	Transición de bit codificado
1		
		
0		
		

En este caso el valor de  $a_i$  toma los mismos valores que para NRZI.

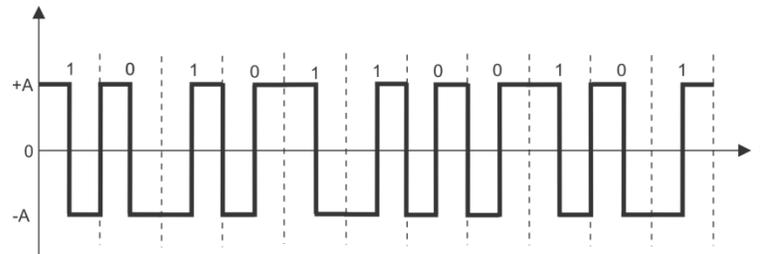


Figura 1.28. Codificación Manchester Diferencial.

Esta codificación, presentada en la figura 1.28, posee las mismas características que el código Manchester o Bifase L, pero al igual que el código NRZI también es posible decodificar la señal recibida teniendo en cuenta las transiciones de los niveles de tensión.

### Espectro de Potencia Teórico

Para este código la forma de pulso  $h(t)$  es igual que para la codificación Manchester.

Empleando la ecuación 1.10 la potencia media de la señal Manchester Diferencial es:

$$P_{S_{Manchester\ Dif.}} = A^2, \quad (1.33)$$

donde  $A$  debe ser igual a 1 para normalizar la potencia y su densidad espectral de potencia es igual que para la codificación Manchester (ec. 1.31).

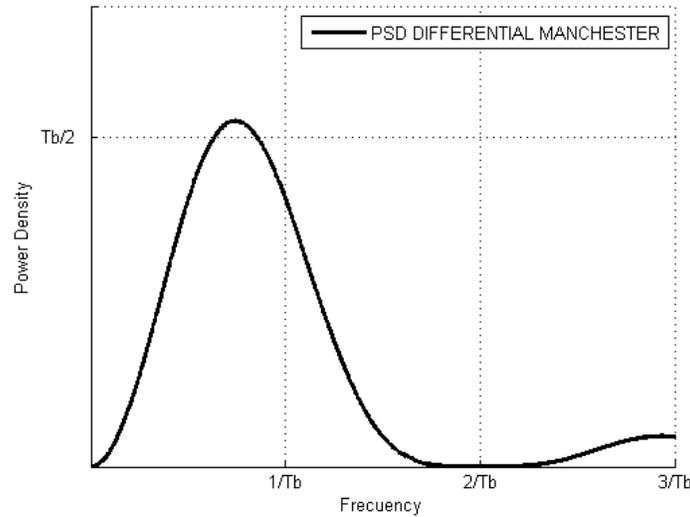


Figura 1.29. Espectro de potencia código Manchester Diferencial.

Como se puede apreciar en la figura 1.29, éste código presenta una componente DC nula al igual que el código Manchester y el ancho de banda requerido por la señal codificada es el doble del de los datos, similar a todos los códigos que presentan una transición a la mitad de tiempo de bit.

La eficiencia espectral de este código es:

$$\eta_{Manchester\ Dif.} = \frac{1/T_b}{2/T_b} = 0.5 \frac{bps}{Hz}. \quad (1.34)$$

### 1.6.8. Inversión de Marcas Alternadas sin Retorno a Cero (AMI-NRZ)

Es un tipo de codificación bipolar en donde los niveles de tensión se alternan solo para un dígito binario ("1" o "0"). Además, no presenta transiciones durante el período de bit de la señal.



### Algoritmo de codificación

Este código asigna alternadamente niveles de tensión positivos y negativos para un “1” binario, mientras que un “0” binario se codifica con nivel cero, esto se puede observar en la figura 1.30.

0: Nivel cero sin transiciones durante el periodo de bit.

1: Niveles de tensión alternados sin transiciones durante el periodo de bit.

Por ser un código de tipo bipolar el valor de  $a_i$  toma los siguientes valores:

$$a_i = \begin{cases} 0 & b_i = 0 \\ \pm A & b_i = 1 \end{cases} \quad (1.35)$$

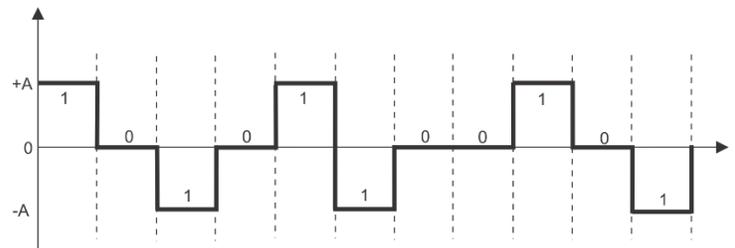


Figura 1.30. Codificación AMI-NRZ.

En caso tal de que el receptor pierda sincronía con el transmisor, su recuperación es posible, basta con rectificar la onda recibida a niveles positivos y detectar las transiciones dadas por los “1”s binarios.

### Espectro de Potencia Teórico

Ya que es un código sin retorno a cero, la función conformadora del pulso es similar a la codificación UNRZ (Ec.1.9).

La potencia media de la señal AMI-NRZ es:

$$P_{S_{AMI-NRZ}} = \frac{A^2}{2}, \quad (1.36)$$

donde  $A$  debe ser igual a  $\sqrt{2}$  para normalizar la potencia y la función de la PSD es:

$$S_{S_{AMI-NRZ}} = A^2 T_b \text{sinc}^2(T_b f) \text{sen}^2(\pi T_b f). \quad (1.37)$$

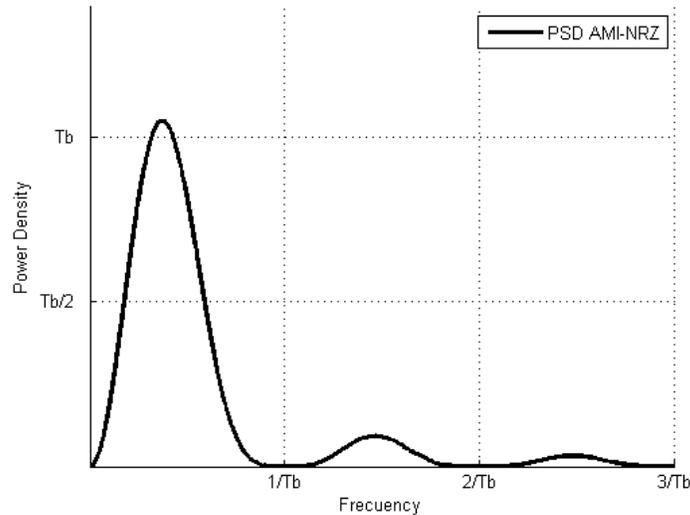


Figura 1.31. Espectro de potencia código AMI-NRZ.

Por ser un código bipolar, la componente DC se aproxima a cero como se visualiza en la figura 1.31. Además, la eficiencia espectral de este código es:

$$\eta_{AMI-NRZ} = \frac{1/T_b}{1/T_b} = 1 \frac{\text{bps}}{\text{Hz}}. \quad (1.38)$$

### 1.6.9. Inversión de Marcas Alternadas con Retorno a Cero (AMI-RZ)

También es un tipo de codificación bipolar en donde los niveles de tensión se alternan solo para un dígito binario (“1” o “0”), pero en este caso se presenta una transición durante el período de bit de la señal.

#### Algoritmo de codificación

Este código asigna alternadamente niveles de tensión positivos y negativos para un “1” binario, mientras que un “0” binario se codifica con nivel cero. El resultado de este proceso se observa en la figura 1.32.

0: Nivel cero sin transiciones durante el periodo de bit.

1: Niveles de tensión alternados con una transición durante el periodo de bit.

Por ser un código de tipo bipolar el valor de  $a_i$  toma los siguientes valores:

$$a_i = \begin{cases} 0 & b_i = 0 \\ \pm A & b_i = 1 \end{cases} \quad (1.39)$$

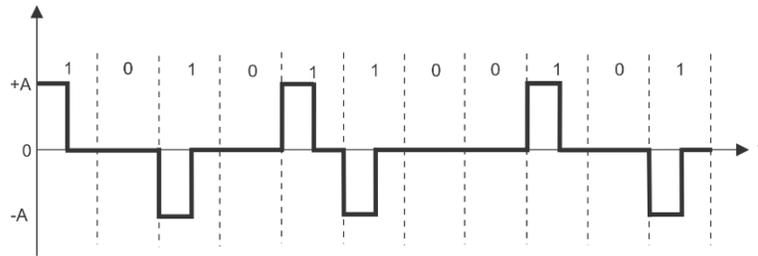


Figura 1.32. Codificación AMI-RZ.

En esta codificación también es posible recuperar la sincronía con el transmisor en caso tal de que ésta se pierda con el receptor, solo se rectifica la onda recibida a niveles positivos y se detectan las transiciones dadas por los “1”s binarios.

### Espectro de Potencia Teórico

Este código junto con la codificación URZ, comparte la misma función conformadora de pulso (Ec.1.14).

Teniendo en cuenta la ecuación 1.10 la potencia media de la señal AMI-RZ es:

$$P_{S_{AMI-RZ}} = \frac{A^2}{4}, \quad (1.40)$$

siendo  $A$  igual a 2 para normalizar la potencia. Por su parte la función que describe la PSD de este código es:

$$S_{S_{AMI-RZ}} = \frac{A^2}{4} T_b \text{sinc}^2 \left( \frac{T_b f}{2} \right) \text{sen}^2(\pi T_b f), \quad (1.41)$$

y su representación gráfica se ilustra en la figura 1.33.

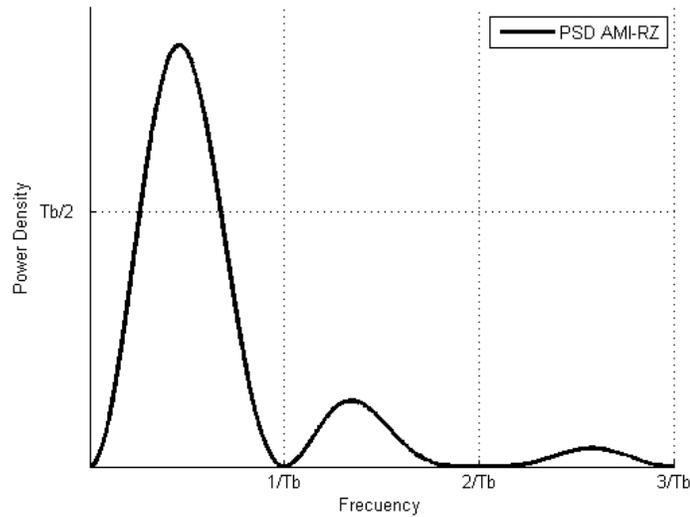


Figura 1.33. Espectro de potencia código AMI-RZ.

La eficiencia espectral de este código es:

$$\eta_{AMI-RZ} = \frac{1/T_b}{1/T_b} = 1 \frac{bps}{Hz}. \quad (1.42)$$

La figura 1.34, expone en conjunto la representación gráfica de la PSD para todos los códigos de línea implementados en el presente trabajo de grado, en donde se puede apreciar que las áreas bajo las curvas de cada PSD son iguales<sup>10</sup>, esto en relación a la normalización de potencia.

<sup>10</sup> Teniendo en cuenta que la potencia media de una la señal se calcula mediante la integral de su densidad espectral de potencia entre  $-\infty$  e  $\infty$  [25].

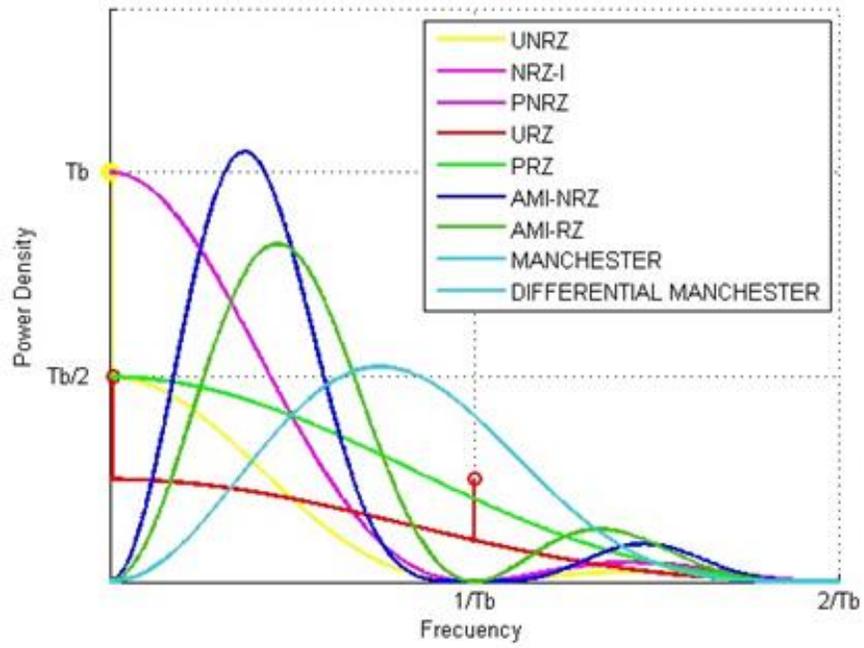


Figura 1.34. Comparación de la PSD para los diferentes códigos de línea.





## CAPÍTULO 2

### MODELADO, SIMULACIÓN E IMPLEMENTACIÓN

Con el objetivo de presentar el modelo del sistema de comunicación digital en banda base con codificación de línea sobre hardware reconfigurable, este capítulo expone las actividades de modelado, simulación e implementación.

#### 2.1. METODOLOGÍA DE TRABAJO

Se emplea la Metodología Universal de Diseño (UDM, *Universal Design Methodology*) de Bod Zeidman [39]. La cual está basada en el diseño hardware de forma eficiente y en el menor tiempo posible. Para ello se tienen en cuenta las fases relacionadas en la figura 2.1.



Figura 2.1. Metodología de trabajo.

Las anteriores fases están distribuidas en las secciones 2.2, 2.3, 2.4, 2.5 y 2.6 en las cuales se profundiza sobre el tema.



## 2.2. FASE 0: RECOLECCIÓN DE INFORMACIÓN Y ESPECIFICACIONES.

En esta sección se describen los requisitos que deben cumplir tanto los elementos software como hardware para la implementación del sistema de comunicación, teniendo en cuenta los objetivos del proyecto de grado.

### 2.2.1.Requisitos de los elementos software y hardware

Las herramientas de desarrollo deben permitir:

- Implementar las técnicas de codificación de línea UNRZ, URZ, PNRZ, PRZ, NRZI, Manchester, Manchester Diferencial, AMI-NRZ Y AMI-RZ empleando un modelo de canal con Ruido Aditivo Blanco Gaussiano (AWGN, *Additive White Gaussian Noise*).
- Identificar y analizar las características y parámetros de desempeño de las codificaciones de línea.
- Evaluar el desempeño a nivel de la tasa de error de bit (BER, *Bit Error Rate*) y de eficiencia espectral en un sistema de comunicaciones en banda base que implementa la codificación.

### 2.2.2.Búsqueda de herramientas de desarrollo

En la actualidad existen una gran cantidad de herramientas software y hardware para desarrollar este tipo de trabajos:

- **Hardware**

**RASPBERRY PI:** Es una pequeña placa de desarrollo con grandes prestaciones que se utiliza en proyectos de electrónica y posee funcionalidades similares a las de un computador de escritorio [29]. En el mercado existen dos modelos de esta placa, el A y el B, este último tiene un chip integrado *Broadcom BCM2835*, que contiene un



procesador *ARM11* con varias frecuencias de funcionamiento, un procesador gráfico *VideoCore IV*, y distintas capacidades de memoria de acceso aleatorio (RAM, *Random Access Memory*) [30].

Con el paquete de soporte de *Matlab*, se puede establecer una comunicación con la *Raspberry Pi* a través de *Ethernet* o *Wi-Fi* para controlar los dispositivos periféricos, permitiendo adquirir datos para luego visualizarlos y analizarlos en *Matlab* [31].

**ARDUINO:** Es una plataforma de prototipos electrónica, de código abierto (*open-source*), basada en hardware y software flexibles los cuales son fáciles de usar. *Arduino* puede sentir el entorno mediante la recepción de datos desde una variedad de sensores y puede afectar su alrededor mediante el control de luces, motores y otros artefactos [27].

*Matlab* ofrece un paquete de soporte para el hardware *Arduino*, el cual le permite comunicarse con esta placa a través del Bus Serial Universal (USB, *Universal Serial Bus*). También realizar tareas tales como: Adquirir datos de los sensores analógicos y digitales y controlar otros dispositivos. Debido a que *Matlab* es un lenguaje de alto nivel, se puede ver el resultado de las instrucciones de entrada y salida en tiempo real [28].

**FPGA:** Los Arreglos de Compuertas Programables en Campo (FPGA, *Field Programmable Gate Array*) son chips de silicio reprogramables. Al utilizar bloques de lógica pre-construidos y recursos para ruteo programables, estos chips se pueden configurar para implementar funcionalidades personalizadas en hardware, mediante el desarrollo de tareas de cómputo digital en software que luego son compiladas en un archivo de configuración, el cual establece la conexión de los componentes internos [26].

La adopción de chips FPGA en la industria ha sido impulsada por el hecho de que combinan lo mejor de los Circuitos Integrados de Aplicación Específica (ASIC, *Application-Specific Integrated Circuit*) y de los sistemas basados en procesadores. Además, ofrecen velocidades temporizadas por hardware y fiabilidad, pero sin requerir



altos volúmenes de recursos para compensar el gran gasto que genera un diseño personalizado de ASIC. Los chips de silicio reprogramables tienen la misma capacidad de ajustarse que un software que se ejecuta en un sistema basado en procesadores, pero no está limitado por el número de núcleos disponibles. A diferencia de los procesadores, los FPGAs llevan a cabo diferentes operaciones de manera paralela, por lo que éstas no necesitan competir por los mismos recursos. Cada proceso independiente se asigna a una sección dedicada del chip y puede ejecutarse de manera autónoma sin ser afectada por otros bloques de lógica; como resultado, el rendimiento de una parte de la aplicación no se ve afectada cuando se agregan otros procesos [26].

**USRP** (*Universal Software Radio Peripheral*): es un dispositivo de bajo costo que trabaja en conjunto con un computador a través de un FPGA para la realización de radio software. Este periférico realiza las funciones de llevar la señal banda base de Radio Frecuencia (RF, *Radio Frecuency*), a través de una sección de Frecuencia Intermedia (IF, *Intermediate Frecuency*) y viceversa. Es un diseño totalmente libre que utiliza como software Radio GNU.

En Radio GNU, el código para el procesamiento de la señal está escrito en C++, utilizando herramientas de software libre como la Colección de Compiladores de GNU (GCC, *GNU Compiler Collection*). El control y la Interfaz Gráfica de Usuario (GUI, *Graphical User Interface*) están escritos en código Python, que también es software libre. Al igual que el Firmware para el controlador USB está escrito en C y compilado con el pequeño dispositivo compilador de C (SDCC, *Small Device C Compiler*). El código de la FPGA del USRP está escrito en Verilog, el cual también se puede combinar con el Lenguaje de Descripción Hardware para un Circuito Integrado de Muy Alta Velocidad (VHSIC, *Very High Speed Integrated Circuit*) (VHDL, *VHSIC Hardware Description Language*) si se desea. Y el software libre de Altera, *Quartus II* se utiliza para la síntesis del código [32].

- **Software**

**OCTAVE:** Es un software libre, que emplea un lenguaje de programación de alto nivel destinado para el cálculo numérico. Tiene una interfaz sencilla, orientada a la



utilización de líneas de comandos en una consola. La gran cantidad de herramientas que posee permite resolver problemas de álgebra lineal, cálculo de raíces, de ecuaciones no lineales, integración de funciones ordinarias, manipulación de polinomios, integración de ecuaciones diferenciales ordinarias y ecuaciones diferenciales algebraicas. Además, sus funciones se pueden extender mediante módulos dinámicamente escritos en el lenguaje propio de *Octave* o escritos en lenguajes como C, C++ y *Fortran* entre otros, presentando así gran compatibilidad con programas similares como *Matlab* y *Scilab* [33].

**SCILAB:** Es un software libre y de código abierto utilizado para el cálculo numérico, el cual proporciona un entorno de computación de gran alcance para aplicaciones científicas y de ingeniería. Incluye cientos de funciones matemáticas. Tiene un lenguaje de programación de alto nivel que permite el acceso a estructuras de datos avanzadas y herramientas que facilitan la visualización de funciones en *2D* y *3D*, el análisis estadístico de datos y el procesamiento de señales y filtraje en los dominios de tiempo y frecuencia. Este software es similar a *Matlab* con una gran cantidad de aplicaciones [34].

**MATLAB:** Es un lenguaje de alto nivel que provee un entorno interactivo utilizado por millones de ingenieros y científicos en todo el mundo. Permite explorar y visualizar ideas, así como colaborar interdisciplinariamente en procesamiento de señales e imagen, comunicaciones, sistemas de control y finanzas computacionales. Este software matemático ofrece un entorno de desarrollo integrado con un lenguaje de programación propio (Lenguaje M), que permite crear interfaces de usuario para automatizar procesos de cálculo y facilitar la comunicación con programas de diferentes lenguajes y dispositivos hardware, cuenta con *Simulink*®, una herramienta que tiene bloques prediseñados que facilitan la construcción de sistemas de comunicación y su respectivo análisis [35].

**DSP Builder:** Los Procesadores Digitales de Señal (DSP, *Digital Signal Processing*) de Altera integran dos herramientas: la primera, de alto nivel para el desarrollo de algoritmos y la segunda, el Lenguaje de Descripción Hardware (HDL, *Hardware Description Language*) a través de la combinación del desarrollo de algoritmos,



simulación y verificación de las capacidades de las herramientas de diseño de *Mathworks*.

Consta de un conjunto de bloques de construcción que acortan los ciclos de diseño DSP, ayudando a crear la representación hardware en un entorno de desarrollo amigable. Permite utilizar las funciones de *Matlab* y los bloques de *Simulink*® existentes como parte del proceso de diseño y prueba, permitiendo evaluar rápidamente el desempeño de los diseños en dispositivos de Altera, generar código HDL optimizado en tiempo real y verificar las implementaciones hardware con modelos de *Simulink*®.

Con todo lo anterior, *DSP Builder* ofrece bloques en *Simulink*® capaces de desarrollar tareas básicas como funciones aritméticas y de almacenamiento, así como funciones complejas de corrección de errores y filtrado. [36]

**SYSTEM GENERATOR XILINXS (XGS):** Es una herramienta de diseño de sistemas de alto nivel que permite el desarrollo y verificación de algoritmos DSP, optimizada para FPGAs *Xilinx*. *System Generator* funciona dentro del modelado y simulación del ambiente de *Simulink*®. Su principal ventaja radica en la posibilidad de combinar bloques funcionales de *Xilinx* con *Matlab* y *Simulink*® para crear un banco de pruebas y analizar así los datos generados por el modelo. El alto nivel de abstracción proporcionado por *System Generator* simplifica en gran medida el desarrollo de algoritmos y su verificación incluso en sistemas de comunicaciones multifrecuencia muy sofisticados.

Así, las librerías de *Xilinx* que incluyen bloques de comunicación, lógica de control, procesamiento de señales matemáticas y memoria, permiten integrar código HDL, funciones de *Matlab* y componentes de hardware diseñados para FPGAs de *Xilinx*, con el objetivo de crear modelos de sistemas de comunicaciones completos que se puedan simular en el entorno *Simulink*® [36].

De acuerdo a las características expuestas anteriormente sobre las herramientas software y hardware existentes y teniendo en cuenta los requisitos para el desarrollo del trabajo de grado, en la siguiente sección se dan a conocer las herramientas seleccionadas.

