

**ANÁLISIS DEL DESEMPEÑO DE UN SISTEMA DE COMUNICACIONES CON
MODULACIÓN 16/64 QAM BASADO EN HARDWARE RECONFIGURABLE**



**Julián Andrés Muñoz Hidalgo
Juan Camilo Zemanate Zuñiga**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELECOMUNICACIONES
GRUPO DE RADIO E INALÁMBRICAS-GRIAL
GRUPO I+D NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES-GNTT
POPAYÁN, 2014**

**ANÁLISIS DEL DESEMPEÑO DE UN SISTEMA DE COMUNICACIONES CON
MODULACIÓN 16/64 QAM BASADO EN HARDWARE RECONFIGURABLE**



**Julián Andrés Muñoz Hidalgo
Juan Camilo Zemanate Zuñiga**

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

**Director
Víctor Manuel Quintero Flórez**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELECOMUNICACIONES
GRUPO DE RADIO E INALÁMBRICAS-GRIAL
GRUPO I+D NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES-GNTT
POPAYÁN, 2014**

AGRADECIMIENTOS

A Dios porque sin él nada habría sido posible.

A nuestras familias por haber sido un apoyo incondicional e invaluable a lo largo de todo este proceso.

A nuestras novias y amigos por su compañía en el día a día.

A nuestro director Ing. Victor Manuel Quintero por su abnegada labor.

Al Ing. Jesús Mauricio Ramírez por sus valiosos consejos.

A todos ustedes, ¡infinitas gracias!

TABLA DE CONTENIDO

Pág.

INTRODUCCIÓN

1. GENERALIDADES	1
1.1 SISTEMAS DE COMUNICACIONES DIGITALES.....	1
1.1.1 Componentes básicos de un sistema de comunicaciones digital	2
1.1.2 Modulación Digital.....	4
1.1.2.1 Modulación QAM	5
1.1.2.2 Sistema básico de comunicaciones banda base con modulación m-QAM.	6
1.2 HERRAMIENTAS HARDWARE Y SOFTWARE	16
1.2.1 Computación reconfigurable.	16
1.2.1.1 Tecnologías de hardware reconfigurable.	16
1.2.1.2 Herramientas software.	18
2. MODELADO, SIMULACIÓN E IMPLEMENTACIÓN	20
2.1 METODOLOGÍA DE TRABAJO	20
2.1.1 Análisis de requerimientos.	20
2.1.2 Modelado.	21
2.1.2.1 Modelo de referencia.	21
2.1.3 Simulación.	29
2.1.3.1 Selección de herramientas.....	30
2.1.3.2 Elementos utilizados en el modelo en System Generator.	30
2.1.3.3 Modelo en System Generator.	40
2.1.3.4 Validación del modelo en System Generator.	84
2.1.4 Implementación.	85
2.1.4.1 Generación del Archivo de Programación Bitstream (.bit).	86
2.1.4.2 Programación del hardware reconfigurable (FPGA).	88
3. PLAN DE PRUEBAS, RESULTADOS Y ANÁLISIS	89
3.1 PLAN DE PRUEBAS	89
3.2 RESULTADOS	90
3.2.1 Prueba 1. Variación de la velocidad de transmisión de datos.....	90
3.2.2 Prueba 2. Variación del tipo de mapeo.	91
3.2.3 Prueba 3. Variación de la cantidad de bits transmitidos.	94
3.3 ANÁLISIS	95
3.3.1 Señales de entrada y salida del sistema básico de comunicaciones banda base con modulación 16/64 QAM.	95
3.3.2 Prueba 1. Variación de la velocidad de transmisión de datos.....	95
3.3.3 Prueba 2. Variacion del tipo de Mapeo.	96
3.3.4 Prueba 3. Variacion de la cantidad de bits transmitidos.	96
3.3.5 Curvas de desempeño.	98
4. CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.....	102
4.1 CONCLUSIONES	102
4.2 RECOMENDACIONES.....	102