### 2.3. FASE 1: SELECCIÓN DE HERRAMIENTAS.

- **Hardware**

Se seleccionó la tarjeta de desarrollo *Stater Kit Board de Xilinx* de la familia *Spartan 3A*, basada en FPGA mostrada en la figura 2.2, disponible en la Universidad del Cauca. Cuenta con el dispositivo de la serie *XC3S700A*, el cual posee elementos lógicos para la implementación de sistemas de complejidad media, óptimos para la implementación de la codificación de línea.



Figura 2.2. FPGA *Spartan 3A XC3S700A*.

- **Software**

Software *Matlab*, el cual es utilizado para crear un modelo de referencia del sistema de comunicación digital desarrollado en *Simulink*<sup>®</sup>. Además, se utiliza el software de *Xilinx* que permite analizar y compilar diseños HDL, simular la reacción de un diseño a diferentes estímulos, configurar el hardware a través de una interfaz de programación y analizar los tiempos de simulación [37]. Del paquete de diseño integrado de *Xilinx* se utilizan los siguientes componentes:

**System Generator:** Es un Ambiente de Diseño Integrado (IDE, *Integrated development environment*), que hace uso de *Simulink*<sup>®</sup> como entorno de desarrollo. Tiene un flujo de diseño integrado, para pasar directo al archivo de configuración (\*.bit) necesario para la programación del FPGA. Una característica importante es la generación automática del



código VHDL y de un proyecto de Ambiente de Software Integrado (ISE, *Integrated Software Environment*) del modelo que se desarrolle. Los bloques en XGS operan con valores booleanos o valor arbitrario en punto fijo para dar una mejor aproximación a la implementación hardware. Sin embargo, *Simulink*® trabaja con números de punto flotante de doble precisión, por lo cual la conexión entre los bloques de *System Generator* y los de *Simulink*® se realizan mediante los bloques “gateway” [38].

**Project Navigator:** Proporciona un análisis de todas las restricciones de diseño y el cálculo de todos los recursos hardware requeridos para una posible implementación [38].

**Impact:** Realiza la configuración directamente de los FPGAs Xilinx o Dispositivo Lógico Programable Complejo (CPLD, *Complex Programmable Logic Device*) y Memoria Programable de Solo Lectura (PROM, *Programmable Read-only Memory*) con diferentes interconexiones físicas de Xilinx como el cable serial, cable USB entre otros [38].

Esta selección de tecnologías se hace teniendo en cuenta que la universidad cuenta con la licencia del software *Matlab* y la tarjeta de desarrollo *Spartan 3A* basada en FPGA.

## 2.4. FASE 2. DISEÑO DEL SISTEMA

En esta sección se propone conceptualmente la obtención de un modelo básico para el sistema de comunicaciones en banda base, con las diferentes codificaciones de línea y un canal AWGN.

### 2.4.1. Modelo de referencia

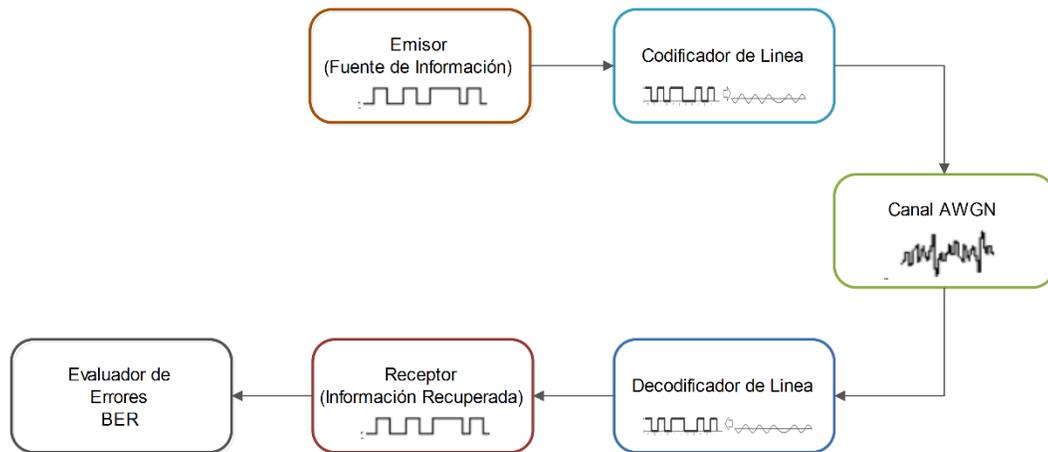


Figura 2.3. Modelo de referencia del sistema de comunicaciones.

De acuerdo a la figura 1.3, se elabora un modelo de referencia que cuenta con tres componentes y dos bloques adicionales mostrados en la figura 2.3: Emisor o fuente de información 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 BER.

### 2.4.2. Diseño del modelo de referencia

A partir del modelo de referencia de la figura 2.3, se diseñan uno a uno los bloques mostrados, teniendo en cuenta los diferentes sistemas de comunicación construidos con la herramienta *Simulink*® de *Matlab* haciendo uso de los paquetes de *Xilinx*.

#### 2.4.2.1. Emisor o Fuente de información

Se encarga de generar una secuencia de bits pseudoaleatoria con una velocidad de transmisión de datos determinada ( $R_b$ ). Los bits generados son equiprobables.



Figura 2.4. Bloque Fuente de información.

El bloque *Bits Generator* mostrado en la figura 2.3 implementa un Registro de Desplazamiento con Realimentación Lineal (LFSR, *Linear Feedback Shift Register*), compuesto por una cadena de biestables, el cual puede generar largas secuencias pseudoaleatorias de unos y ceros.

#### 2.4.2.2. Canal AWGN

Se asume un canal de comunicaciones no limitado en banda que adiciona ruido AWGN a la señal presente en la entrada *Signal* y la transfiere a la salida *Out* (Figura 2.5).

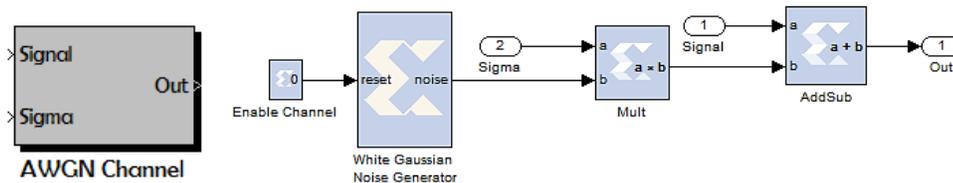


Figura 2.5. Bloque *AWGN Channel*.

En la figura 2.5 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<sup>11</sup> [23].

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 [23].

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

<sup>11</sup> Indica que la suma de n variables aleatorias independientes entre sí, corresponde a una variable aleatoria con distribución normal cuando la suma de estas variables es lo suficientemente grande.

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 ( $\sigma$ ). 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  $\sigma_x^2$  definida por la ecuación 2.1 [9].

$$\sigma_x^2 = \frac{N_0}{2}. \quad (2.1)$$

De esta manera, la señal de entrada *Sigma* se utiliza para controlar la potencia de ruido generada por el bloque de la figura 2.5. 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 ( $E_s/N_0$ ).

### 2.4.2.3. Evaluador de BER.

- **Sistema de Control de Errores**

Este bloque 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 transfieren a su salida los bits errados (*Wrong Bits*) sobre un total de bits transmitidos (*Total Bits*).

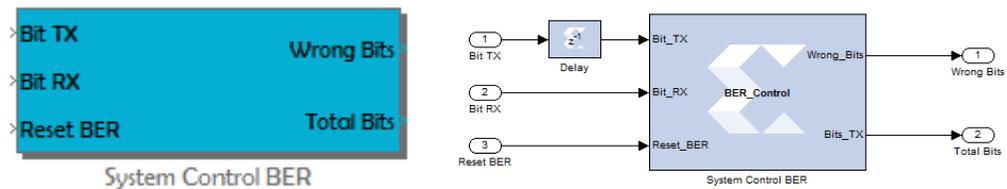


Figura 2.6. Bloque que registra la Tasa de Error de Bit.



El bloque *System Control BER* mostrado en la figura 2.6 realiza una comparación entre el bit transmitido y el bit recibido cada  $T_b$  instantes de tiempo. Si los valores leídos difieren, entonces la salida numérica *Wrong Bits* se incrementa en 1. Por su parte *Total Bits* indica el total de bits que se han transmitido. Además, la señal transmitida está desfasada con respecto a la señal recibida debido al retardo de transmisión y procesamiento de los datos, es por eso que se requiere del bloque *Delay* que adiciona el mismo retardo a la señal transmitida con el fin de que ambas señales (transmitida y recibida) estén correctamente sincronizadas para lograr una correcta comparación de cada bit transmitido.

- **Sistema de Control para la transmisión serial de la información procesada por la FPGA**

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.

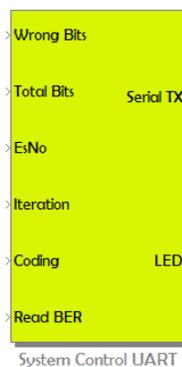


Figura 2.7. Bloque Control UART.

El subsistema visualizado en la figura 2.7 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 a partir de la información presente a las entradas del sub-bloque *Frame ASCII Encoder* mostrado en la figura 2.8, que brindan 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*) [24], 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.

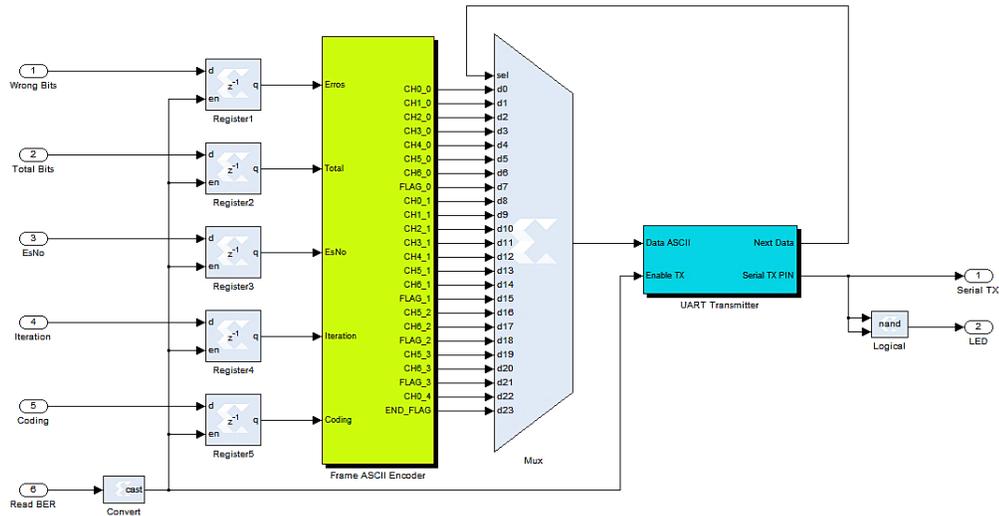


Figura 2.8. Estructura del bloque *System Control UART*.

La conformación de la trama se realiza mediante la conversión del dato presente en cada entrada del bloque *Frame ASCII Encoder* de la figura 2.8. Cada valor de tipo *double* se convierte en una cadena de caracteres de tipo *String*.

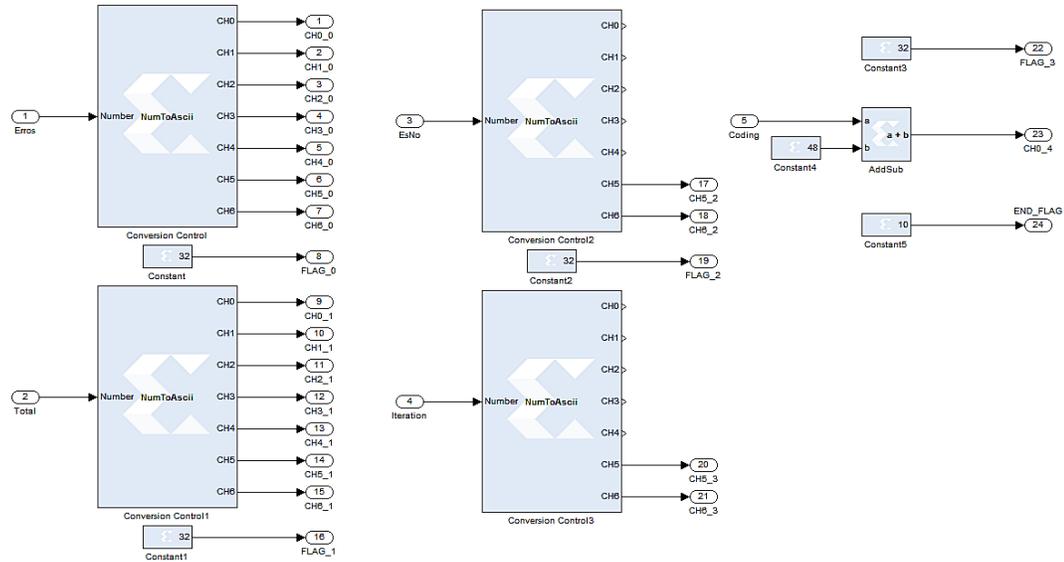


Figura 2.9. Estructura del bloque *Frame ASCII Encoder*.

En la figura 2.9 se observa la reutilización del bloque *Conversion Control*, cuya función es realizar la conversión del valor numérico de la entrada *Number* en 7 salidas (CH0... CH6) cada una de 8 bits de longitud, que representan caracteres alfanuméricos<sup>12</sup>. Además, se observan bloques con salidas constantes con valor decimal 32, alfanuméricamente correspondiente al carácter de espaciado que establece el inicio y fin de cada valor, por último, se establece una salida constante de valor decimal 10 que alfanuméricamente representa un salto de línea. De esta manera la trama puede visualizarse fácilmente en cualquier aplicación con conexión a un puerto serie.



Figura 2.10. Estructura de la trama de transmisión serial.

Una vez conformada la trama, con el formato visualizado en la figura 2.10, cada salida del bloque *Frame ASCII Encoder* se conecta al multiplexor *Mux* de la figura 2.8, el cual se encarga de serializar los bytes a una velocidad dada por la salida *Next Data* del subsistema *UART Transmitter*.

<sup>12</sup> Debido a que se tienen 7 salidas (CH0... CH6), el máximo valor numérico a decodificar es 9999999. Sin embargo, se trabajó con un valor máximo de 1048576 correspondiente al valor máximo de bits transmitidos.

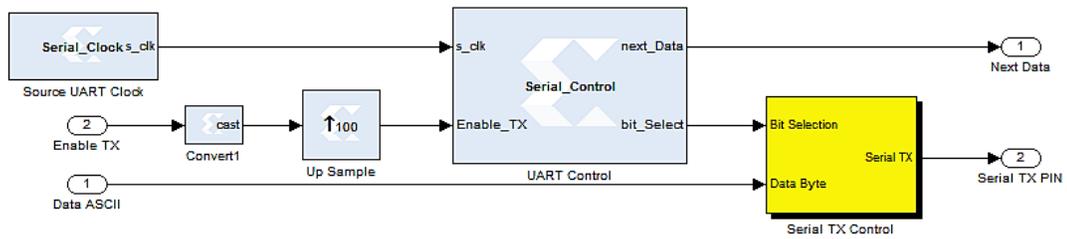


Figura 2.11. Estructura del subsistema *UART Transmitter*.

El bloque *Source UART Clock* de la figura 2.11 genera una señal de reloj a una velocidad aproximada de 9600bps, esto establece la velocidad de transmisión de los datos por el puerto serial (9600 baudios). Por otra parte, el bloque *UART Control*, conoce el total de bytes presentes en la trama, de esta manera la señal de salida *Next Data*, controla mediante el *Mux* de la figura 2.8, el byte presente en la entrada *Data ASCII* de la figura 2.11, el cual a su vez, debe ser serializado en bits que finalmente son los que se transmiten por el pin del puerto serial de la placa de desarrollo *Spartan 3A*. Este último proceso se lleva a cabo en el bloque *Serial TX Control* mostrado en la figura 2.11, que serializa cada byte mediante el uso del multiplexor *Mux* de la figura 2.12, que requiere de la señal de entrada *Bit Selection*, proporcionada por el bloque *UART Control*, para dividir el byte en señales binarias o bits a la salida *Serial TX Pin*.

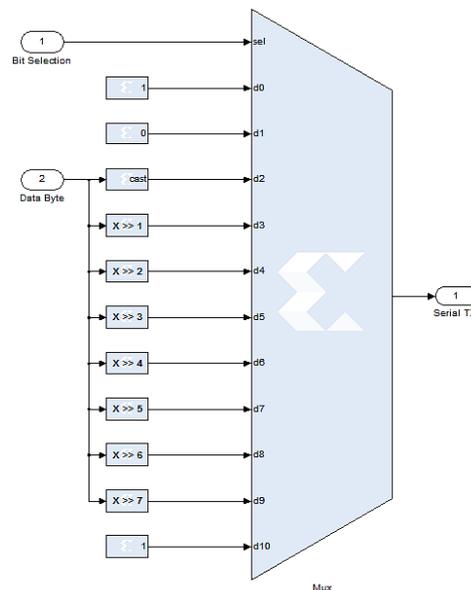


Figura 2.12. Estructura del bloque *Serial TX Control*.

- **Sistema de Control Principal**

Este módulo genera los valores de desviación estándar ( $\sigma$ ) 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  $E_s/N_0$ .

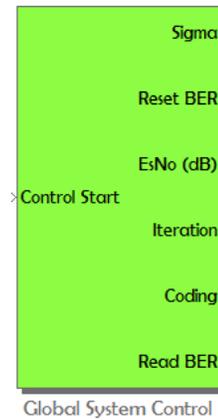


Figura 2.13. Sistema de Control Principal.

La entrada y las salidas del bloque *Global System Control* mostradas en la figura 2.13 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 2.6.
- **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.

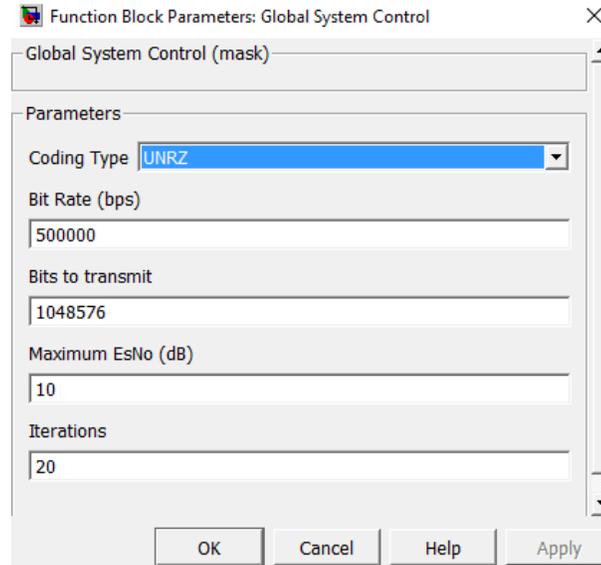


Figura 2.14. Configuraciones del bloque *Global System Control*.

Los parámetros de configuración del bloque mostrado en la figura 2.14 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.

El bloque *System Control* de la figura 2.15, incluye el algoritmo de programación que controla las operaciones de iteración, transmisión de cierta cantidad de bits y generación de la señal de lectura de la tasa de error de bit. Además, se observa un bloque adicional bajo el nombre de *Sigma Control*, este bloque, al igual que el anterior, está configurado con un archivo de programación que recibe como parámetro de entrada el valor de *EsNo* previamente configurado, para calcular a su salida el parámetro *Sigma* en un instante de tiempo determinado.

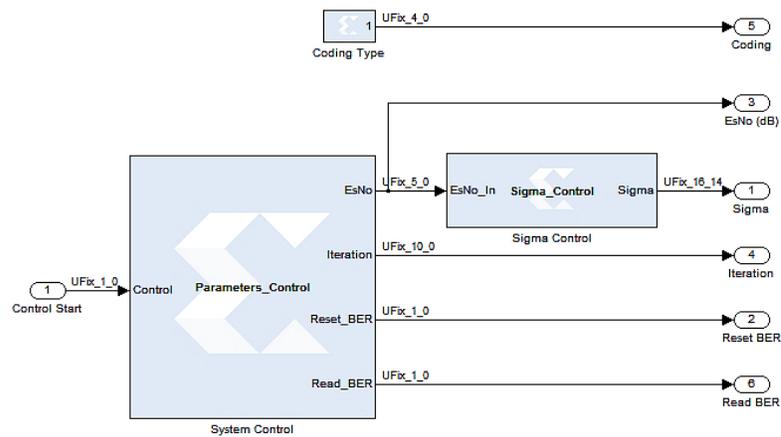


Figura 2.15. Estructura del bloque *Global System Control*.

Partiendo de la definición de la relación señal a ruido (SNR, *Signal to Noise Ratio*) se tiene que:

$$\frac{P_{señal}}{P_{ruido}} = \frac{1}{2\sigma^2} = SNR, \quad (2.2)$$

donde  $P_{señal}$  es la potencia normalizada de la señal codificada ( $P_{señal} = 1W$ ),  $SNR$  es la relación señal a ruido y  $2\sigma^2$  es la potencia de ruido AWGN que adiciona el canal para un sistema en banda base.

Por otra parte, en un sistema de comunicación digital que implementa un esquema de modulación se cumple que [9]:



$$SNR [dB] = \left(\frac{E_b}{N_0}\right) [dB] - 10 \log\left(\frac{1}{\log_2(M)}\right). \quad (2.3).$$

Para un sistema que implementa codificación de línea  $M = 2$  [9]. Además, la relación entre la energía de símbolo de la señal codificada y la energía de bit es:

$$\left(\frac{E_s}{N_0}\right) [dB] = \left(\frac{E_b}{N_0}\right) [dB] - 10 \log\left(\frac{1}{\log_2(2)}\right), \quad (2.4)$$

que corresponde a la ecuación 2.3. Y por ende:

$$SNR [dB] = \left(\frac{E_s}{N_0}\right) [dB]. \quad (2.5).$$

A partir de la ecuación 2.5, finalmente se tiene que:

$$\sigma = \frac{1}{\sqrt{2 \cdot 10^{\frac{E_s/N_0 [dB]}{10}}}}. \quad (2.6)$$

Como se mencionó anteriormente, el valor de la desviación estándar ( $\sigma$ ) corresponde al valor del factor que es multiplicado por la señal de salida del WGNG dentro del bloque *AWGN Channel* de la figura 2.5.

La ecuación 2.6 se implementa como un parámetro más de configuración dentro del subsistema *Sigma Control* visualizado en la figura 2.15 y se declara de forma general para todas las codificaciones, tal como lo muestra la figura 2.16. En este punto cabe destacar la ventaja de normalizar la potencia de la señal codificada ya que permite establecer una comparación de los diferentes esquemas de modulación bajo las mismas condiciones en el canal.

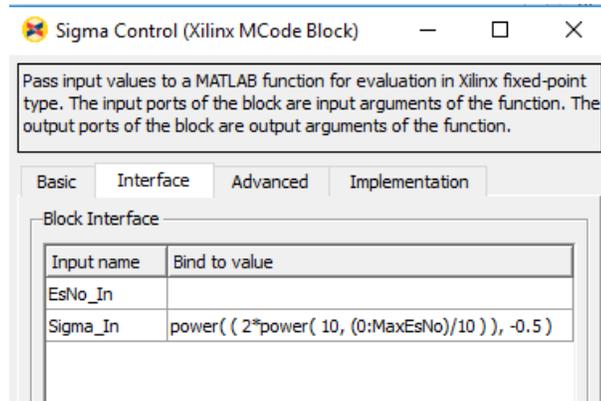
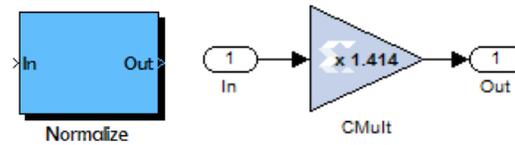


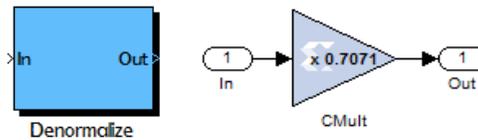
Figura 2.16. Configuración para el cálculo de  $\sigma$  en el bloque *Sigma Control*.

#### 2.4.2.4. Bloques Adicionales

- Normalizador y Desnormalizador de Potencia



a) Normalizador de Potencia.



b) Desnormalizador de Potencia.

Figura 2.17. Normalizador y Desnormalizador de Potencia.

Los bloques expuestos en la figura 2.17, controlan la potencia de la señal codificada. Constan de los bloques *CMult* los cuales amplifican o atenúan la señal a su entrada, mediante la configuración de su ganancia, cuyo valor corresponde al valor de amplitud  $A$  de la señal codificada y depende del tipo de código de línea. Estos valores son los mismos que se calcularon en la sección 1.6 del capítulo uno y se resumen en la tabla 2.1.

Tabla 2.1. Valores de  $A$  para potencia normalizada.

CÓDIGO DE LÍNEA	VALOR DE $A$
UNRZ	$\sqrt{2}$
NRZI	1
PNRZ	1
URZ	2
PRZ	$\sqrt{2}$
AMI-NRZ	$\sqrt{2}$
AMI-RZ	2
MANCHESTER OR BIPHASE L	1
DIFFERENTIAL MANCHESTER	1

- **Sincronizador**

Es el encargado de sincronizar la señal en el receptor con el tiempo de muestreo de la señal recibida.

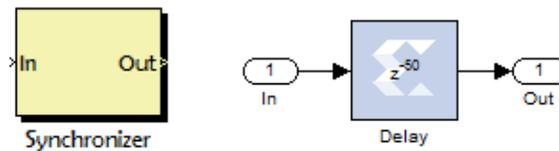


Figura 2.18. Bloque sincronizador.

La estructura de este bloque muestra un módulo de retardo en la figura 2.19, el cual se encarga de generar un retraso en la señal recibida sincronizándola con el instante de muestreo.

- **Muestreo**

Tiene la tarea de tomar cierta cantidad de muestras a una determinada velocidad con el fin de obtener una representación lo más acertada posible de la información transmitida.

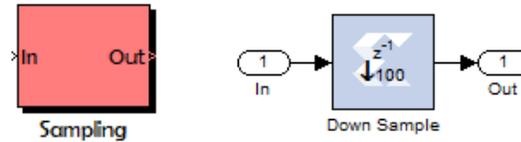


Figura 2.19. Bloque de muestreo.

- **Decisión**

Luego de que la señal recibida es muestreada a la misma tasa de transmisión, el bloque *Symbol Decision*, utiliza un punto de referencia, denominado umbral de decisión, para reconstruir la señal codificada.

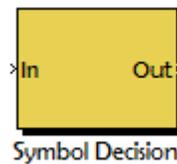


Figura 2.20. Bloque de decisión de símbolos.

El valor del umbral se calcula de forma independiente para cada código de línea. La tabla 2.2 muestra los valores de umbral para las codificaciones según sea su polaridad.

Tabla 2.2. Umbrales de decisión para cada código de línea.

CÓDIGO DE LÍNEA	UMBRAL DE DECISIÓN
UNRZ	0.5
NRZI	0
PNRZ	0
URZ	0.5
PRZ	0
AMI-NRZ	0
AMI-RZ	0
MANCHESTER OR BIPHASE L	0
DIFFERENTIAL MANCHESTER	0

Es fácil deducir a partir de la tabla 2.2, que para los códigos de tipo unipolar el umbral de decisión es 0.5 mientras que para los polares es 0.



Los módulos adicionales descritos anteriormente se implementan en todas las codificaciones analizadas en el presente trabajo de grado.

En el siguiente apartado se muestra brevemente los bloques desarrollados para los codificadores y decodificadores de línea.

#### 2.4.2.5. Codificador y decodificador de línea.

- **UNRZ**

**Codificador UNRZ:** Ya que el bloque generador de bits de la figura 2.4, entrega a su salida una señal de tipo unipolar sin retorno a cero, este cumple las funciones tanto de fuente como codificador UNRZ.

**Decodificador UNRZ:** No se implementa un decodificador como tal, ya que el bloque decisor ilustrado en la figura 2.20, entrega a su salida una señal de tipo UNRZ que finalmente representa los datos decodificados en el receptor.

Para los siguientes códigos de línea se implementan algoritmos de programación, los cuales describen el proceso de codificación y decodificación, a través de archivos con extensión .m, incluidos en los bloques *M-Code* de *System Generator*, teniendo en cuenta los algoritmos de codificación descritos en la sección 1.6 del capítulo uno. La estructura de los bloques implementados para cada código se visualiza a partir de la figura 2.21 hasta la figura 2.28 y una descripción más detallada de estos se encuentra en el apéndice B.



- NRZI

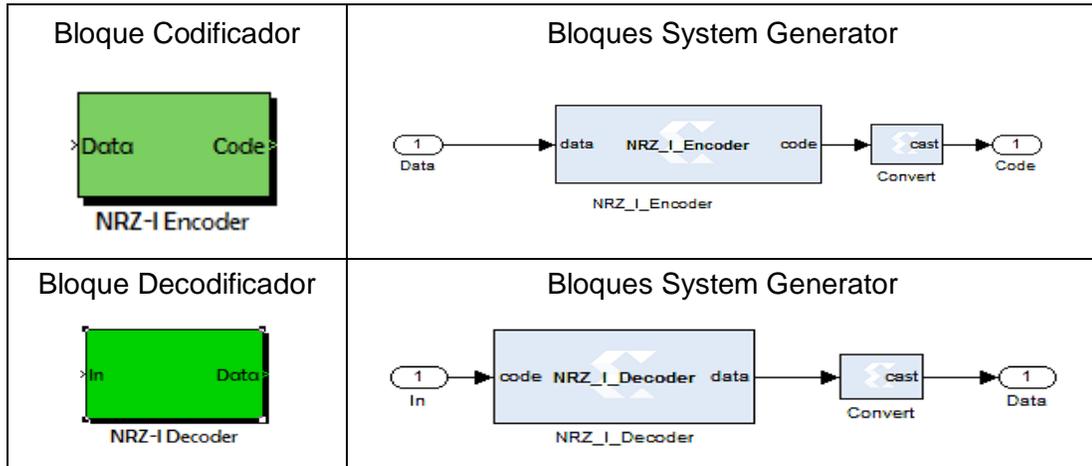


Figura 2.21. Codificador y Decodificador NRZI.

- PNRZ

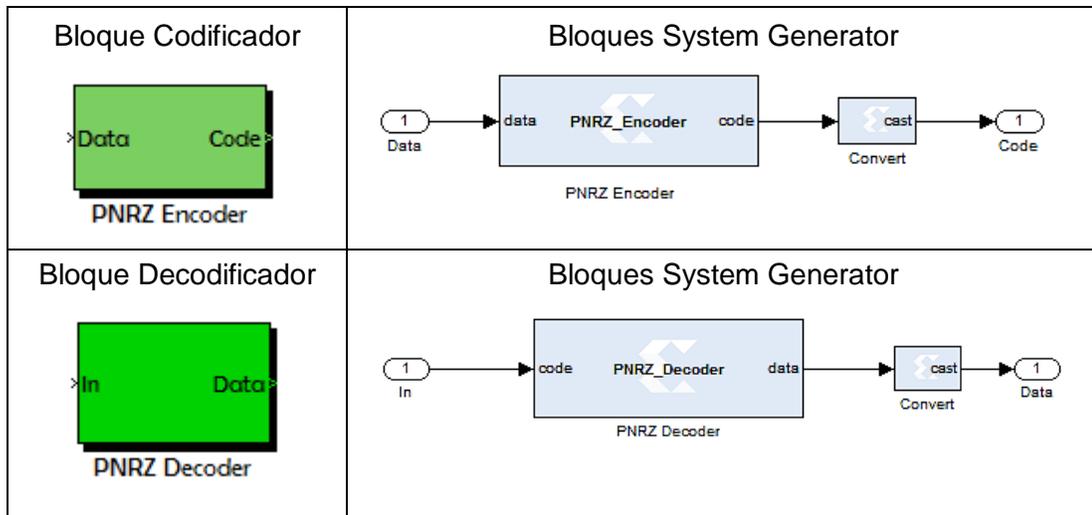


Figura 2.22. Codificador y Decodificador PNRZ.



- URZ

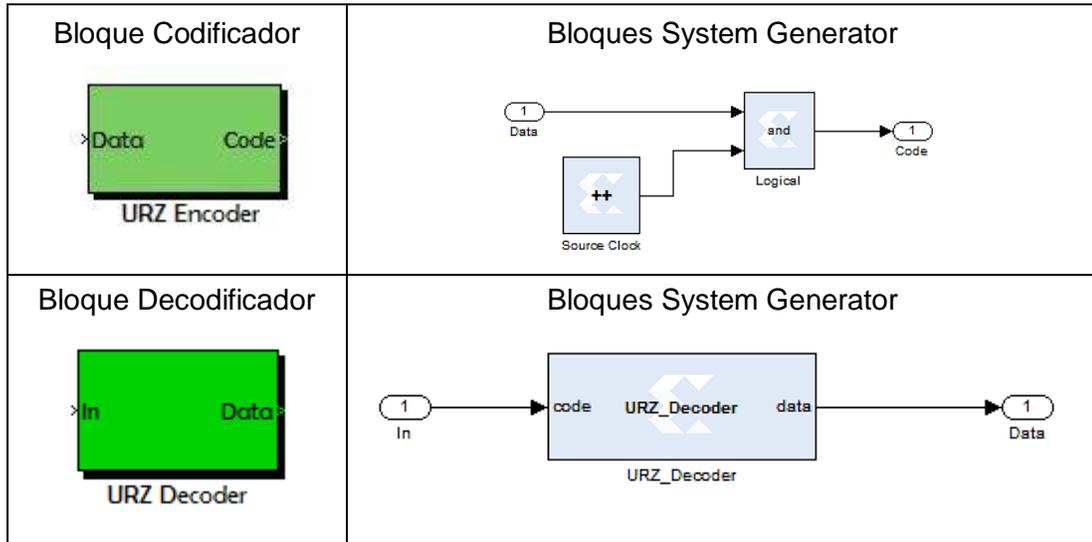


Figura 2.23. Codificador y Decodificador URZ.

- PRZ

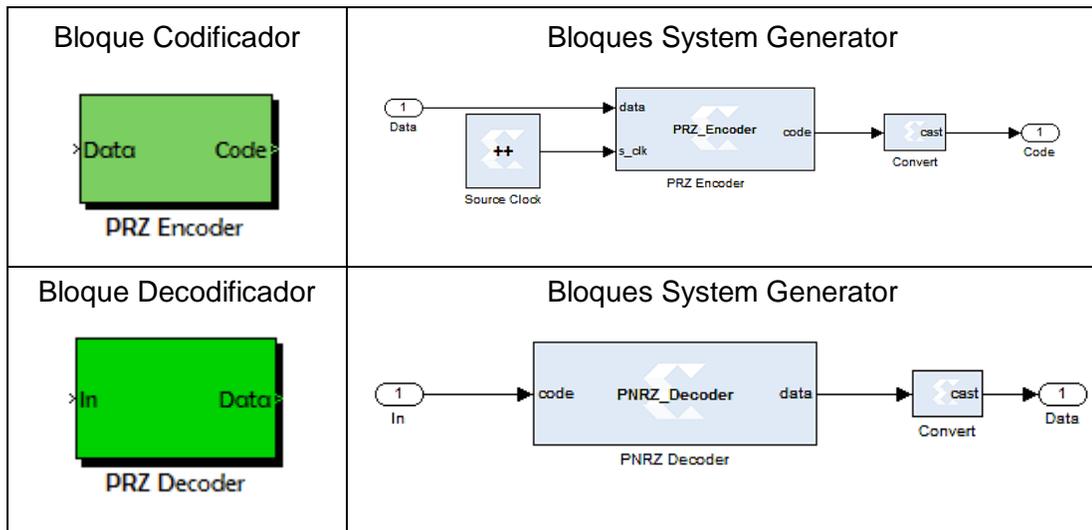


Figura 2.24. Codificador y Decodificador PRZ.

• AMI-NRZ

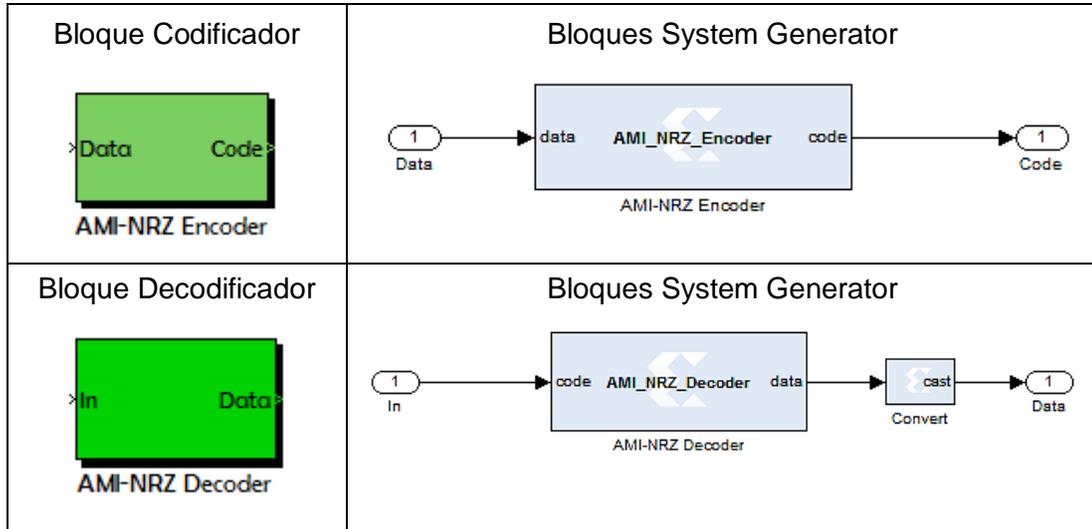


Figura 2.25. Codificador y Decodificador AMI-NRZ.

• AMI-RZ

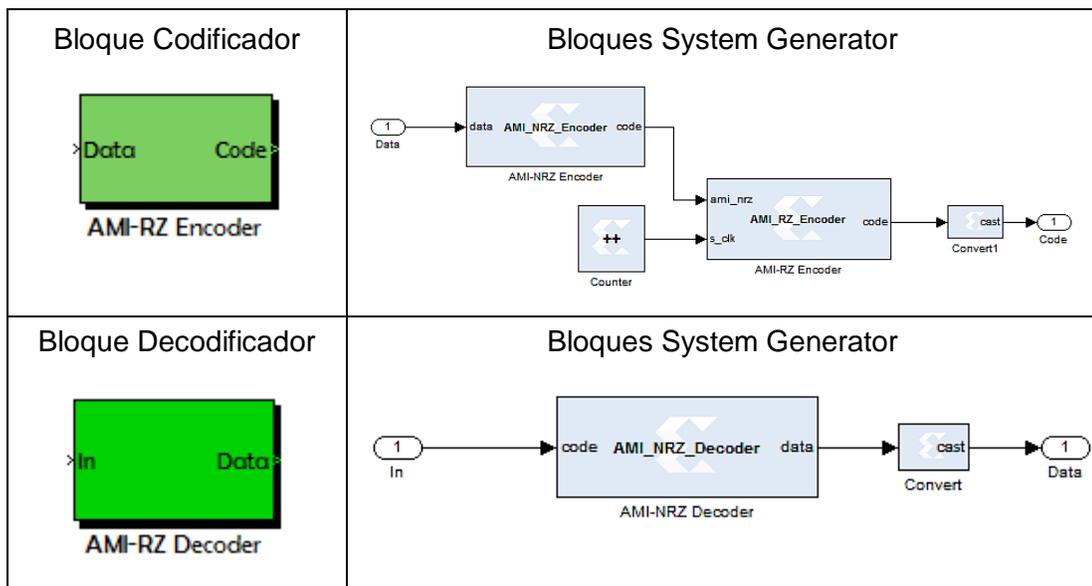


Figura 2.26. Codificador y Decodificador AMI-NRZ.

• **MANCHESTER OR BIPHASE – L**

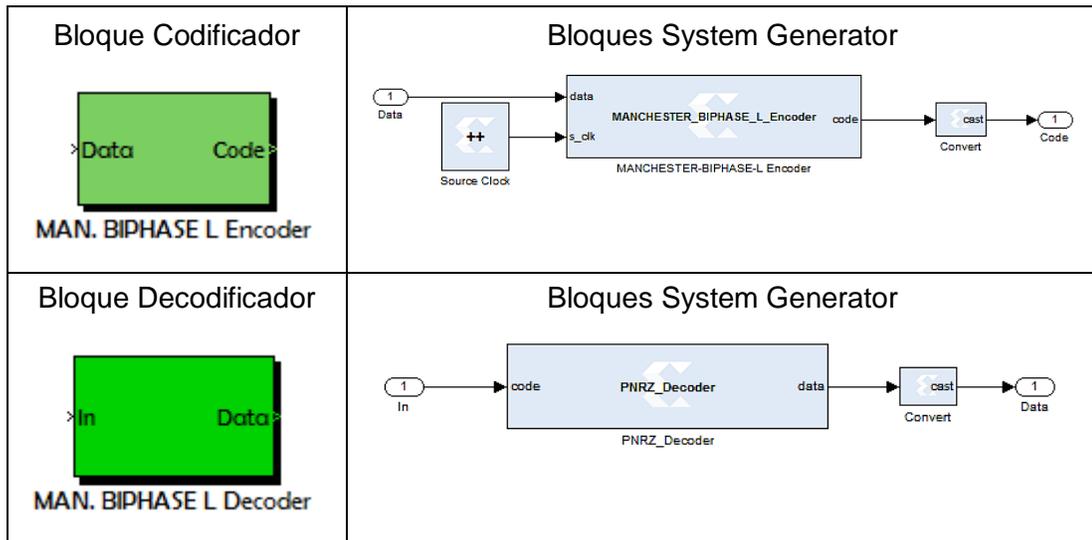


Figura 2.27. Codificador y Decodificador MANCHESTER OR BIPHASE – L.

• **MANCHESTER DIFERENTIAL**

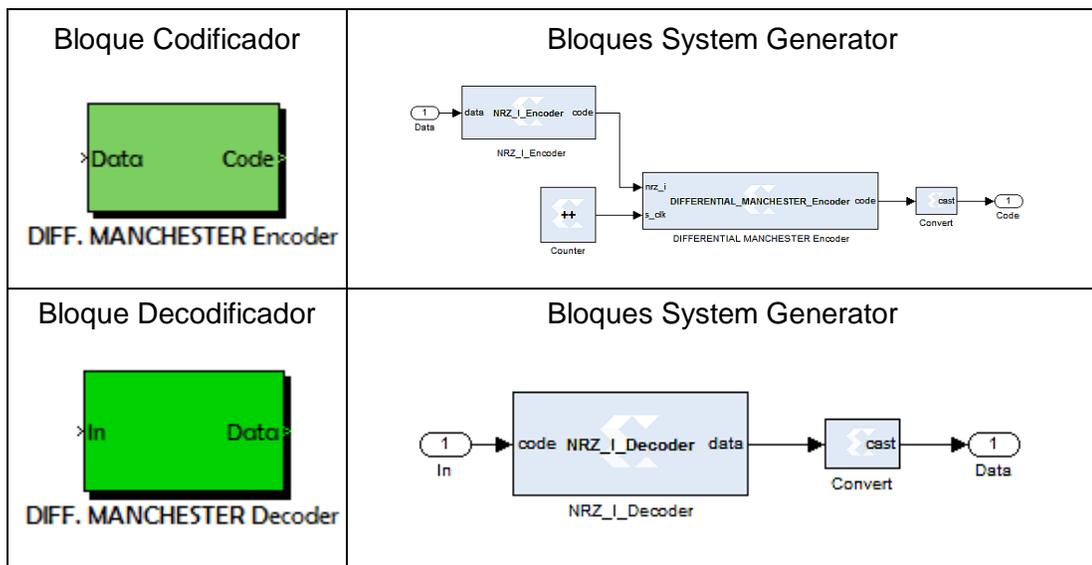


Figura 2.28. Codificador y Decodificador MANCHESTER DIFERENTIAL.

**2.4.2.6. Diseño completo del Sistema de comunicaciones**

Culminado el diseño de los bloques anteriormente descritos, estos se integran con el fin conformar el sistema de comunicación digital con codificación de línea. Figuras 2.29 y 2.30.

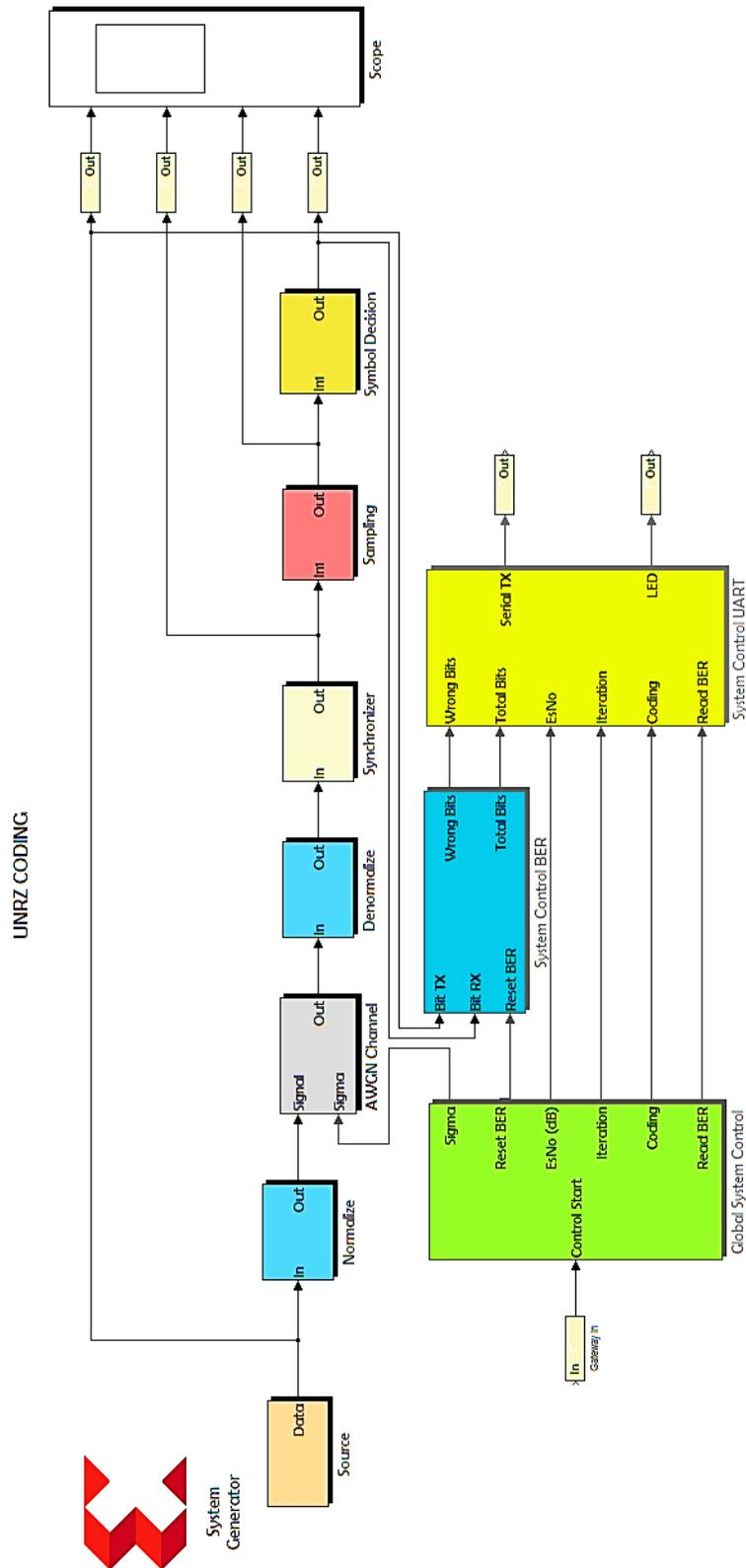


Figura 2.29. Sistema de comunicación digital con codificación UNRZ.