4.3 TRABAJOS FUTUROS..... 103

BIBLIOGRAFÍA

APÉNDICES

LISTADO DE TABLAS

	Pág.
Tabla 2.1. Representación de los bits de entrada al divisor de datos.....	23
Tabla 2.2. Tabla de verdad del divisor de datos.....	45
Tabla 2.3. Mapeo Binario y Gray, modulación 16 QAM.....	52
Tabla 2.4. Mapeo Binario y Gray, modulación 64 QAM.....	53
Tabla 2.5. Umbrales de decisión Mapeo Binario y Gray, modulación 16 QAM.....	68
Tabla 2.6. Umbrales de decisión Mapeo Binario y Gray, modulación 64 QAM.....	68
Tabla 2.7. Demapeo Binario y Gray, modulación 16 QAM.....	71
Tabla 2.8. Demapeo Binario y Gray, modulación 64 QAM.....	71
Tabla 2.9. Recursos requeridos por el sistema básico de comunicaciones banda base con modulación 16QAM	87
Tabla 2.10. Recursos requeridos por el sistema básico de comunicaciones banda base con modulación 64QAM.....	87
Tabla 3.1. Plan de pruebas.....	89

LISTADO DE FIGURAS

	Pág.
Figura 1.1. Diagrama en bloques sistema de comunicaciones general.....	1
Figura 1.2. Diagrama en bloques sistema de comunicaciones digital	2
Figura 1.3. Etapas de modulación y demodulación.....	3
Figura 1.4. Translación lineal espectral.....	4
Figura 1.5. Sistema básico de comunicaciones banda base con modulación m-QAM	6
Figura 1.6. Diagramas de constelación, modulaciones 16 y 64 QAM, mapeo Binario	8
Figura 1.7. Diagramas de constelación, modulaciones 16 y 64 QAM, mapeo Gray	8
Figura 1.8. Regiones de decisión de una constelación 16-QAM	12
Figura 1.9. Conformación del Hardware Reconfigurable.....	16
Figura 1.10. Primer modelo de Raspberry para Simulink®	18
Figura 2.1. Modelo Lineal Secuencial	20
Figura 2.2. Modelo de referencia	22
Figura 2.3. Fuente de información	22
Figura 2.4. Divisor de datos, modulación 16 QAM	23
Figura 2.5. Divisor de datos, modulación 64 QAM	23
Figura 2.6. Conversor de 2 a M niveles, modulación 16 QAM.....	24
Figura 2.7. Conversor de 2 a M niveles, modulación 64 QAM.....	24
Figura 2.8. Normalizador de potencia, modulación 16 QAM	25
Figura 2.9. Normalizador de potencia, modulación 64 QAM	25
Figura 2.10. Canal AWGN complejo, modulación 16 QAM	25
Figura 2.11. Canal AWGN complejo, modulación 64 QAM	26
Figura 2.12. Desnormalizador de potencia, modulación 16 QAM.....	26
Figura 2.13. Desnormalizador de potencia, modulación 64 QAM.....	26
Figura 2.14. Decisor, modulación 16 QAM.....	27
Figura 2.15. Decisor, modulación 64 QAM.....	27
Figura 2.16. Conversor de M a 2 niveles, modulación 16 QAM.....	28
Figura 2.17. Conversor de M a 2 niveles, modulación 64 QAM.....	28
Figura 2.18. Multiplexor de datos, modulación 16 QAM	29
Figura 2.19. Multiplexor de datos, modulación 64 QAM	29
Figura 2.20. Calculador de BER	29
Figura 2.21. System Generator Token	32
Figura 2.22. Elemento <i>LFSR</i>	32
Figura 2.23. Elemento <i>Convert</i>	33
Figura 2.24. Elemento <i>Counter</i>	33
Figura 2.25. Elemento <i>Mux</i>	34
Figura 2.26. Elemento <i>Delay</i>	34
Figura 2.27. Elemento <i>Serial to Parallel</i>	34
Figura 2.28. Elemento <i>Down Sample</i>	35
Figura 2.29. Elemento <i>Mcode</i>	35
Figura 2.30. Elemento <i>Constant</i>	36
Figura 2.31. Elemento <i>Mult</i>	36
Figura 2.32. Elemento <i>WGNG</i>	37
Figura 2.33. Elemento <i>AddSub</i>	37
Figura 2.34. Elemento <i>Up Sample</i>	37