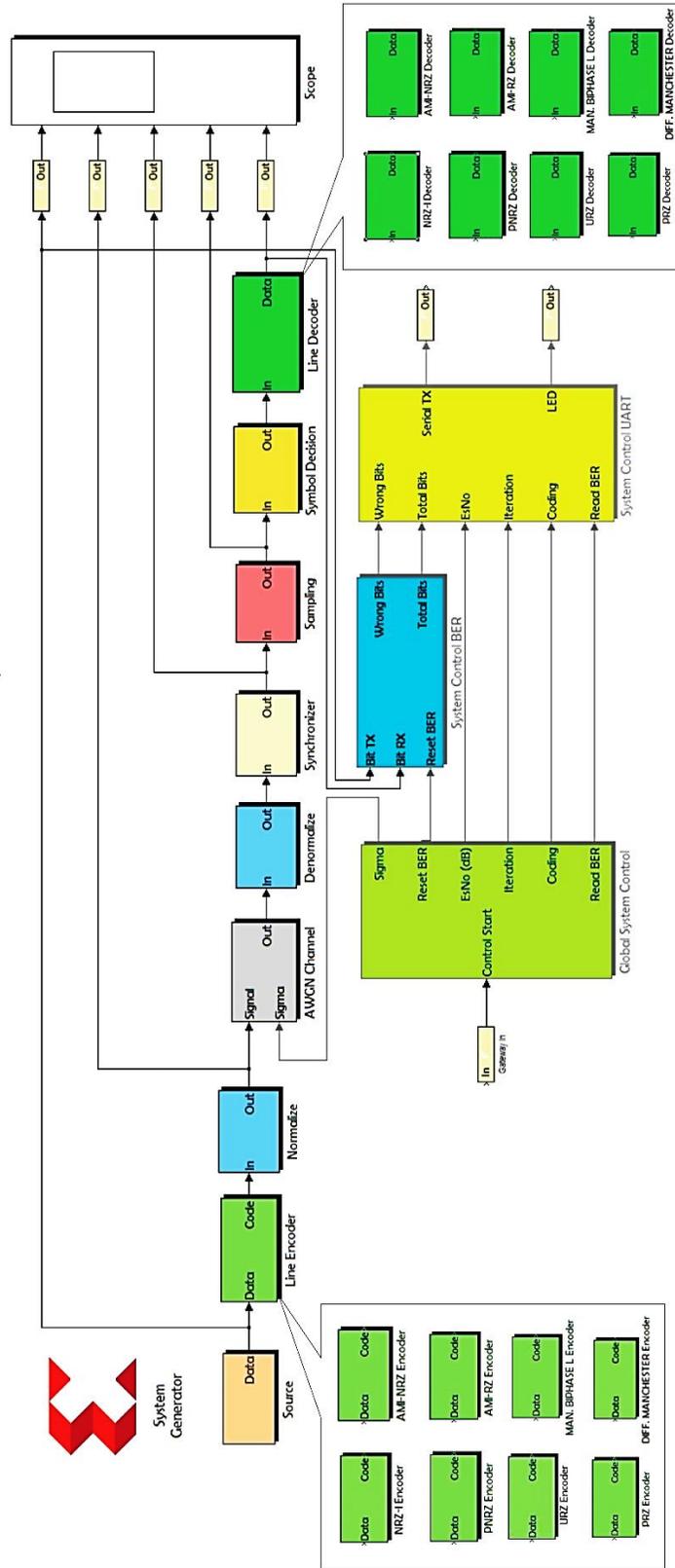


Figura 2.30. Sistema de comunicación digital para diferentes codificaciones de línea.



La Figura 2.30 muestra un sistema de comunicaciones en el cual se pueden implementar todas las codificaciones de línea. Además, como se había propuesto en el capítulo uno, los diferentes esquemas de codificación serán evaluados a partir de un sistema con codificación UNRZ de la figura 2.29, que servirá como punto de referencia.

## 2.5. FASE 3. SIMULACIÓN E IMPLEMENTACIÓN FÍSICA DEL SISTEMA

### 2.5.1 Simulación

La Metodología de simulación utilizada se presenta en la figura 2.31.

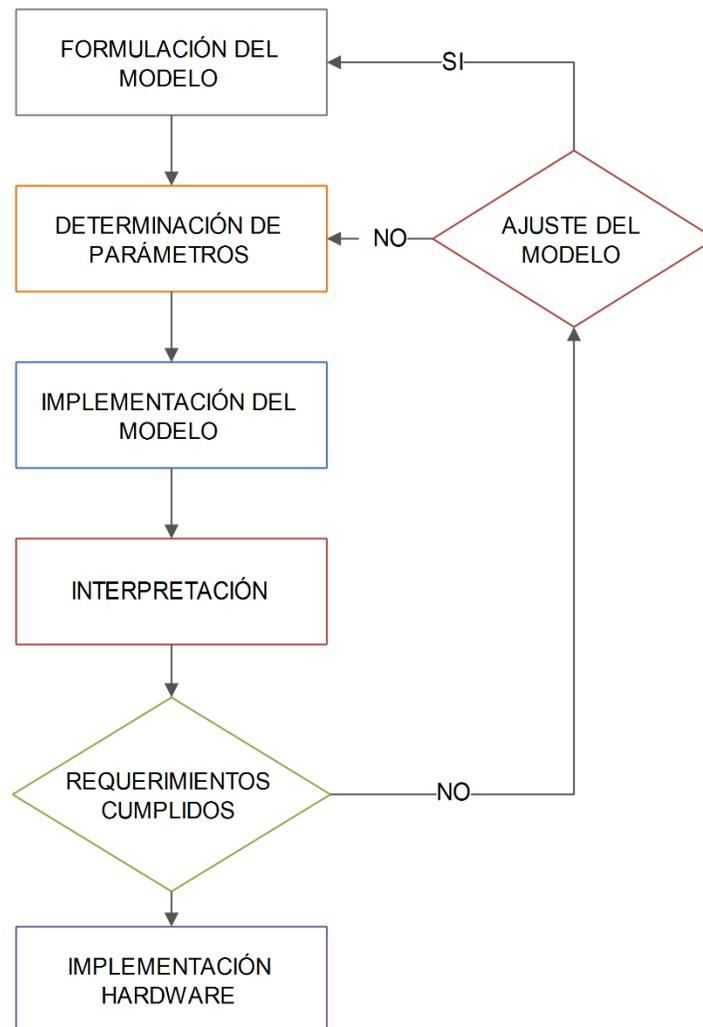


Figura 2.31. Metodología de simulación [23].



- **Formulación del modelo**

Inicialmente se plantea el modelo con codificación UNRZ observado en la figura 2.32, con el fin de observar el comportamiento en paralelo de los sistemas desarrollados *Simulink®* y de *System Generator*.

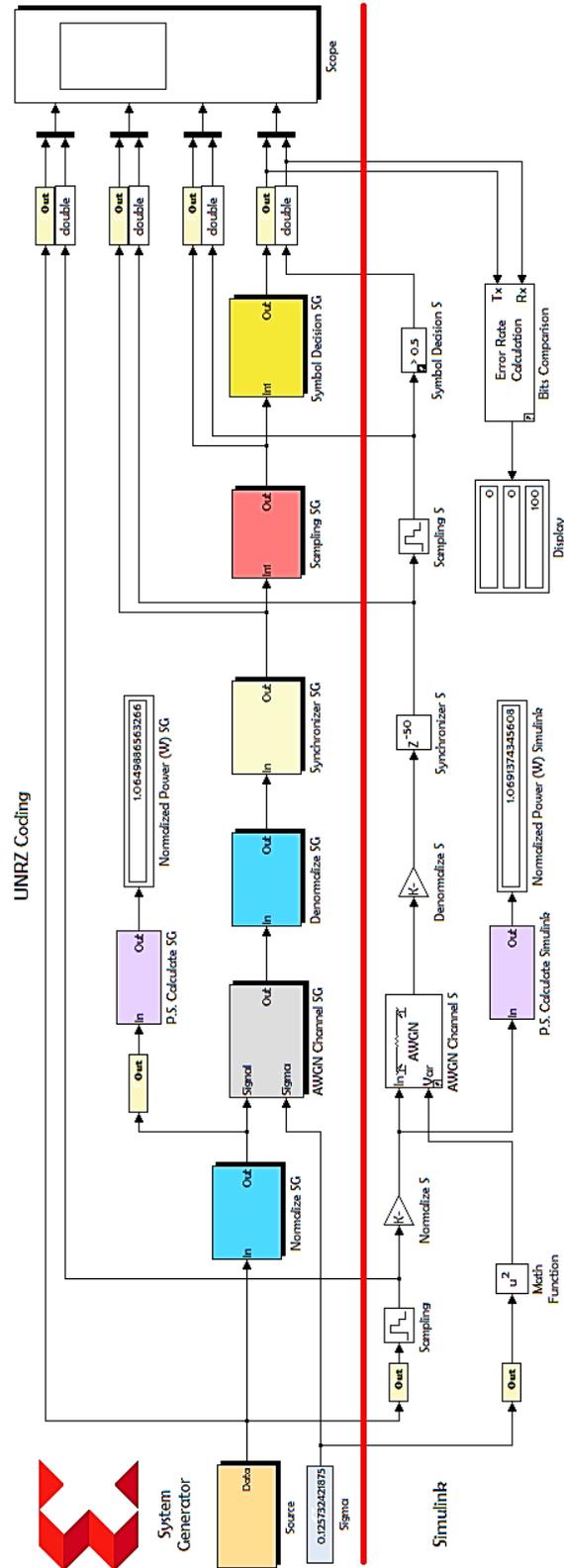


Figura 2.32. Modelo de simulación para el código UNRZ.

En el modelo formulado, se ha dispuesto de un bloque encargado de comparar los bits decodificados a la salida de cada sistema (*Bits Comparison*), habiendo utilizado la misma fuente de datos binarios. Además un bloque que calcula la potencia media de la señal codificada (*P.S. Calculate* y *SG P.S. Calculate Simulink*), siendo igual a  $1W$  debido al proceso de normalización realizado por los bloques *Normalize SG* y *Normalize S* de *System Generator* y de *Simulink®*, respectivamente. En la figura 2.33 se aprecia el bloque encargado del cálculo de la potencia media para una señal aleatoria, determinada por la suma de la varianza y la media [9].

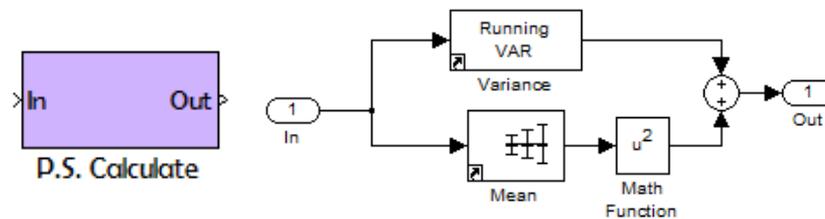


Figura 2.33. Bloque para el cálculo de potencia media de una señal aleatoria.

El modelo de la figura 2.32 es tomado como referencia para el diseño, desarrollo e implementación de las demás codificaciones.

- **Determinación de parámetros**

Para validar el funcionamiento del sistema a implementar, se configura el parámetro sigma calculado para un valor de  $10dB$ , mediante la ecuación 2.2, este valor se escoge en razón de poder observar claramente las formas de onda generadas por los codificadores UNRZ del modelo de *Simulink®* y de *System Generator* en presencia de poco ruido. Además, se configura una velocidad de transmisión de 500 kbps, con la cual se establece el período de bit  $T_b$ . Ya que todo el sistema se modela bajo dispositivos digitales es preciso establecer el número de muestras utilizadas para representar las señales generadas tanto en el codificador como en el canal AWGN, de esta manera se consideró trabajar con un total de 100 muestras por cada período de bit  $T_b$ .



- **Implementación del modelo.**

Determinados los parámetros de configuración, se procede con la simulación del modelo de la figura 2.32. Con el fin de establecer una comparación entre los sistemas de *Simulink*® y de *System Generator*, se consideró apropiado transmitir un total de 100 bits suficientes para observar las señales generadas en varias etapas de la transmisión. Por lo tanto, la figura 2.34 ilustra el proceso para la codificación UNRZ. Igualmente, la formulación del modelo de simulación y las formas de onda correspondientes a las demás codificaciones implementadas se documentaron en el apéndice C.

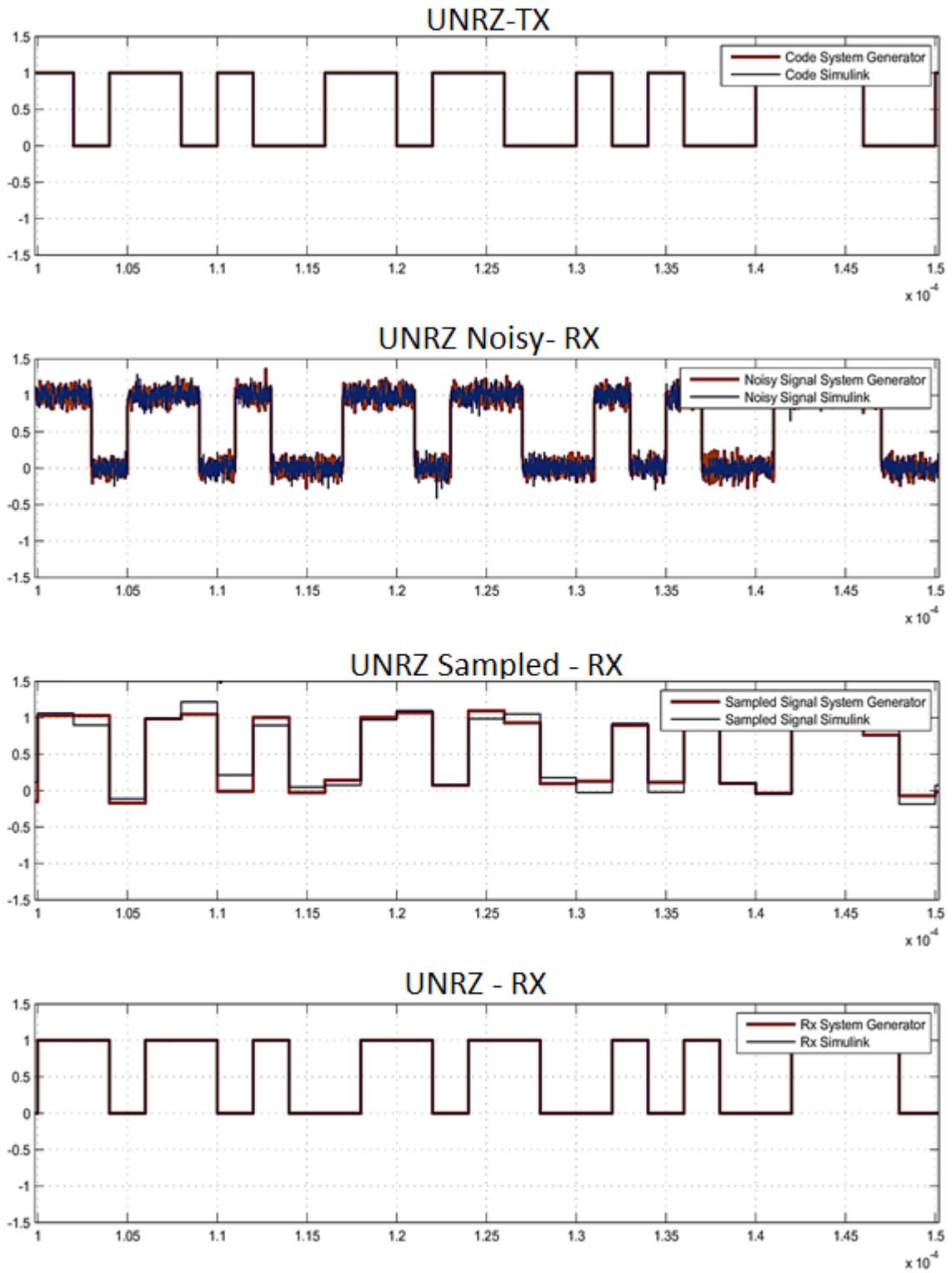


Figura 2.34. Formas de onda de la simulación del código UNRZ.

- **Interpretación**

En la figura 2.34 se observan las gráficas que representan las formas de onda generadas en varias etapas del sistema de comunicación digital con codificación UNRZ. La primera gráfica muestra la señal codificada sin retorno a cero para una secuencia de bits aleatorios, en la segunda gráfica se observa la deformación de las señales debido a la adición de ruido por parte del canal AWGN. En la siguiente gráfica se ve claramente una diferencia entre los niveles de las señales muestreadas, esto se debe a la aleatoriedad por parte del ruido, finalmente en la última gráfica se visualiza el correcto proceso de decodificación de las señales, validando que el comportamiento de ambos modelos a nivel de simulación es muy similar.

### 2.5.2 Implementación Hardware Del Sistema

Para la implementación del sistema de comunicación digital, 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 para ejecutar las tareas requeridas.

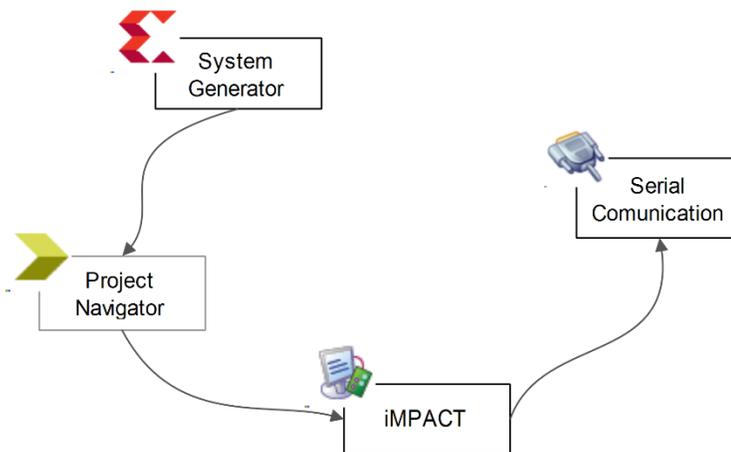


Figura 2.35. Proceso de implementación Hardware.



El proceso de la implementación hardware ilustrado en la figura 2.35, se inicia con la configuración de *System Generator*, mediante su panel de opciones, en el cual se establecen los parámetros principales como el modo de compilación y la referencia del 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 *.bit* y las limitaciones de tiempo (*timing constraints*), los cuales determinan el retardo máximo que el sistema estaría dispuesto a soportar. Luego de obtener el archivo *.bit* se procede a cargar el código sobre el FPGA seleccionado (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 procesados por la placa de desarrollo a través del puerto serial que envía la información del sistema de comunicación con codificación de línea a la aplicación de escritorio, desarrollada en *visual basic*, cuya descripción se encuentra en el apéndice D.

## 2.6. FASE 4. VALIDACIÓN Y PRUEBAS

### 2.6.1. Validación

Para realizar la validación de la implementación hardware del sistema de comunicación en banda base con codificación de línea, se utiliza la aplicación desarrollada en *Visual Basic*, la cual captura los datos de la tasa de error de bit obtenida a partir de la trama enviada desde la placa de desarrollo *Spartan 3A*, cuyo formato se describió en la figura 2.15.

En la figura 2.36, se aprecia la captura de los datos obtenidos para la codificación UNRZ.

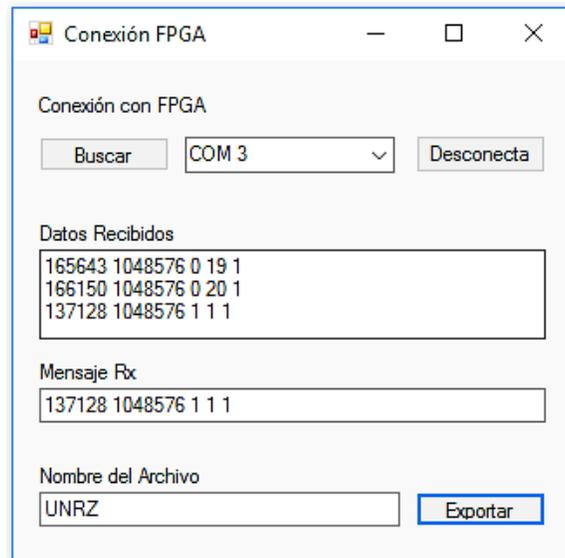


Figura 2.36. Captura de la tasa de error de bit en la aplicación de escritorio.

Los datos recibidos en la aplicación de escritorio se almacenan en una memoria temporal de la aplicación hasta que la FPGA finalice la transmisión. Luego estos son exportados por el usuario a un archivo Excel.

Por otra parte la recolección de datos desde el modelo implementado en *Simulink*® se realiza mediante algoritmos de programación de Matlab, los cuales controlan los parámetros de configuración para cada valor de  $E_s/N_0$  en cada modelo, además los datos obtenidos se exportan en archivos Excel que contienen la información de la BER para la simulación correspondiente al modelo desarrollado.

Los archivos Excel obtenidos de la implementación en la FPGA y *Simulink*® son procesados por un *script* desarrollado en Matlab, con el fin de obtener las curvas de desempeño a nivel de BER para cada codificación. Estas curvas son comparadas con la curva teórica graficada a partir de las ecuaciones presentadas en la tabla 2.3 y cuya deducción se presenta en el Anexo B, y corresponde a las probabilidades de error para cada código de línea dadas en términos de la función  $Q$ .



Tabla 2.3. Probabilidad de Error para los códigos de línea [25].

CÓDIGO DE LÍNEA	PROBABILIDAD DE ERROR
UNRZ	$P_e = Q\left(\sqrt{\frac{E_s}{N_0}}\right)$
NRZ-I	$P_e \approx 2Q\left(\sqrt{2\frac{E_s}{N_0}}\right)$
PNRZ	$P_e = Q\left(\sqrt{2\frac{E_s}{N_0}}\right)$
URZ	$P_e = Q\left(\sqrt{\frac{E_s}{N_0}}\right)$
PRZ	$P_e = Q\left(\sqrt{2\frac{E_s}{N_0}}\right)$
AMI-NRZ	$P_e = \frac{3}{2}Q\left(\sqrt{\frac{E_s}{N_0}}\right)$
AMI-RZ	$P_e = \frac{3}{2}Q\left(\sqrt{\frac{E_s}{N_0}}\right)$
MANCHESTER OR BIPHASE L	$P_e = Q\left(\sqrt{2\frac{E_s}{N_0}}\right)$
DIFFERENTIAL MANCHESTER	$P_e \approx 2Q\left(\sqrt{2\frac{E_s}{N_0}}\right)$

Con el objetivo de validar el funcionamiento del sistema de comunicación digital con los diferentes esquemas de codificación a nivel simulación e implementación se propuso inicialmente la transmisión de 1048576 bits a una velocidad de 500 Kbps con el fin de observar el comportamiento de los modelos desarrollados mediante las curvas de desempeño.

Las figuras 2.37 a 2.45 muestran los resultados obtenidos en *Simulink*® y en la FPGA, correspondiente a la simulación e implementación hardware.

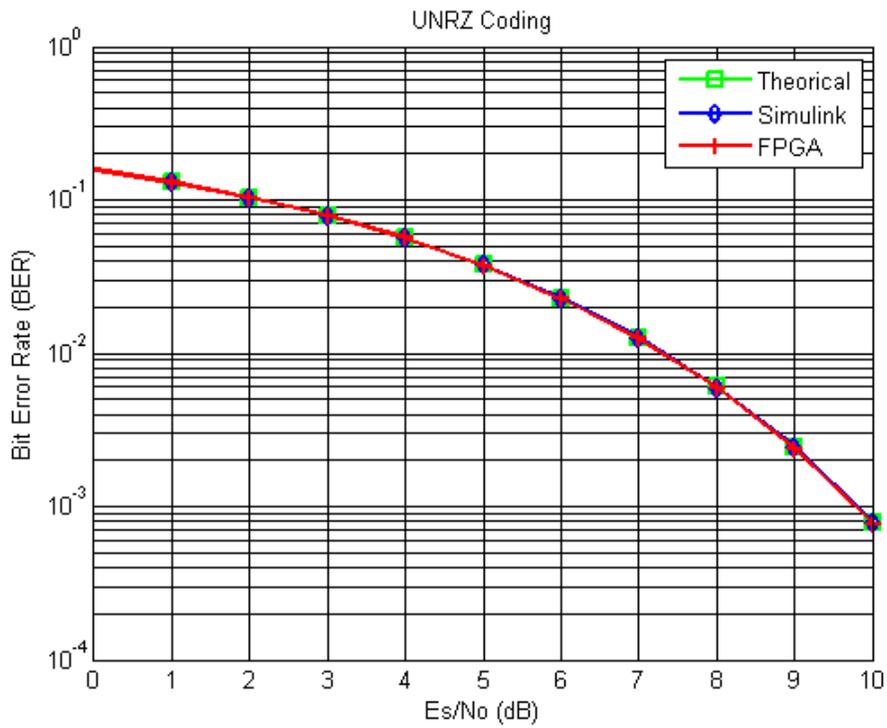


Figura 2.37. Curva de desempeño codificación UNRZ.

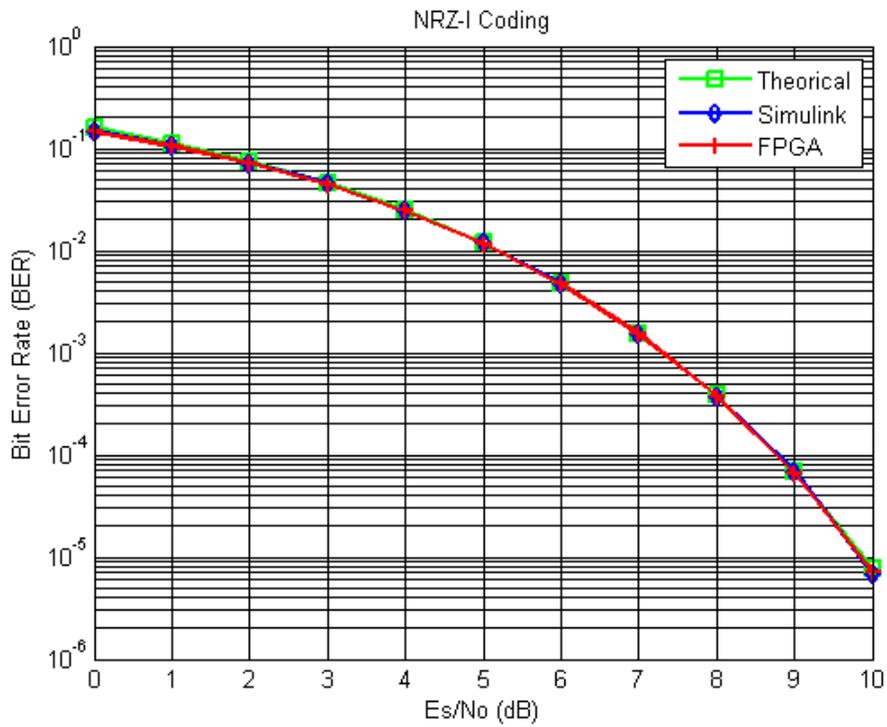


Figura 2.38. Curva de desempeño codificación NRZ-I.

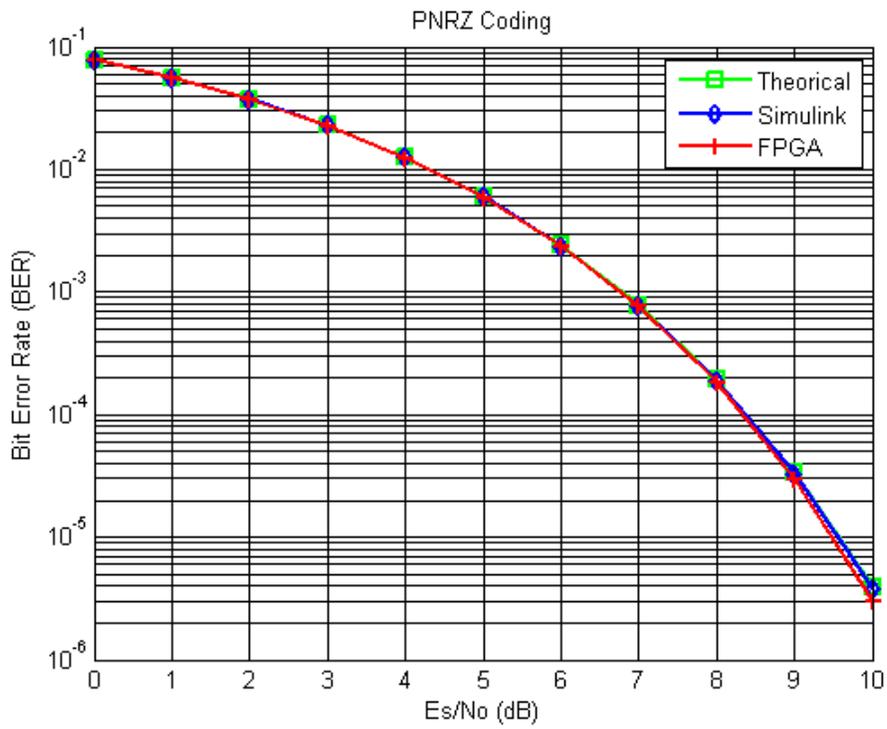


Figura 2.39. Curva de desempeño codificación PNRZ.

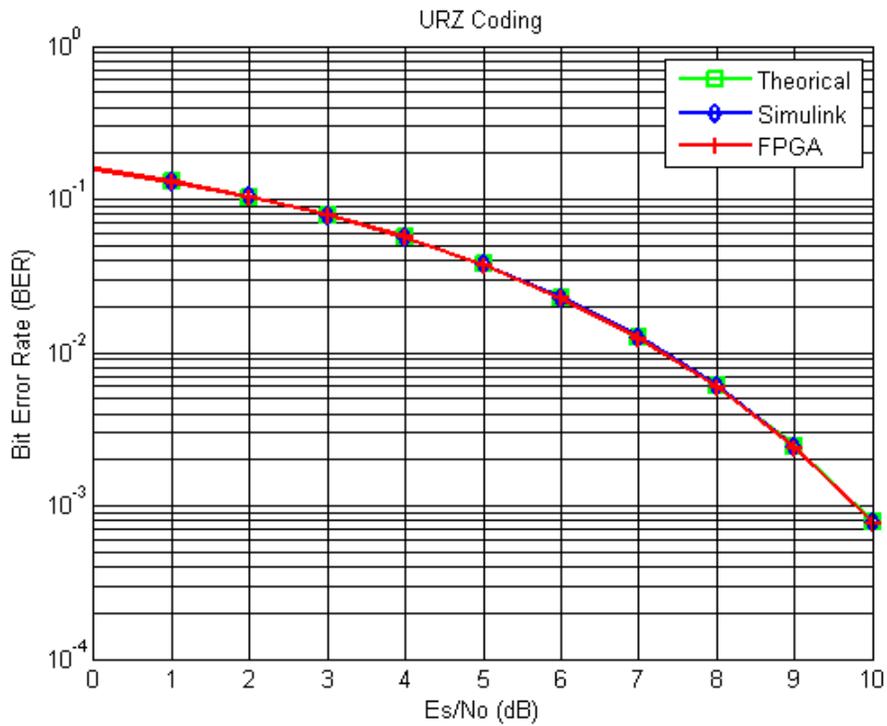


Figura 2.40. Curva de desempeño codificación URZ.

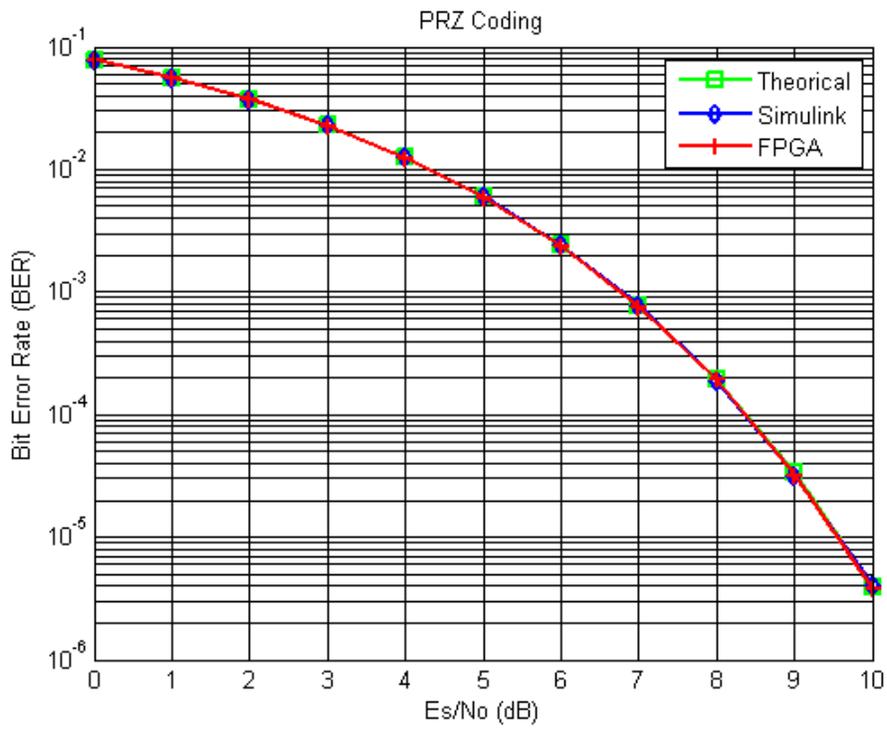


Figura 2.41. Curva de desempeño codificación PRZ.

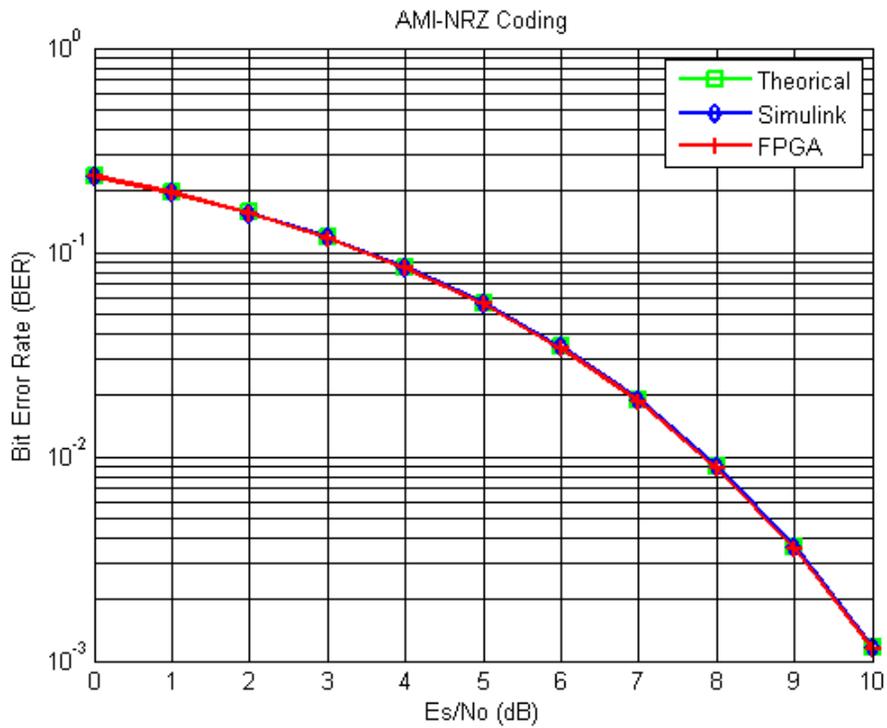


Figura 2.42. Curva de desempeño codificación AMI-NRZ.

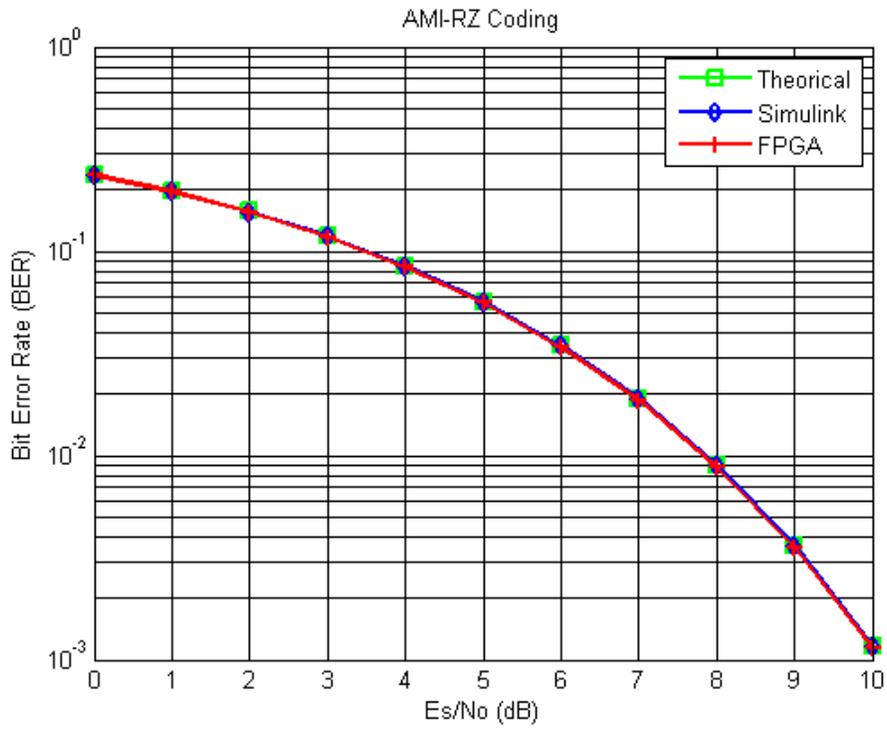


Figura 2.43. Curva de desempeño codificación AMI-RZ.

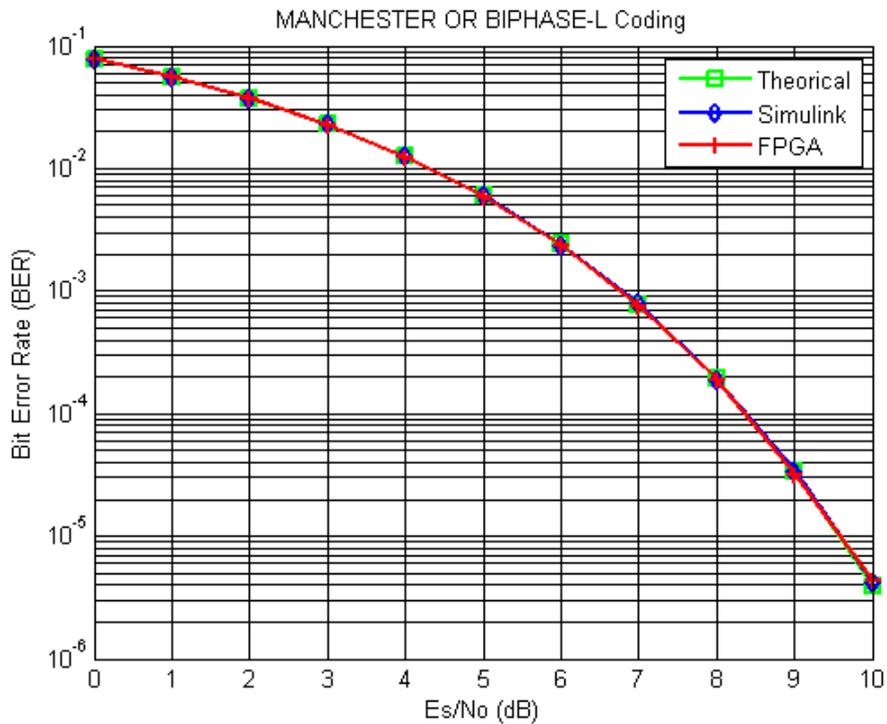


Figura 2.44. Curva de desempeño codificación Manchester or Biphase-L.

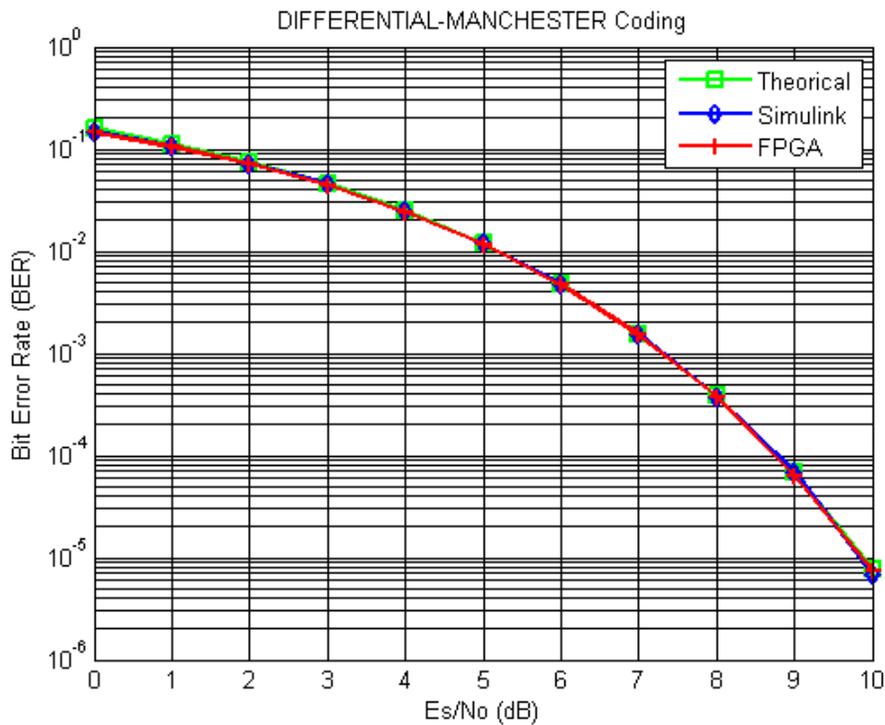


Figura 2.45. Curva de desempeño codificación Differential Manchester.

Teniendo en cuenta las figuras 2.37 a 2.45 se valida correctamente el funcionamiento del modelo desarrollado *Simulink*® y *System Generator* presentado en la figura 2.30 e implementado en la FPGA. Se puede observar en las ilustraciones que las curvas de desempeño del sistema implementado y simulado tienen el mismo comportamiento que las teóricas, lo cual permite asegurar que el modelo propuesto es consistente y por ende apto para el estudio del desempeño sobre hardware reconfigurable. Las tablas de datos a partir de las cuales se graficaron las anteriores curvas de desempeño se muestran en el apéndice E.

Una vez finalizada la fase de validación tanto a nivel de simulación como de implementación hardware, en el siguiente capítulo se continúa con la fase de pruebas y análisis de los resultados obtenidos al implementar el sistema de comunicación digital con codificación de línea mediante la variación de parámetros de funcionamiento.





## CAPÍTULO 3

### PRUEBAS Y ANÁLISIS DE RESULTADOS

Para el desarrollo de este capítulo se procedió a establecer un plan de pruebas con el objetivo de analizar el comportamiento del sistema de comunicación en diferentes escenarios.

#### 3.1. PLAN DE PRUEBAS

El plan de pruebas consistió en la variación de dos parámetros reconfigurables como son la velocidad de transmisión de datos y la cantidad de bits transmitidos, con el objetivo de observar el desempeño de cada código de línea a nivel de simulación e implementación bajo los dos escenarios que se ilustran en la tabla 3.1. En el primer escenario se plantea la transmisión de 1536 bits variando la tasa de transmisión de los datos entre 10 y 500 Kbps, de forma similar en el escenario dos se realiza la misma variación en la tasa de transmisión, pero para un total de 1048576 bits. Los valores establecidos en el plan de pruebas se documentaron en razón de dar contraste a los resultados obtenidos en varios escenarios que tuvieron en cuenta diferentes rangos, de esta manera el plan de pruebas de la tabla 3.1, busca considerar el peor y mejor de los casos para evaluar el desempeño del sistema de comunicaciones con codificación de línea.

Tabla 3.1. Escenarios de plan de pruebas.

ESCENARIO	PARÁMETROS DE CONFIGURACIÓN			
	CANTIDAD DE BITS TRANSMITIDOS		VELOCIDAD DE TRANSMISIÓN	
	1536	1048576	10 Kbps	500 Kbps
1	✓		✓	
	✓			✓
2		✓	✓	
		✓		✓

#### 3.2. RESULTADOS Y ANÁLISIS

En esta sección se comparan los resultados obtenidos al implementar el sistema diseñado sobre hardware reconfigurable (FPGA), para los diferentes escenarios planteados en la



tabla 3.1. Cabe mencionar que luego de observar detenidamente los valores de BER obtenidos en cada sistema para diferente número de repeticiones en cada escenario, se decidió documentar un total de 20 iteraciones por cada valor de  $E_s/N_0$ , ya que las variaciones en cada una de las pruebas no tuvieron una desviación significativa, lo que llevó a una obtención de resultados que presentaran una tendencia estadística confiable y acorde a los valores teóricos, tomando como dato de tabulación para las gráficas expuestas, la media de los valores medidos.

### 3.2.1. Escenario 1

Las figuras 3.1 a 3.9, presentan las curvas de desempeño a nivel de BER de los códigos de línea, habiéndose transmitido un total de 1536 bits a una velocidad de 10 Kbps y 500 Kbps. Las tablas a partir de las cuales se graficaron las curvas de desempeño se muestran en el Apéndice F.1.

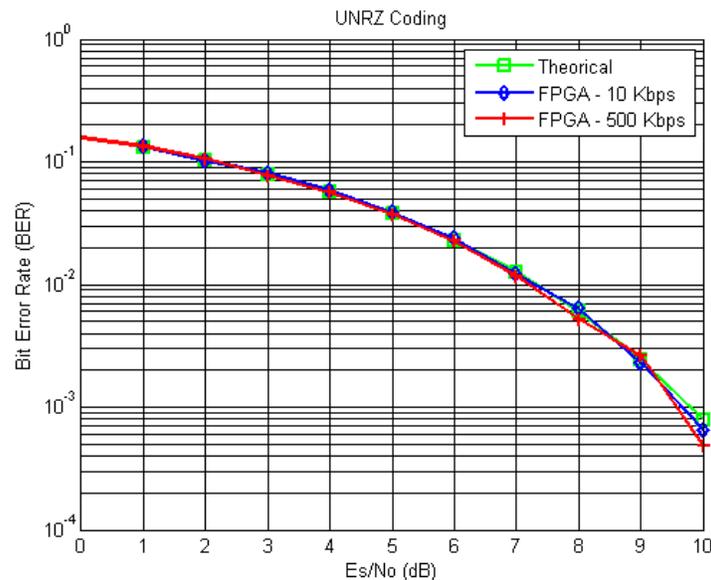


Figura 3.1. Curvas de desempeño para la codificación UNRZ. Transmisión de 1536 bits.