Figura 2.35. Elemento <i>Parallel to Serial</i>	38
Figura 2.36. Elemento <i>Accumulator</i>	38
Figura 2.37. Elemento <i>Relational</i>	39
Figura 2.38. Elemento <i>Dual Port Ram</i>	39
Figura 2.39. Elemento <i>Gateway In</i>	39
Figura 2.40. Elemento <i>Gateway Out</i>	40
Figura 2.41. Modelo en System Generator.....	41
Figura 2.42. Fuente de información y transmisor, modelo en <i>System Generator</i>	42
Figura 2.43. Canal de comunicaciones, modelo en <i>System Generator</i>	42
Figura 2.44. Receptor, modelo en System Generator	43
Figura 2.45. Calculador de BER, modelo en System Generator	43
Figura 2.46. Fuente de información	44
Figura 2.47. Secuencia de bits predefinida, modulación 16 QAM	44
Figura 2.48. Secuencia de bits predefinida, modulación 64 QAM	44
Figura 2.49. Divisor de datos	45
Figura 2.50. Señal de entrada y señales de salida de las ramas I y Q del divisor de datos, modulación 16 QAM	46
Figura 2.51. Señal de entrada y señales de salida de las ramas I y Q del divisor de datos, modulación 64 QAM	46
Figura 2.52. Conversor de 2 a M niveles.....	47
Figura 2.53. Señales de las ramas I y Q sincronizadas, modulación 16 QAM.....	47
Figura 2.54. Señales de las ramas I y Q sincronizadas, modulación 64 QAM.....	48
Figura 2.55. Señales de entrada y salida del <i>Serial to Parallel</i> de la rama I, modulación 16 QAM.....	48
Figura 2.56. Señales de entrada y salida del <i>Serial to Parallel</i> de la rama Q, modulación 16 QAM.....	49
Figura 2.57. Señales de entrada y salida del <i>Serial to Parallel</i> de la rama I, modulación 64 QAM.....	49
Figura 2.58. Señales de entrada y salida del <i>Serial to Parallel</i> de la rama Q, modulación 64 QAM.....	50
Figura 2.59. Señales de entrada y salida del <i>Down Sample</i> de la rama I, modulación 16 QAM.....	50
Figura 2.60. Señales de entrada y salida del <i>Down Sample</i> de la rama Q, modulación 16 QAM.....	51
Figura 2.61. Señales de entrada y salida del <i>Down Sample</i> de la rama I, modulación 64 QAM.....	51
Figura 2.62. Señales de entrada y salida del <i>Down Sample</i> de la rama Q, modulación 64 QAM.....	52
Figura 2.63. Señales de entrada y salida del <i>Mcode</i> de la rama I, mapeo Gray, modulación 16 QAM.....	53
Figura 2.64. Señales de entrada y salida del <i>Mcode</i> de la rama Q, mapeo Gray, modulación 16 QAM	54
Figura 2.65. Señal de entrada y salida del <i>Mcode</i> de la rama I, mapeo Gray, modulación 64 QAM.....	54
Figura 2.66. Señales de entrada y salida del <i>Mcode</i> de la rama Q, mapeo Gray, modulación 64 QAM	55
Figura 2.67. Normalizador de potencia	55
Figura 2.68. Señales de entrada y salida del normalizador de potencia de la rama I, modulación 16 QAM	56