En la figura 3.1 se observa que las curvas de desempeño para la implementación de la codificación UNRZ a 10 y 500 Kbps están estrechamente relacionadas y aunque presentan desviaciones relativamente pequeñas con respecto a la curva teórica, la

mayor parte del trazo de las curvas están traslapadas, lo que indica que para una transmisión de 1536 bits el sistema implementado funciona correctamente.

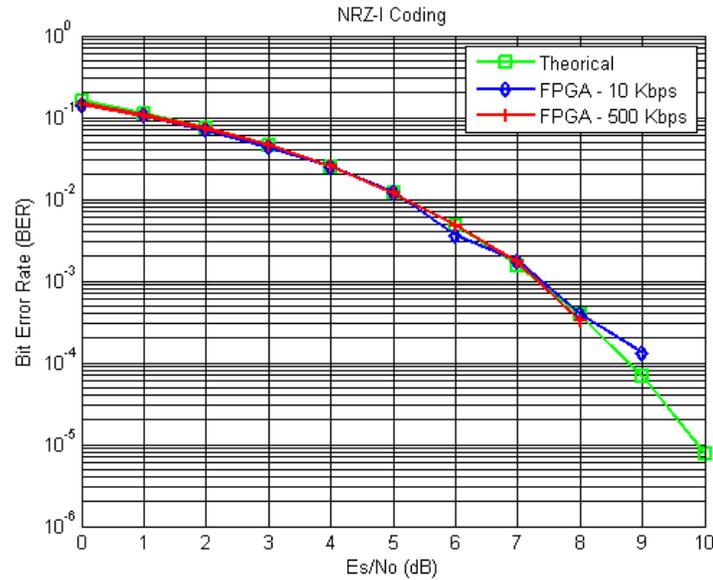


Figura 3.2. Curvas de desempeño para la codificación NRZ-I. Transmisión de 1536 bits.

Como se puede apreciar en la figura 3.2, las curvas de desempeño de la codificación NRZ-I se traslapan con la curva teórica pero solamente hasta valores objetivo de BER aproximados de  $10^{-4}$ , sin embargo para valores de  $E_s/N_0$  mayores a 8 dB las curvas obtenidas no brindan una idea clara sobre el desempeño de la codificación para ambas velocidades de transmisión. Esto quiere decir que para la transmisión de 1536 bits no es posible medir confiablemente la BER.

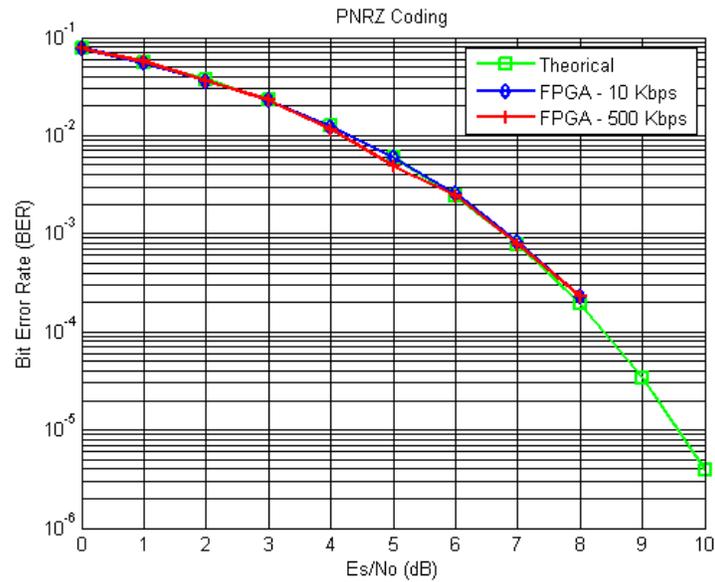


Figura 3.3. Curvas de desempeño para la codificación PNRZ. Transmisión de 1536 bits.

La figura 3.3 indica que la medición de la BER para la codificación PNRZ, es confiable solo hasta niveles de  $E_s/N_0$  menores a 7 dB en donde las curvas obtenidas tienen un comportamiento aproximado a la teórica.

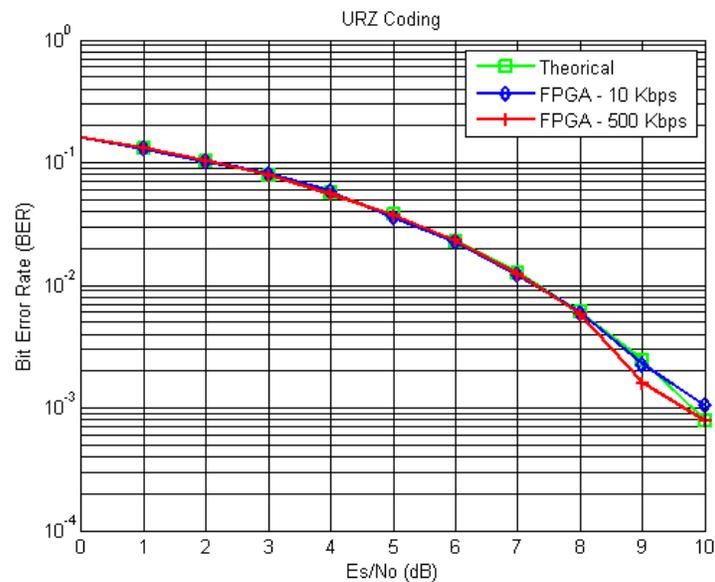


Figura 3.4. Curvas de desempeño para la codificación URZ. Transmisión de 1536 bits.

De acuerdo a la figura 3.4, se concluye que el desempeño de la codificación URZ sigue el mismo comportamiento representado por la curva teórica, sin embargo se observa que para una velocidad de transmisión de 500 Kbps y un valor de  $E_s/N_0$  igual a 9 dB existe una variación relativamente mayor en el valor BER con respecto a la curva teórica.

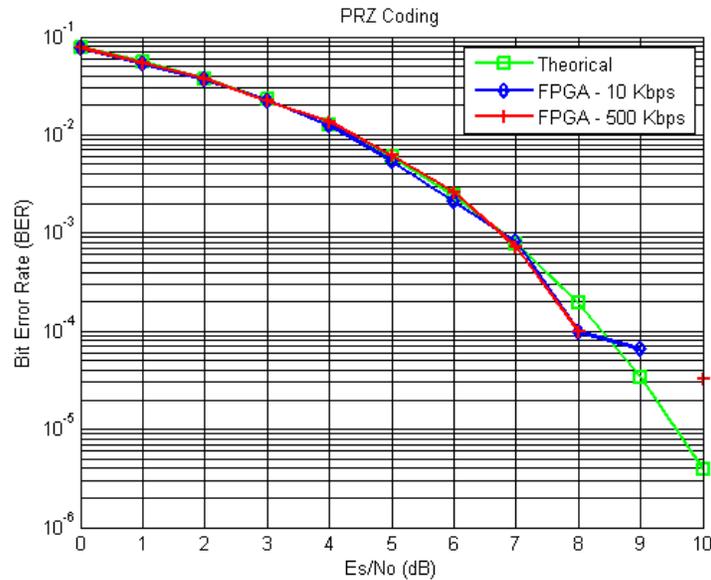


Figura 3.5. Curvas de desempeño para la codificación PRZ. Transmisión de 1536 bits.

En el caso del desempeño de la codificación PRZ, se observa a partir de la figura 3.5 que, en las curvas de desempeño obtenidas, existen variaciones similares a las presentadas en la figura 3.2 correspondientes a las curvas de desempeño para la codificación NRZ-I.

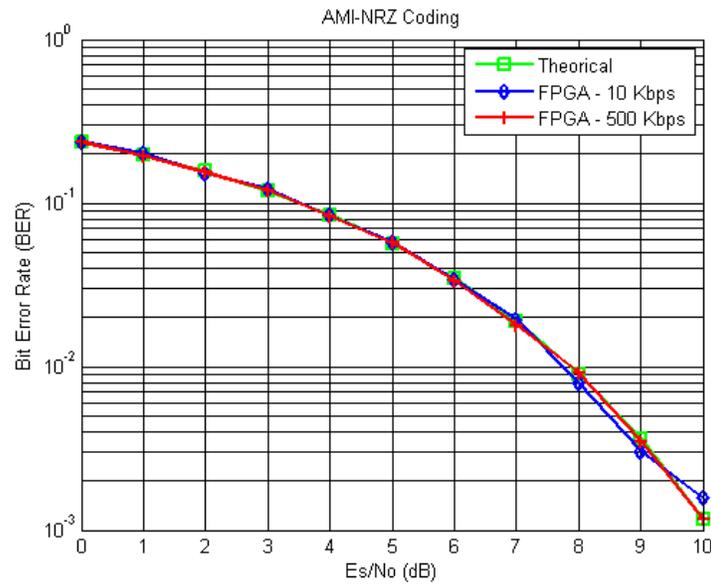


Figura 3.6. Curvas de desempeño para la codificación AMI-NRZ. Transmisión de 1536 bits.

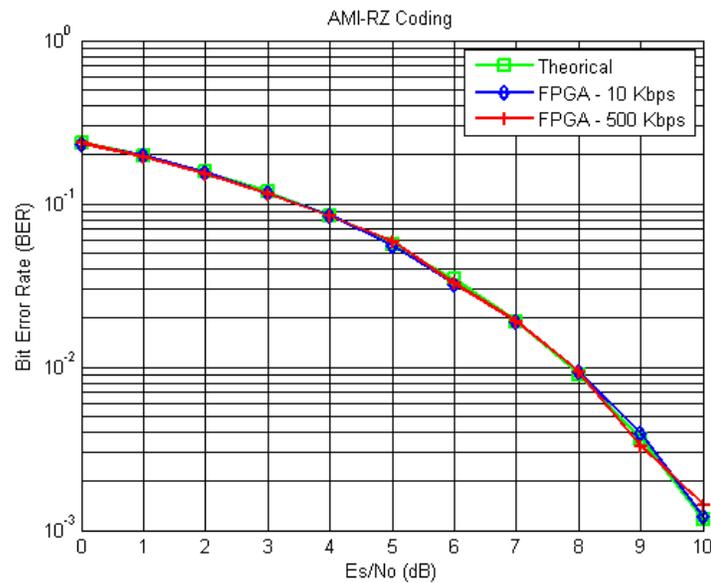


Figura 3.7. Curvas de desempeño para la codificación AMI-RZ. Transmisión de 1536 bits.

Las curvas de desempeño para la codificación AMI, en sus dos variantes NRZ y RZ, mostradas en las figuras 3.6 y 3.7 respectivamente, exponen un desempeño muy similar, adicionalmente es claro que las tres curvas graficadas están estrechamente relacionadas indicando que para una transmisión de 1536 bits es posible obtener una medida confiable en el cálculo de la BER.

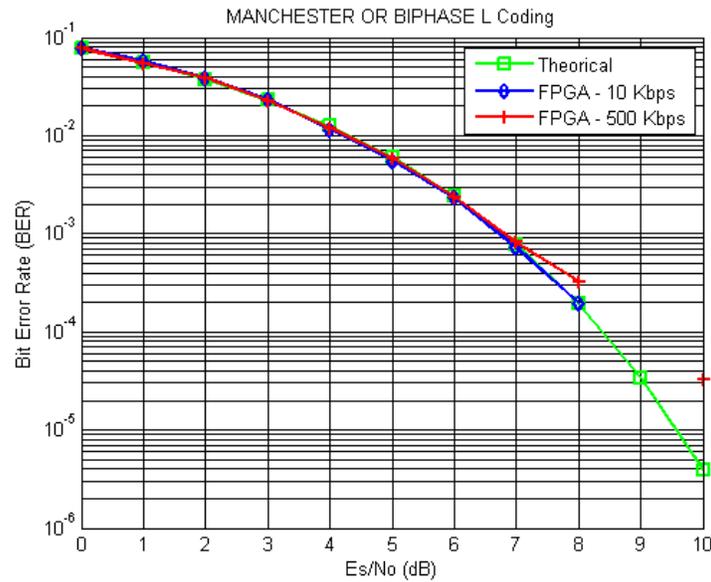


Figura 3.8. Curvas de desempeño para la codificación MANCHESTER OR BIPHASE L. Transmisión de 1536 bits.

De la misma forma que la codificación PNRZ y PRZ, en la figura 3.8 se aprecia un comportamiento muy similar por parte de las curvas que describen el desempeño para la codificación MANCHESTER OR BIPHASE L, en el que para valores grandes de  $E_s/N_0$  medir la BER no resulta un proceso confiable para una trasmisión de 1536 bits.

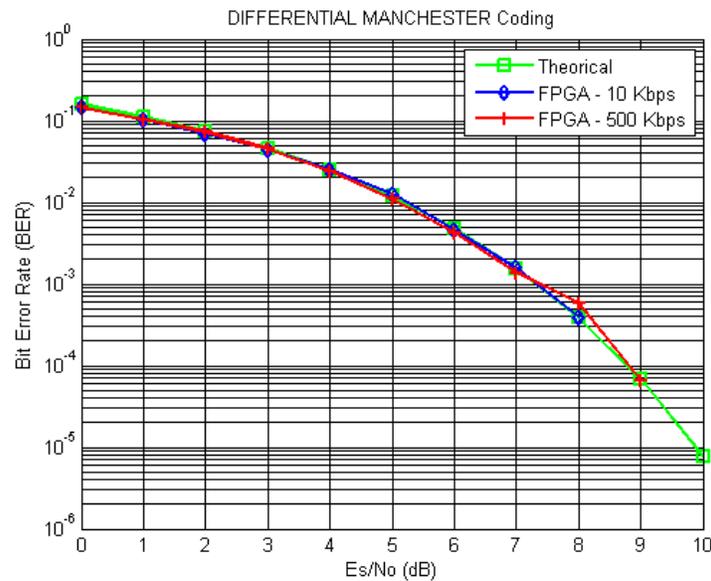


Figura 3.9. Curvas de desempeño para la codificación DIFFERENTIAL MANCHESTER. Transmisión de 1536 bits.



La gráfica 3.9 muestra un comportamiento muy similar que la codificación NRZ-I, cuyo desempeño se muestra en la gráfica 3.5. Esto debido a que DIFFERENTIAL MANCHESTER es un código diferencial al igual que NRZ-I.

Teniendo en cuenta las figuras 3.1 a 3.9, consistentemente se aprecia que las gráficas para los códigos de tipo polar presentaron variaciones en los últimos valores de BER medidos con respecto a la curva teórica, específicamente para  $E_s/N_0$  mayor a 7 dB, concerniente a una BER teórica menor a  $10^{-3}$ , estas desviaciones están ligadas a la polaridad de la señal codificada, que siendo de tipo polar, la distancia entre los niveles de tensión (positivo y negativo) y el umbral de decisión en el receptor es mayor que para el caso unipolar, esto supone una menor probabilidad de decodificar erróneamente un bit dentro de un rango de bits transmitidos por la fuente. Si este rango es lo suficientemente grande, es más probable decodificar erróneamente un bit y obtener una medición más precisa de la BER, y justamente la transmisión de 1536 bits, en este escenario, no muestra curvas con una tendencia definida para valores de  $E_s/N_0$  relativamente grandes. Por otra parte, para los códigos de tipo unipolar y bipolar, aunque también se aprecia cierta desviación de los valores medidos, estos siguen una tendencia estadística un poco más definida alrededor de la curva teórica, esto se atribuye a que la probabilidad de error de bit es mayor que para el caso de los códigos polares, pero en general, no garantizan una estimación confiable para valores de BER por debajo de  $10^{-2}$ .

### 3.2.2. Escenario 2

Para la transmisión de un total de 1048576 bits a una velocidad de 10 Kbps y 500 Kbps, las figuras 3.10 a 3.18, ilustran las curvas de desempeño a nivel de BER de los códigos de línea implementados. Las tablas que contienen los datos de tabulación para las gráficas se documentaron en el Apéndice F.2.

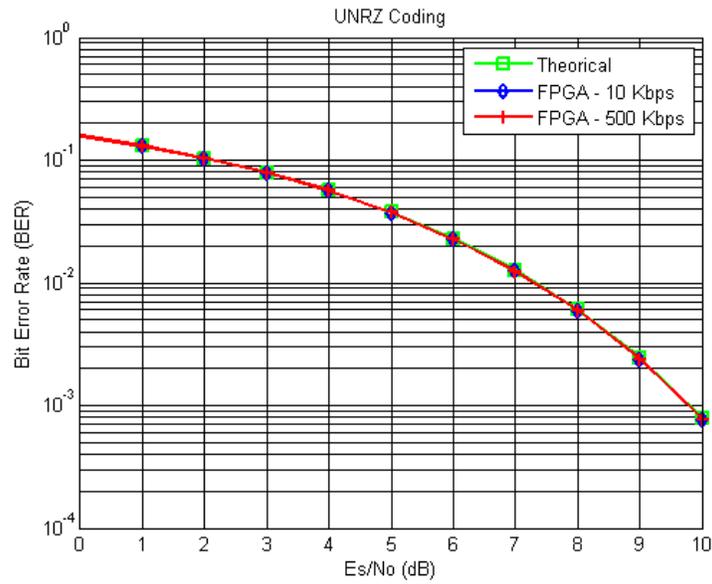


Figura 3.10. Curvas de desempeño para la codificación NRZ. Transmisión de 1048576 bits.

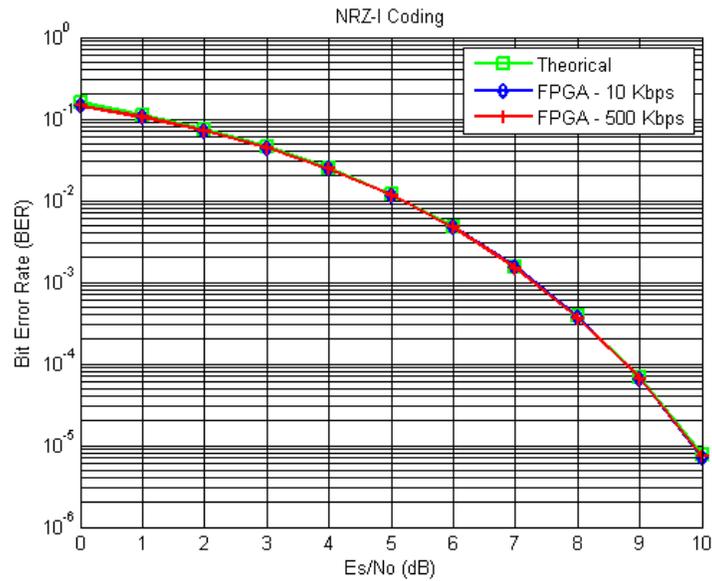


Figura 3.11. Curvas de desempeño para la codificación NRZ-I. Transmisión de 1048576 bits.

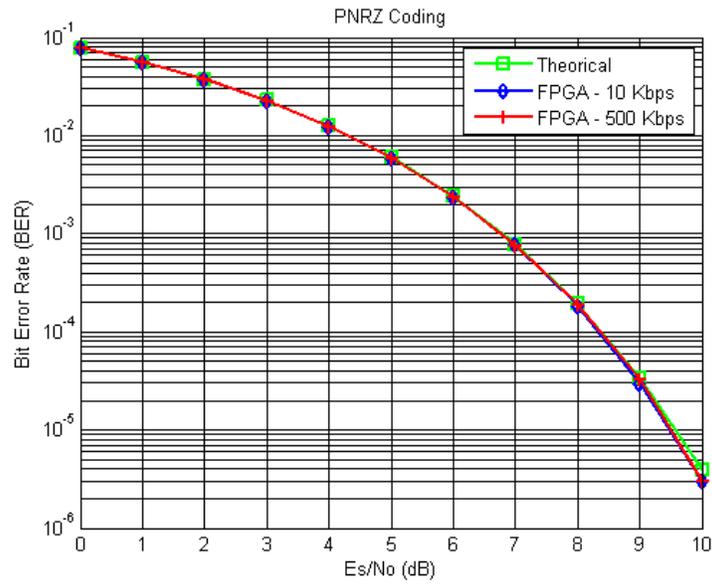


Figura 3.12. Curvas de desempeño para la codificación PNRZ. Transmisión de 1048576 bits.

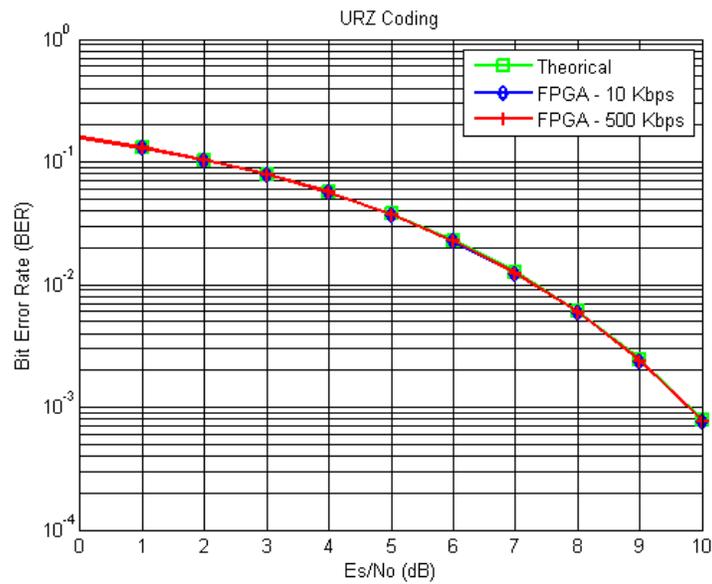


Figura 3.13. Curvas de desempeño para la codificación URZ. Transmisión de 1048576 bits.

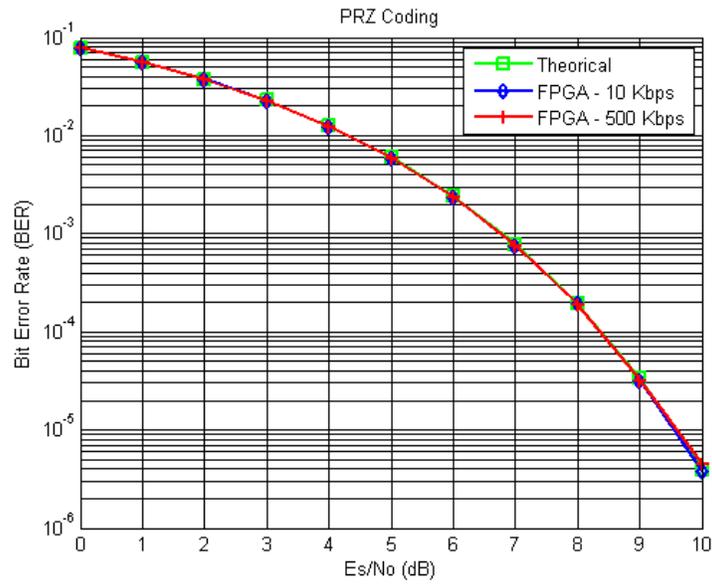


Figura 3.14. Curvas de desempeño para la codificación PRZ. Transmisión de 1048576 bits.

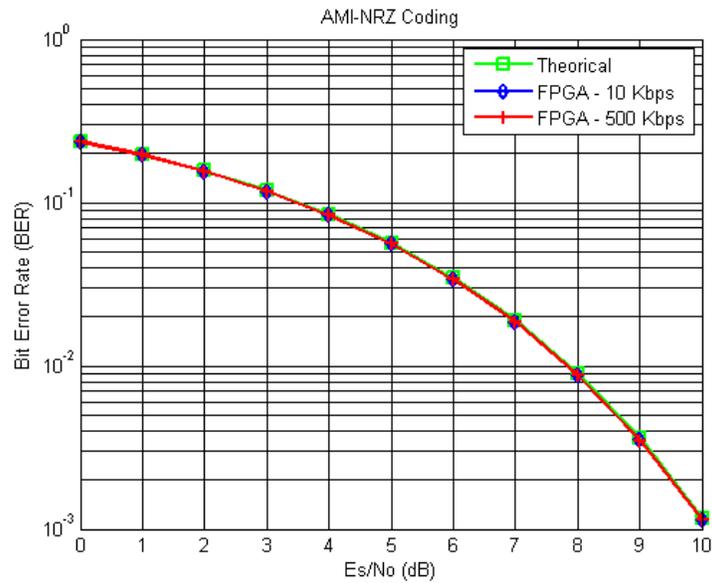


Figura 3.15. Curvas de desempeño para la codificación AMI-NRZ. Transmisión de 1048576 bits.

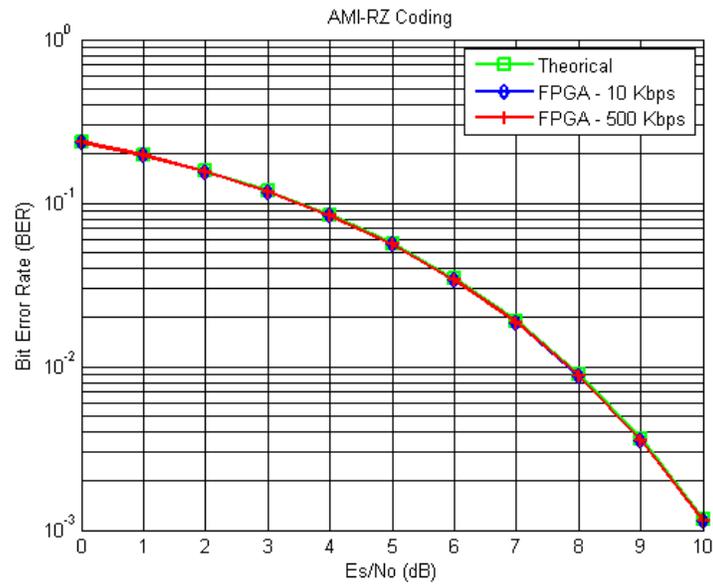


Figura 3.16. Curvas de desempeño para la codificación AMI-RZ. Transmisión de 1048576 bits.

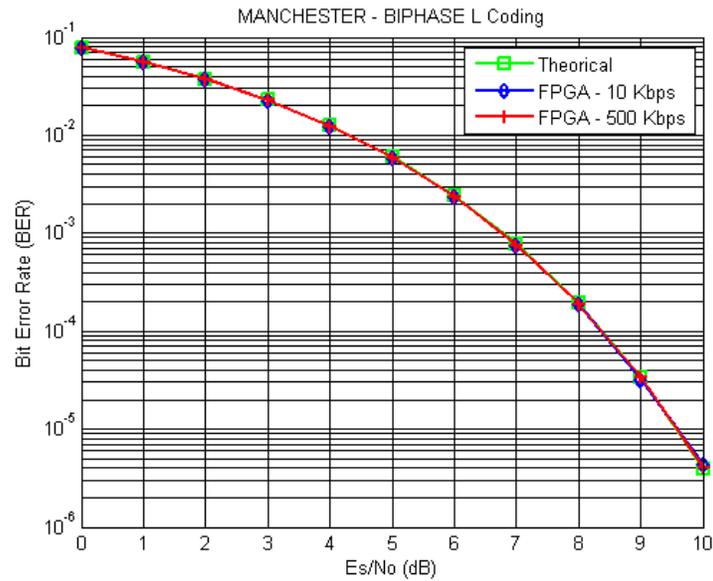


Figura 3.17. Curvas de desempeño para la codificación MANCHESTER OR BIPHASE L. Transmisión de 1048576 bits.

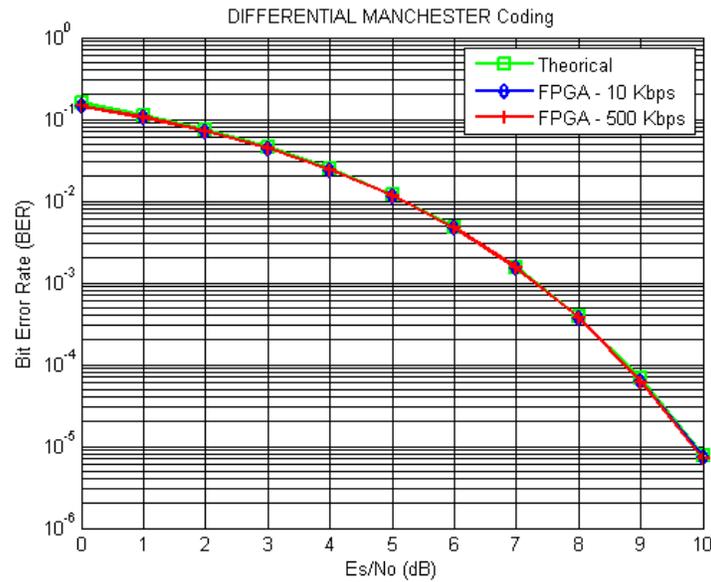


Figura 3.18. Curvas de desempeño para la codificación DIFFERENTIAL MANCHESTER. Transmisión de 1048576 bits.

Las figuras 3.10 a 3.18, ilustran las curvas de desempeño con una tendencia estadística más clara y coherente respecto a las curvas teóricas habiendo transmitido 1048576 bits. Por lo tanto, se concluye que una mayor cantidad de bits transmitidos implica una mayor precisión en el cálculo de la BER y aunque los sistemas de comunicaciones propuestos soportan la transmisión de una mayor cantidad de bits, los resultados obtenidos con 1048576 bits fueron suficientes para dar cumplimiento al objetivo general del presente trabajo de grado.

Finalmente cabe resaltar que aunque se haya variado la velocidad de transmisión en ambos escenarios los resultados fueron muy similares, con lo que se pudo verificar la independencia del sistema digital de comunicación con codificación de línea ante las variaciones de este parámetro, sin importar la cantidad de bits transmitidos, ya que no se tiene en consideración el efecto de multitrayectoria o corrimiento en frecuencia, sino solamente la distorsión del ruido aditivo Gaussiano introducida en el canal a la señal transmitida, por otro lado, no existe limitación en el ancho de banda del canal que pueda generar un comportamiento selectivo en frecuencia, de ahí que no se haya implementado una etapa de ecualización mediante filtros adaptados a la señal transmitida, con lo que finalmente la velocidad de transmisión de datos no dio lugar a



ISI, un factor determinante en la etapa de sincronización y decodificación de la señal recibida.

### 3.2.3. Análisis de desempeño y Eficiencia Espectral

En esta sección se realiza un análisis general del desempeño del sistema de comunicación digital con codificación de línea a nivel de BER y de Eficiencia Espectral.

Tabla 3.2. Eficiencia espectral de los 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

La tabla 3.2, muestra un resumen de la eficiencia espectral de cada esquema de codificación, analizadas en el capítulo uno. Por otro lado la figura 3.19, muestra la comparación de las curvas de desempeño del sistema de comunicación digital basado en hardware reconfigurable (FPGA), bajo diferentes esquemas de codificación de línea teniendo en consideración la transmisión de 1048576 bits, así como una velocidad de transmisión de datos igual a 500 Kbps, dado que es el escenario en donde se puede analizar de forma confiable las curvas de rendimiento obtenidas en cada implementación.

Como se estableció en el capítulo uno, el análisis en conjunto de los códigos de línea, toma como referencia la codificación UNRZ; de esta manera y en base a la observación de la figura 3.19 y la tabla 3.2, es preciso analizar cada codificación acorde a la clasificación propuesta de la siguiente manera:



**Códigos Unipolares:** Corresponden a los códigos UNRZ y URZ. Según la gráfica 3.19 estos representan un bajo desempeño con respecto a los códigos Polares y Diferenciales. Adicionalmente son tomados como referencia para la evaluación de las demás codificaciones, ya que usualmente son los más utilizados en sistemas de comunicación digital de bajo rendimiento y baja complejidad. Con respecto a la eficiencia espectral, la codificación URZ hace uso del doble de ancho de banda utilizado por el código UNRZ, obteniendo el mismo desempeño en ambos casos.

**Códigos Bipolares:** Conformados por los códigos AMI-NRZ y AMI-RZ. Aunque su desempeño es comparable con el desempeño del código UNRZ, estos códigos constituyen el peor de los casos, esto se debe 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. Con respecto a la tabla 3.2, se observa que ambos códigos tienen la misma eficiencia espectral que el código UNRZ, la diferencia radica en la duración del pulso que utilizan para codificar los datos binarios.

**Códigos Diferenciales:** Corresponden a los códigos NRZ-I y DIFFERENTIAL MANCHESTER. El desempeño de la codificación diferencial empieza a mejorar con respecto a los códigos Unipolares para valores de  $E_s/N_0$  superiores a  $1 \text{ dB}$ . Esto significa que en presencia de mucho ruido los códigos diferenciales presentan, a nivel de BER, un comportamiento similar que los códigos UNRZ. Por otra parte el código NRZ-I utiliza el mismo ancho de banda que el código UNRZ, sin embargo para el código DIFFERENTIAL MANCHESTER se requiere el uso del doble de ancho de banda utilizado por NRZ-I con el fin de transmitir la misma cantidad de información.

**Códigos Polares:** Conformado por los códigos PNRZ, PRZ y MANCHESTER OR BIPHASE L. Su desempeño mejora considerablemente con respecto a los códigos Unipolares aproximadamente en  $3 \text{ dB}$ . En cuanto a la eficiencia espectral la codificación PNRZ utiliza el mismo ancho de banda requerido por UNRZ, mientras que los códigos PRZ y MANCHESTER OR BIPHASE L necesitan el doble de ancho de



banda con respecto al código UNRZ, esto con el fin de transmitir la misma cantidad de información.

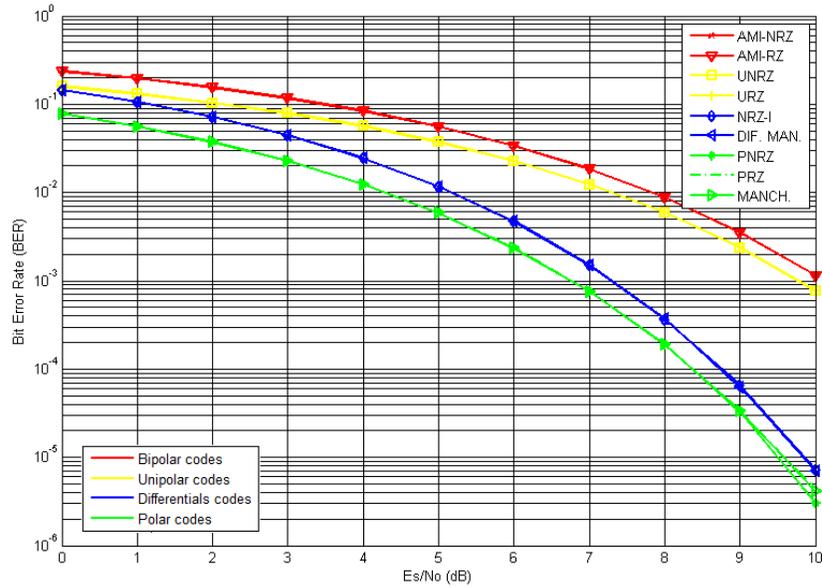


Figura 3.19. Comparación de curvas de desempeño para los diferentes esquemas de codificación implementados sobre hardware reconfigurable.



## CAPÍTULO 4

### CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

#### 4.1. CONCLUSIONES

- Las herramientas software y hardware utilizadas para la implementación de los códigos de línea fueron adecuadas en cuanto a capacidad y velocidad de procesamiento de datos.
- A nivel de diseño todas las codificaciones de línea desarrolladas representan la misma complejidad, puesto que los bloques ofrecidos por *System Generator* permitieron una reconfiguración de los parámetros de funcionamiento, sin la necesidad de rediseñar cada sistema por separado.
- La implementación sobre hardware reconfigurable (FPGA), permitió establecer apropiadamente un análisis completo del desempeño del sistema de comunicación digital con codificación de línea.
- El sistema digital de comunicación con codificación de línea, alcanzó resultados adecuados en la implementación sobre hardware reconfigurable (FPGA) dada la obtención de las curvas de desempeño acorde a las curvas teóricas.
- Una evaluación confiable de la BER para los diferentes esquemas de codificación de línea, depende proporcionalmente de la cantidad de bits transmitidos. Una mayor cantidad de bits transmitidos implica una mayor precisión en el cálculo de la BER.
- 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.



- Los sistemas de comunicación digital que implementan los códigos de línea PNRZ, PRZ y MANCHESTER requieren de la mitad de energía que el código UNRZ a fin de lograr una BER objetivo específica. Además presentan el mejor desempeño que todas las demás codificaciones, aunque su eficiencia espectral es menor que para los códigos sin retorno a cero.
- Los sistemas de comunicación digital que implementan los códigos de línea NRZ-I y DIFFERENTIAL MANCHESTER, presentan un desempeño similar que los códigos UNRZ para un nivel de  $E_s/N_0$  igual a 0 dB; sin embargo, para valores grandes de  $E_s/N_0$  tienen un desempeño similar a los códigos polares.

#### 4.2. RECOMENDACIONES

- El bloque *MCode* de *System Generator* no permite la programación de algoritmos complejos, sin embargo, se recomienda el uso de este bloque para la implementación de operaciones lógicas que permiten disminuir la utilización de bloques adicionales.
- Para la instalación de la herramienta software *System Generator*, se recomienda verificar la compatibilidad con la versión de *Matlab*, antes de configurar el entorno de desarrollo. Adicionalmente se sugiere que el nombre de usuario del computador no contenga caracteres especiales ni espacios. Ya que *System Generator* crea archivos temporales dentro de la carpeta de usuario y al momento de recuperar la información tiene en cuenta la restricción mencionada.
- Para un computador con sistema operativo Windows con arquitectura de 64 bits es necesario realizar una configuración extra para las herramientas *Project Navigator* e *iMPACT*, con el fin de permitir la generación del archivo .bit y su posterior programación en la FPGA. Por lo que se recomienda copiar el archivo *libPortabilityNOSH.dll* alojado en la carpeta de Xilinx en la dirección “directorio\_raiz\_xilinx\version\ISE\_DS\ISE\lib\nt64” por el archivo *libPortability.dll* ubicado en “directorio\_raiz\_xilinx \version\ISE\_DS\ISE\lib\nt64”.



- Al integrar la herramienta software *Matlab* con *System Generator* se recomienda ejecutar las aplicaciones bajo permisos administrativos del computador.
- Para trabajar con la aplicación de escritorio diseñada es necesario el uso de un conversor USB a serie.

#### 4.3. TRABAJOS FUTUROS

- Análisis del desempeño de un sistema de comunicación digital basado en hardware reconfigurable (FPGA) que implementa codificación multinivel.
- Análisis del desempeño de un sistema de comunicación digital basado en hardware reconfigurable (FPGA) sobre un canal AWGN limitado en banda que implementa la codificación de línea.
- Análisis del desempeño de un sistema de comunicación digital basado en hardware reconfigurable (FPGA) sobre un canal con multitrayecto que implementa la codificación de línea.
- Implementar un sistema de comunicación digital basado en hardware reconfigurable, utilizando dos placas de desarrollo que implementen la codificación de línea para el intercambio de datos sobre un canal de transmisión físico.





## BIBLIOGRAFÍA

- [1] M. E. Marin, J. K. Daza, "Análisis del desempeño de un sistema de comunicaciones con modulación MSK/FSK basado en hardware reconfigurable", Tesis de Pregrado, Dept. Telecomunicaciones, Universidad del Cauca, Popayán, Colombia, 2014.
- [2] E. Coimbra "Transmisión digital en banda base", [En línea], 2010. Disponible: [http://coimbraweb.com/documentos/digital/4.3\\_transmision\\_digital\\_bbase.pdf](http://coimbraweb.com/documentos/digital/4.3_transmision_digital_bbase.pdf) [Último acceso: 30 de Agosto de 2016].
- [3] J. A. Ruiz Cruz, "Tema IV. Comunicaciones digitales - Teoría de la Comunicación", [En línea], Escuela Politécnica Superior, Universidad Autónoma de Madrid, 2011, Disponible: [www.eps.uam.es/~tco](http://www.eps.uam.es/~tco) [Último acceso: 30 de Agosto de 2016].
- [4] J. E. Briceño M., "Transmisión de Datos", Dep. de Publicaciones Mérida, Venezuela, 2005.
- [5] A. Bonafonte, "Señales y Sistemas I", Dep. de Teoría de la Señal y Comunicaciones, Universidad Politécnica de Catalunya, España, 2009.
- [6] V. E. Checa, J. D. Velásquez, "Implementación de códigos de línea en una tarjeta de entrenamiento basada en un FPGA", Córdova, Quito Educador, Marzo 2011.
- [7] J. Luque, S. Clavijo, "Modulación de Señales Digitales", Dep. Tecnología Electrónica, Universidad de Sevilla, España, 2008.
- [8] R. G. Gallager, "Principles of Digital Communication", Massachusetts Institute of Technology, January 5, 2008.
- [9] A. Artés, F. Pérez, "Comunicaciones digitales", Departamento de Teoría de la Señal y Comunicaciones, Madrid 2012.
- [10] Procesamiento Digital de Señales, "Análisis espectral de señales analógicas", [En línea], Universidad Nacional del Sur, Bahía Blanca, Argentina, 2011, Disponible: [http://www.ingelec.uns.edu.ar/pds2803/pds2803\\_2012a.pdf](http://www.ingelec.uns.edu.ar/pds2803/pds2803_2012a.pdf) [Último acceso: 13 de Noviembre de 2015].
- [11] F. Troncoso, "Evaluación Y Aplicación de Estrategias para Control de Errores en Canales Satelitales Mediante Codificación Algebraica", Trabajo de grado, Departamento de Ingeniería Eléctrica, Universidad de Chile, Santiago de Chile, Agosto 2012.
- [12] J. Ramos, F. Díaz de María, D. Luengo, F. Pérez, "Transmisión Digital en Banda Base", [En línea], Departamento de Teoría de la Señal y Comunicaciones, Universidad



Carlos III de Madrid, Disponible: [www.tsc.uc3m.es/~hmolina/wp-content/uploads/2010/02/lsyc\\_ittst\\_p8.pdf](http://www.tsc.uc3m.es/~hmolina/wp-content/uploads/2010/02/lsyc_ittst_p8.pdf) [Último acceso: 30 de Agosto de 2016].

[13] G. Santamaria, A. Castejón, "Electrotecnia", 1 Edición, Editorial Editex S.A., 2009.

[14] J. A. Ruiz Cruz, "Tema II. Señales, Sistemas y Perturbaciones - Teoría de la Comunicación", [En línea], Escuela Politécnica Superior, Universidad Autónoma de Madrid, 2011, Disponible: [http://arantxa.ii.uam.es/~tco/Documentacion/Tema\\_II\\_2\\_Perturbaciones\\_en\\_los\\_sistemas\\_de\\_transmision\\_ver0.pdf](http://arantxa.ii.uam.es/~tco/Documentacion/Tema_II_2_Perturbaciones_en_los_sistemas_de_transmision_ver0.pdf) [Último acceso: 30 de Agosto de 2016].

[15] N. H. Avila, "Diseño Y Construcción de un Codec Didáctico para Transmisión Digital en Banda Base", Trabajo de grado, Escuela politécnica Nacional, Ecuador, Junio 1994.

[16] G. A. Smillie, "Analogue and Digital Communication Techniques", Newnes, ISBN 0-340-73125-7, 2002.

[17] J. G. Proakis, "Digital Communications", 4th Edition, McGraw-Hill Companies, 2000.

[18] Wikiversity, "Codificaciones digitales", [En línea], Consultado en Enero 2016, Disponible: [https://es.wikiversity.org/wiki/Codificaciones\\_digitales](https://es.wikiversity.org/wiki/Codificaciones_digitales) [Último acceso: 30 de Agosto de 2016].

[19] W. Stallings, "Comunicaciones y Redes de Computadores", 6th Edición, Prentice Hall, 1999.

[20] I. M. Zamora, "Transmisión Digital de Señales Banda Base", [En línea], Universidad Nacional de Ingeniería, Nicaragua 2009, Disponible: <http://es.slideshare.net/nica2009/lecture-9-codificacin-de-lnea-y-psd> [Último acceso: 30 de Agosto de 2016].

[21] J. López, E. Martos, "Señales aleatorias: Teoría y ejercicios resueltos", 1ª Edición, Marcombo, Oct 1, 2010

[22] L. Márquez, "Eficiencia Espectral", [En línea], Consultado en Mayo 2016, Disponible: <https://prezi.com/alqmemwbppw3/eficiencia-espectral/> [Último acceso: 30 de Agosto de 2016].

[23] 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.



- [24] J. D. Allen, D. Anderson, et all. "The Unicode® Standard", versión 8, [En línea], Disponible: <http://www.unicode.org/versions/Unicode8.0.0/UnicodeStandard-8.0.pdf>
- [25] Fuqin Xiong, "Digital Modulation Techniques", 2nd Edición, Artech House Inc., Boston London, 2006.
- [26] National Instrument, "Introducción a la Tecnología FPGA: Los Cinco Beneficios Principales", [En línea], Colombia, 2011. Disponible: <http://www.ni.com/white-paper/6984/es/>. [Último acceso: 30 de Agosto de 2016].
- [27] Arduino, "¿Que es Arduino?", [En línea]. Santiago de Chile. Disponible: <http://arduino.cl/que-es-arduino/>. [Último acceso: 30 de Agosto de 2016].
- [28] Mathworks, "Arduino Support from MATLAB", [En línea]. Estados Unidos, 2014. Disponible: <http://www.mathworks.com/hardware-support/arduino-matlab.html>. [Último acceso: 30 de Agosto de 2016].
- [29] Raspberry, "What is a Raspberry Pi?" [En línea]. Estados Unidos, 2014. Disponible: <https://www.raspberrypi.org/help/faqs/> [Último acceso: 30 de Agosto de 2016].
- [30] Xataka, [En línea]. Estados Unidos, 2014. Disponible: <http://www.xataka.com/tag/raspberry-pi> [Último acceso: 30 de Agosto de 2016].
- [31] MathWorks, "Raspberry Pi Support from MATLAB", [En línea]. Estados Unidos, 2014. Disponible: [http://es.mathworks.com/hardware-support/raspberry-pi-matlab.html?refresh=true&s\\_tid=gn\\_loc\\_drop](http://es.mathworks.com/hardware-support/raspberry-pi-matlab.html?refresh=true&s_tid=gn_loc_drop). [Último acceso: 30 de Agosto de 2016].
- [32] I. Domínguez. J. J. Murillo, "Laboratorio de Comunicaciones Digitales Radio Definida por Software" Dep. Teoría de la Señal y Comunicaciones, Universidad de Sevilla, España, 2011.
- [33] D. A. H. Aponte, "Introducción a GNU O a GNU Octave", [En línea]. España, 2007. Disponible: <http://softlibre.unizar.es/manuales/aplicaciones/octave/octave.pdf>. [Último acceso: 30 de Agosto de 2016].
- [34] "Open source software for numerical computation", [En línea]. Estados Unidos, 2015. Disponible: <http://www.scilab.org/scilab/about>. [Último acceso: 30 de Agosto de 2016].
- [35] MathWorks, "Matlab, The language of Technical Computing", [En línea]. Estados Unidos, 2014. Disponible: <http://www.mathworks.com/products/matlab/>. [Último acceso: 30 de Agosto de 2016].



[36] J. A. Muñoz, J. C. Zemanate, "Análisis del desempeño de un sistema de comunicaciones con modulación 16/64 QAM basado en hardware reconfigurable" Tesis de Pregrado, Dept. Telecomunicaciones, Universidad del Cauca, Popayán, Colombia, 2014.

[37] Xilinx All Programmable "Software Development", [En línea]. Estados Unidos, 2015. Disponible: <http://www.xilinx.com/products/design-tools/vivado.html#documentation>. [Último acceso: 30 de Agosto de 2016].

[38] A. C. Tobar, D. P. Sánchez, "Análisis del desempeño de un sistema de comunicaciones con codificación BCH binaria en hardware reconfigurable" Tesis de Pregrado, Dept. Telecomunicaciones, Universidad del Cauca, Popayán, Colombia, 2015.

[39] B. Zeidman, "The Universal Design Methodology taking hardware from conception through production", Design Feature, Diciembre, 2002.



## ANEXOS

### ANEXO A. ENERGÍA Y POTENCIA [9]

La energía o potencia de una señal cualquiera, se define de forma similar a la definición en términos eléctricos. Suponiendo una resistencia  $R$  con una tensión aplicada  $v(t)$ , por la que pasa una corriente  $i(t) = v(t)/R$ . La potencia instantánea disipada es

$$P(t) = \frac{|v(t)|^2}{R} \quad (\text{A.1})$$

o equivalentemente

$$P(t) = |i(t)|^2 * R. \quad (\text{A.2})$$

La energía total y la potencia media respectivamente se definen como las integrales

$$E = \int_{-\infty}^{\infty} P(t) dt \quad (\text{A.3})$$

$$P = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} P(t) dt. \quad (\text{A.4})$$

De una forma equivalente haciendo  $R = 1$  se define para cualquier señal  $x(t)$  la **Energía de una señal** como

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 dt \quad (\text{A.5})$$

asimismo, la **Potencia media** es

$$P_x = \lim_{T \rightarrow \infty} \frac{\text{Energía en } T}{T} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 dt. \quad (\text{A.6})$$

### TIPOS DE SEÑALES

Atendiendo a la energía o potencia de una señal, éstas se clasifican en

- Señales de energía finita (E.F.):  $0 < E_x < \infty$ .
- Señales de potencia media finita (P.M.F.):  $0 < P_x < \infty$ .
- Señales que no satisfacen ninguna de las propiedades anteriores y no son ni de E.F. ni de P.M.F.



Una señal de E.F. tiene potencia media nula y una señal de P.M.F. tiene energía infinita. Las señales de E.F. cumplen los criterios de convergencia de la transformada de Fourier. Dentro de este grupo están todas las señales de duración finita de interés en procesamiento de señal (variación acotada), y algunas de duración infinita. Dentro del grupo de señales de P.M.F. se encuentran las señales periódicas (variación acotada) señales de duración infinita cuya transformada de Fourier se obtienen por medio de funciones de distribución (escalón, rampa, etc.) y las correspondientes a procesos aleatorios.

Un grupo muy importante de señales de P.M.F. lo forman las señales periódicas. Una señal periódica de periodo  $T_0$  tiene la propiedad de que la integral en cualquier intervalo de duración  $T_0$  es constante:

$$\int_{t_0}^{t_0+T_0} X(t)dt = \int_0^{T_0} X(t)dt = \int_{\langle T_0 \rangle} x(t)dt \quad (A.7)$$

donde  $t_0$  es una constante arbitraria. Esta propiedad puede aplicarse para simplificar el cálculo de la Potencia media. Eligiendo el intervalo de integración  $T$  de la ecuación (6) múltiplo del periodo,  $T = MT_0$  la integral en el intervalo  $T$ , es  $M$  veces la integral en  $T_0$ , por lo que:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 dt = \lim_{M \rightarrow \infty} \frac{M}{MT_0} \int_{\langle T_0 \rangle} |x(t)|^2 dt \quad (A.8)$$

y el cálculo de la potencia media coincide con la potencia en un periodo

$$P_x = \frac{1}{T_0} \int_{T_0} |x(t)|^2 dt . \quad (A.9)$$

### TEOREMA DE PARSEVAL

Sean las señales de E.F.  $x(t)$  e  $y(t)$  con transformada de Fourier  $X(f)$  e  $Y(f)$  respectivamente. El teorema de Parseval para señales de E.F. establece que:

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df. \quad (A.10)$$

Demostración

$$\begin{aligned} E_x &= \int_{-\infty}^{\infty} x(t)x^*(t)dt \\ &= \int_{-\infty}^{\infty} x^*(t) \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} df dt \end{aligned}$$



$$\begin{aligned} &= \int_{-\infty}^{\infty} X(f) \left[ \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \right]^* df \\ &= \int_{-\infty}^{\infty} |X(f)|^2 df. \end{aligned} \quad (\text{A.11})$$

Por tanto, la energía de una señal puede calcularse conociendo el módulo de su transformada de Fourier.

### SEÑALES PERIÓDICAS. TEOREMA DE PARSEVAL

La expresión (10) no aplica para señales periódicas puesto que la primera integral no converge y el cuadrado de  $\delta(t)$  no está definido. Sea una señal periódica de periodo  $T_0$  de la forma

$$x(t) = \sum_{m=-\infty}^{\infty} c(m) e^{j2\pi mt/T_0} \quad (\text{A.12})$$

donde  $c(m)$  son los coeficientes de su desarrollo en serie. El teorema de Parseval establece que:

$$P_x = \frac{1}{T_0} \int_{\langle T_0 \rangle} |x(t)|^2 dt = \sum_{m=-\infty}^{\infty} c(m)^2. \quad (\text{A.13})$$

Demostración:

$$\begin{aligned} P_x &= \frac{1}{T_0} \int_{T_0} |x(t)|^2 dt \\ &= \frac{1}{T_0} \int_{T_0} x^*(t) \sum_{m=-\infty}^{\infty} c(m) e^{j2\pi mt/T_0} dt \\ &= \sum_{m=-\infty}^{\infty} c(m) \frac{1}{T_0} \int_{T_0} x^*(t) e^{j2\pi mt/T_0} dt \\ &= \sum_{m=-\infty}^{\infty} c(m) c^*(m) \\ &= \sum_{m=-\infty}^{\infty} |c(m)|^2. \end{aligned} \quad (\text{A.14})$$

De esta manera, la potencia de una señal periódica es igual a la suma de las amplitudes al cuadrado de las componentes armónicas de la señal. Sólo se requiere información del módulo.



## SEÑALES PERIÓDICAS. DENSIDAD ESPECTRAL DE POTENCIA

La densidad espectral de potencia de una señal periódica  $x(t)$  definida en (12), está formada por impulsos:

$$S_{xx}(f) = \sum_{m=-\infty}^{\infty} |c(m)|^2 \delta(f - mf_0) . \quad (\text{A.15})$$

Donde  $f_0 = 1/T_0$ . Al integrar esta función sobre el eje de frecuencia, se obtiene la potencia de la señal como se puede comprobar mediante el teorema de Parseval enunciado en (13). Nótese que la densidad espectral de potencia no es  $|X(f)|^2$ .

## CORRELACIÓN Y DENSIDAD ESPECTRAL DE POTENCIA

Considerando una función de potencia media finita (PMF), se define su función de correlación mediante:

$$R_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t + \tau) y^*(t) dt . \quad (\text{A.16})$$

Además se puede demostrar que el máximo de la función de autocorrelación  $R_{xx}(\tau)$  está en el origen:

$$|R_{xx}(\tau)| \leq R_{xx}(0) . \quad (\text{A.17})$$

También que la autocorrelación es una función Hermítica:

$$R_{xx}(\tau) = R_{xx}^*(-\tau) . \quad (\text{A.18})$$

Si  $x(t)$  es real, su autocorrelación es real y par:

$$R_{xx}(\tau) = R_{xx}(-\tau) . \quad (\text{A.19})$$

La potencia de una señal es el valor de su autocorrelación en el origen:

$$R_{xx}(0) = P_x . \quad (\text{A.20})$$



## DENSIDAD ESPECTRAL DE POTENCIA. TEOREMA DE WIENER KINTCHINE

Se define la densidad espectral de potencia de una señal  $x(t)$  como:

$$S_{xx}(f) = \lim_{T \rightarrow \infty} \frac{1}{T} R_{x_T x_T}(f) \quad (\text{A.21})$$

siendo

$$x_T(t) = x(t) \Pi\left(\frac{t}{T}\right). \quad (\text{A.22})$$

El teorema de Wiener Kintchine establece que la transformada de Fourier de la autocorrelación es la Densidad espectral de potencia:

$$F[R_{xx}(\tau)] = S_{xx}(f). \quad (\text{A.23})$$

Para comprobarlo, basta calcular la correlación de la señal  $x_T(t)$  y comprobar que:

$$R_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} R_{x_T x_T}(\tau). \quad (\text{A.24})$$

Tomando transformada de Fourier en ambos miembros de la ecuación y aplicando que  $x_T(t)$  es una señal de E.F. finalmente se tiene que:

$$\begin{aligned} F[R_{xx}(\tau)] &= F\left[\lim_{T \rightarrow \infty} \frac{1}{T} R_{x_T x_T}(\tau)\right] \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} G_{x_T x_T}(f) \\ &= S_{xx}(f). \end{aligned} \quad (\text{A.25})$$

## RELACIÓN ENTRE CORRELACIÓN Y DENSIDAD ESPECTRAL DE POTENCIA EN SISTEMAS LINEALES E INVARIANTES EN EL TIEMPO.

Sea un sistema Lineal en Invariante en el Tiempo (LTI) caracterizado por  $h(t)$ . La salida  $y(t)$  cuando a su entrada se aplica  $x(t)$  puede expresarse por la ecuación de convolución:

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(t-t')h(t')dt'. \quad (\text{A.26})$$

Las relaciones entre las correlaciones de la entrada y la salida son idénticas a las establecidas para señales de E.F:



$$R_{xy}(\tau) = R_{xx}(\tau) * h^*(-\tau) \quad (\text{A.27})$$

$$R_{yx}(\tau) = R_{xx}(\tau) * h(\tau) \quad (\text{A.28})$$

$$R_{yy}(\tau) = R_{xx}(\tau) * h(\tau) * h^*(-\tau). \quad (\text{A.29})$$

Tomando transformada de Fourier se verifican las siguientes propiedades:

$$S_{xy}(f) = S_{xx}(f)H^*(f) \quad (\text{A.30})$$

$$S_{yx}(f) = S_{xx}(f)H(f) \quad (\text{A.31})$$

$$S_{yy}(f) = S_{xx}(f)|H(f)|^2. \quad (\text{A.32})$$



## ANEXO B. TASA DE ERROR DE BIT DE LOS CÓDIGOS DE LÍNEA [25]

Un código de línea binario consta de dos tipos de señales:

$$H_1 : s_1(t), \quad 0 \leq t \leq T, \text{ Se transmite con una probabilidad de } p_1.$$

$$H_2 : s_2(t), \quad 0 \leq t \leq T, \text{ Se transmite con una probabilidad de } p_2.$$

Donde  $p_1$  y  $p_2$  son llamados probabilidad *a priori*. La energía de las dos señales es:

$$E_1 = \int_0^T s_1^2(t) dt$$

y

$$E_2 = \int_0^T s_2^2(t) dt.$$

En general, estas dos señales pueden ser correlacionadas. Por lo tanto se define el coeficiente de correlación de  $s_1(t)$  y  $s_2(t)$  mediante

$$\rho_{12} = \frac{1}{\sqrt{E_1 E_2}} \int_0^T s_1(t) s_2(t) dt. \quad (\text{B.1})$$

Donde  $|\rho_{12}| \leq 1$ . De esta manera la señal binaria a la entrada del receptor es:

$$r(t) = s_i(t) + n(t), \quad i = 1, 2.$$

Donde  $n(t)$  es la señal de ruido AWGN con media cero y densidad espectral  $N_0/2$ .

Un receptor óptimo para  $r(t)$ , consiste en un correlador o un filtro adaptado a la señal diferencia  $s_d(t)$ .

$$s_d(t) = s_1(t) - s_2(t).$$



Por el *criterio de probabilidad de error mínima*, siendo  $p_1 = p_2$  el umbral de decisión en el receptor se define como

$$\gamma_0 = \frac{E_1 - E_2}{2}. \quad (\text{B.2})$$

Y la probabilidad de error de bit está dada por

$$P_b = Q\left(\sqrt{\frac{E_d}{2N_0}}\right), \quad (\text{B.3})$$

donde  $Q$  es la función error (también conocida como función error de Gauss) [25].  $E_d$  es la energía de las señales  $s_1(t)$  y  $s_2(t)$ .

$$E_d = \int_0^T s_f^2(t) dt$$

$$E_d = E_1 + E_2 - 2\rho_{12}\sqrt{E_1E_2}$$

La ecuación B.3 muestra que cuanto mayor sea la distancia ( $E_d$ ) entre las dos señales  $s_1(t)$  y  $s_2(t)$ ,  $P_b$  disminuye. Esto es intuitivamente convincente, ya que cuanto mayor sea esta distancia, es más fácil para el detector distinguir las señales presentes a su entrada. Expresando  $P_b$  en términos de la energía de cada señal, se tiene que

$$P_b = Q\left(\sqrt{\frac{E_1 + E_2 - 2\rho_{12}\sqrt{E_1E_2}}{2N_0}}\right) \quad (\text{B.4})$$

Esta expresión indica que  $P_b$  no sólo depende de la energía de cada señal, sino también en la correlación entre ellas. Cuando  $\rho_{12} = -1$ ,  $P_b$  es mínima. Las señales binarias con  $\rho_{12} = -1$  son llamadas antipodales. Por otro lado cuando  $\rho_{12} = 0$ , las señales  $s_1(t)$  y  $s_2(t)$  son ortogonales.



## B.1. BER PARA CÓDIGOS SIN RETORNO A CERO

### PNRZ

$$s_1(t) = A, \quad 0 \leq t \leq T$$

$$s_2(t) = -A, \quad 0 \leq t \leq T$$

$$s_d(t) = 2A, \quad 0 \leq t \leq T$$

En este caso  $\rho_{12} = -1$ ,  $E_1 = E_2 = A^2T$ , y  $E_b = (E_1 + E_2)/2 = A^2T$ . A partir de la ecuación B.3  $P_b$  es:

$$P_b = Q\left(\sqrt{\frac{2A^2T}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (\text{B.5})$$

y el umbral de decisión óptimo es:

$$\gamma_0 = \frac{E_1 - E_2}{2} = 0.$$

### UNRZ

$$s_1(t) = A, \quad 0 \leq t \leq T$$

$$s_2(t) = 0, \quad 0 \leq t \leq T$$

$$s_d(t) = A, \quad 0 \leq t \leq T$$

Así  $E_1 = A^2T$ ,  $E_2 = 0$ ,  $E_b = A^2T/2$  y  $\rho_{12} = 0$  y la probabilidad de error es:

$$P_b = Q\left(\sqrt{\frac{A^2T}{2N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right). \quad (\text{B.6})$$



Para el umbral de decisión óptimo se tiene que

$$\gamma_0 = \frac{A^2 T}{2}.$$

## B.2. BER PARA CÓDIGOS CON RETORNO A CERO

PRZ

$$s_1(t) = \begin{cases} A, & 0 \leq t \leq \frac{T}{2} \\ 0, & \text{en otro caso} \end{cases}$$

$$s_2(t) = -s_1(t)$$

$$s_d(t) = \begin{cases} 2A, & 0 \leq t \leq \frac{T}{2} \\ 0, & \text{en otro caso} \end{cases}$$

Por lo tanto  $\rho_{12} = -1$ ,  $E_1 = E_2 = A^2 T/2$ , y  $E_b = (E_1 + E_2)/2 = A^2 T/2$ . De B.3  $P_b$  es:

$$P_b = Q\left(\sqrt{\frac{A^2 T}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \quad (\text{B.7})$$

El umbral de decisión óptimo es:

$$\gamma_0 = \frac{E_1 - E_2}{2} = 0$$



## URZ

$$s_1(t) = \begin{cases} A, & 0 \leq t \leq \frac{T}{2} \\ 0, & \text{en otro caso} \end{cases}$$

$$s_2(t) = 0, \quad 0 \leq t \leq T$$

$$s_d(t) = \begin{cases} A, & 0 \leq t \leq \frac{T}{2} \\ 0, & \text{en otro parte} \end{cases}$$

Para este caso  $\rho_{12} = 0$ ,  $E_2 = A^2T/2$ ,  $E_1 = 0$  y  $E_b = (E_1 + E_2)/2 = A^2T/4$ . De la ecuación B.3 se tiene que:

$$P_b = Q\left(\sqrt{\frac{A^2T}{4N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right). \quad (\text{B.8})$$

El umbral de decisión óptimo es:

$$\gamma_0 = \frac{A^2T}{4}.$$

### B.3. BER PARA CÓDIGOS PSEUDOTERNARIOS

#### AMI-NRZ

Este código se compone de tres tipos de señales:

$$H_1 : s_1(t) = A, \quad 0 \leq t \leq T, \quad p_1 = \frac{1}{4}$$

$$H_2 : s_2(t) = -A, \quad 0 \leq t \leq T, \quad p_2 = \frac{1}{4}$$

$$H_3 : s_3(t) = 0, \quad 0 \leq t \leq T, \quad p_3 = \frac{1}{2}$$

En este caso existen dos umbrales y tres regiones de decisión como se muestra en la figura B.1. El receptor óptimo consiste de un correlador y un detector de umbral con dos umbrales, como se muestra en la Figura B.1.a.

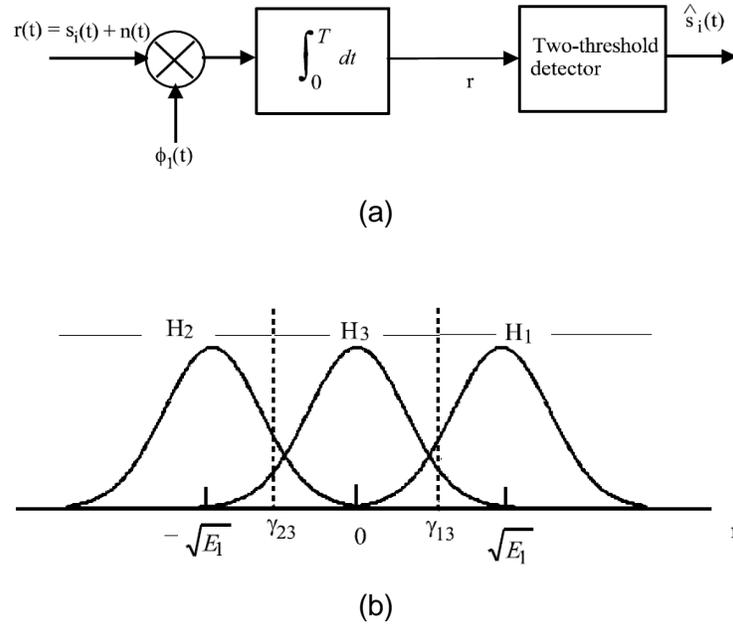


Figura B.1. Receptor óptimo (a). Regiones de decisión (b), para las señales de AMI.

Si se define a  $Pr(e/S_i)$  para indicar la probabilidad de error cuando la señal  $S_i(t)$  se transmite, la probabilidad media de error de bit se puede calcular como:

$$P_b = p_1 Pr(e/S_1) + p_2 Pr(e/S_2) + p_3 Pr(e/S_3)$$

$$P_b = p_1 \int_{-\infty}^{\gamma_{13}} p(r/S_1) dr + p_2 \int_{\gamma_{23}}^{\infty} p(r/S_2) dr + 2p_3 \int_{\gamma_{13}}^{\infty} p(r/S_3) dr$$

$$P_b = 2p_1 \int_{-\infty}^{\gamma_{13}} p(r/S_1) dr + 2p_3 \int_{\gamma_{13}}^{\infty} p(r/S_3) dr$$

$$P_b = \frac{1}{2} Q\left(\frac{E_1 - N_0 \ln 2}{\sqrt{2N_0 E_1}}\right) + Q\left(\frac{E_1 + N_0 \ln 2}{\sqrt{2N_0 E_1}}\right). \quad (\text{B.9})$$



Dado  $E_1 = A^2T$  y  $E_b = 0.5E_1$ , se tiene que:

$$P_b = \frac{1}{2}Q\left(\frac{A^2T - N_0 \ln 2}{A\sqrt{2N_0T}}\right) + Q\left(\frac{A^2T + N_0 \ln 2}{A\sqrt{2N_0T}}\right)$$
$$P_b = \frac{1}{2}Q\left(\frac{2E_b/N_0 - \ln 2}{2\sqrt{E_b/N_0}}\right) + Q\left(\frac{2E_b/N_0 + \ln 2}{2\sqrt{E_b/N_0}}\right). \quad (\text{B.10})$$

Esta es la probabilidad de error de bit para una recepción óptima. Sin embargo, para una relación señal-ruido alta  $N_0 \ln 2 \ll E_1$ , y los umbrales se pueden ajustar a la mitad  $\gamma_{13} \approx \sqrt{E_1}/2$  y  $\gamma_{23} \approx -\sqrt{E_1}/2$ . Así, la expresión B.10 se reduce a

$$P_b \approx \frac{3}{2}Q\left(\sqrt{\frac{E_b}{N_0}}\right). \quad (\text{B.11})$$

### AMI-RZ

Cuya  $E_1 = A^2T/2$  y  $E_b = A^2T/4$ . La BER se puede encontrar a partir de B.9.

$$P_b = \frac{1}{2}Q\left(\frac{A^2T/2 - N_0 \ln 2}{A\sqrt{N_0T}}\right) + Q\left(\frac{A^2T/2 + N_0 \ln 2}{A\sqrt{N_0T}}\right)$$
$$P_b = \frac{1}{2}Q\left(\frac{2E_b/N_0 - \ln 2}{2\sqrt{E_b/N_0}}\right) + Q\left(\frac{2E_b/N_0 + \ln 2}{2\sqrt{E_b/N_0}}\right). \quad (\text{B.12})$$

Tomando las mismas consideraciones que para AMI-NRZ, finalmente se tiene que

$$P_b \approx \frac{3}{2}Q\left(\sqrt{\frac{E_b}{N_0}}\right). \quad (\text{B.13})$$



## B.4. BER PARA CÓDIGOS BIFASE

### MANCHESTER

$$s_1(t) = \begin{cases} A, & 0 \leq t \leq \frac{T}{2} \\ -A, & \frac{T}{2} \leq t \leq T \end{cases}$$

$$s_2(t) = -s_1(t).$$

El coeficiente de correlación entre las señales  $s_1(t)$  y  $s_2(t)$  es  $\rho_{12} = -1$ . La energía de cada señal es

$$E_1 = E_2 = A^2T = E_b,$$

por lo que la señal diferencia corresponde a

$$E_d = E_1 + E_2 - 2\rho_{12}\sqrt{E_1E_2}$$

$$E_d = 4A^2T = 4E_b$$

Así, la probabilidad de error de bit es

$$P_b = Q\left(\sqrt{\frac{2A^2T}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \quad (\text{B.14})$$

## B.5. BER PARA CÓDIGOS DIFERENCIALES

### NRZ-I

Ya que se trata de un código polar la deducción de la BER es similar que para el código PNRZ, sin embargo, el proceso de decodificación que se lleva a cabo después de la de



detección, compara el bit presente con el bit anterior, eso conlleva a definir la probabilidad de error de la siguiente manera

$$P'_b = \Pr(\text{bit presente correcto y bit anterior incorrecto}) \\ + \Pr(\text{bit presente incorrecto y bit anterior correcto})$$

$$P'_b = (1 - P_b)P_b + P_b(1 - P_b) \\ = 2(1 - P_b)P_b \\ \approx 2P_b,$$

Si se tiene que  $P_b$  es muy pequeña entonces  $P'_b \approx 2P_b$ , es decir

$$P_b \approx 2Q\left(\sqrt{\frac{2A^2T}{N_0}}\right) = 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \quad (\text{B.15})$$

## MANCHESTER DIFERENCIAL

Tomando el mismo análisis que para NRZ-I, con referencia al código Manchester se tiene que la probabilidad de error para este código es

$$P_b \approx 2Q\left(\sqrt{\frac{2A^2T}{N_0}}\right) = 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \quad (\text{B.16})$$