Figura 2.69. Señales de entrada y salida del normalizador de potencia de la rama Q, modulación 16 QAM	56
Figura 2.70. Señales de entrada y salida del normalizador de potencia de la rama I, modulación 64 QAM	57
Figura 2.71. Señales de entrada y salida del normalizador de potencia de la rama Q, modulación 64 QAM	57
Figura 2.72. Canal AWGN complejo	58
Figura 2.73. Diagrama de constelación señal modulada después del canal - semillas iguales.....	59
Figura 2.74. Diagrama de constelación señal modulada después del canal - semillas diferentes.....	59
Figura 2.75. Señales de entrada y salida del canal AWGN de la rama I, modulación 16 QAM.....	64
Figura 2.76. Señales de entrada y salida del canal AWGN de la rama Q, modulación 16 QAM.....	64
Figura 2.77. Señales de entrada y salida del canal AWGN de la rama I, modulación 64 QAM.....	65
Figura 2.78. Señales de entrada y salida del canal AWGN de la rama Q, modulación 64 QAM.....	65
Figura 2.79. Desnormalizador de potencia.....	66
Figura 2.80. Señales de entrada y salida del desnormalizador de potencia de la rama I, modulación 16 QAM	66
Figura 2.81. Señales de entrada y salida del desnormalizador de potencia de la rama Q, modulación 16 QAM	66
Figura 2.82. Señales de entrada y salida del desnormalizador de potencia de la rama I, modulación 64 QAM	67
Figura 2.83. Señales de entrada y salida del desnormalizador de potencia de la rama Q, modulación 64 QAM	67
Figura 2.84. Decisor	68
Figura 2.85. Señales de entrada y salida del decisor de la rama I, modulación 16 QAM	69
Figura 2.86. Señales de entrada y salida del decisor de la rama Q, modulación 16 QAM.....	69
Figura 2.87. Señales de entrada y salida del decisor de la rama I, modulación 64 QAM	70
Figura 2.88. Señales de entrada y salida del decisor de la rama Q, modulación 64 QAM.....	70
Figura 2.89. Conversor de M a 2 niveles.....	71
Figura 2.90. Señales de entrada y salida del <i>Mcode</i> de la rama I, modulación 16 QAM	72
Figura 2.91. Señales de entrada y salida del <i>Mcode</i> de la rama Q, modulación 16 QAM	72
Figura 2.92. Señales de entrada y salida del <i>Mcode</i> de la rama I, modulación 64 QAM	73
Figura 2.93. Señales de entrada y salida del <i>Mcode</i> de la rama Q, modulación 64 QAM	73
Figura 2.94. Señales de entrada y salida del <i>Up Sample</i> de la rama I, modulación 16 QAM.....	74
Figura 2.95. Señales de entrada y salida del <i>Up Sample</i> de la rama Q, modulación 16 QAM.....	74
Figura 2.96. Señales de entrada y salida del <i>Up Sample</i> de la rama I, modulación 64 QAM.....	75
Figura 2.97. Señales de entrada y salida del <i>Up Sample</i> de la rama Q, modulación 64 QAM.....	75
Figura 2.98. Señales de entrada y salida del <i>Parallel to Serial</i> de la rama I, modulación 16 QAM.....	76

Figura 2.99. Señales de entrada y salida del <i>Parallel to Serial</i> de la rama Q, modulación 16 QAM.....	76
Figura 2.100. Señales de entrada y salida del <i>Parallel to Serial</i> de la rama I, modulación 64 QAM.....	77
Figura 2.101. Señales de entrada y salida del <i>Parallel to Serial</i> de la rama Q, modulación 64 QAM.....	77
Figura 2.102. Señales de las ramas I y Q desincronizadas, modulación 16 QAM.....	78
Figura 2.103. Señales de las ramas I y Q desincronizadas, modulación 64 QAM.....	78
Figura 2.104. Multiplexor de datos.....	79
Figura 2.105. Secuencia de bits recuperada, modulación 16 QAM.....	79
Figura 2.106. Secuencia de bits recuperada, modulación 64 QAM.....	80
Figura 2.107. Secuencia de bits predefinida y secuencia de bits recuperada, modulación 16 QAM.....	80
Figura 2.108. Secuencia de bits predefinida y secuencia de bits recuperada, modulación 64 QAM.....	81
Figura 2.109. Secuencia de bits predefinida y secuencia de bits recuperada, ruido igual a cero, modulación 16 QAM	81
Figura 2.110. Secuencia de bits predefinida y secuencia de bits recuperada, ruido igual a cero, modulación 64 QAM	82
Figura 2.111. Calculador de BER	82
Figura 2.112. Calculo de la BER.....	83
Figura 2.113. Sistemas básicos de comunicaciones banda base con modulación 16/64 QAM en Simulink®	84
Figura 2.114. Curvas de desempeño a partir de Simulink® y System Generator, modulación 16 QAM	85
Figura 2.115. Curvas de desempeño a partir de Simulink® y System Generator, modulación 64 QAM	85
Figura 2.116. Uso del Token para generación de archivos de programación.....	86
Figura 2.117. Programación del FPGA	88
Figura 2.118. Señales de entrada y salida obtenidos a partir de implementación, sistema básico de comunicaciones banda base con modulación 16/64 QAM.....	88
Figura 3.1. Curvas de desempeño para velocidades de transmisión de datos 500 Kbps y 10 Mbps, simulación en System Generator con modulación 16 QAM.....	90
Figura 3.2. Curvas de desempeño para velocidades de transmisión de datos 500 Kbps y 10 Mbps, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM.....	91
Figura 3.3. Curvas de desempeño mapeos Gray y Binario, simulación en System Generator con modulación 16 QAM.....	92
Figura 3.4. Curvas de desempeño mapeos Gray y Binario, simulación en System Generator con modulación 64 QAM.....	92
Figura 3.5. Curvas de desempeño mapeos Gray y Binario, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM.....	93
Figura 3.6. Curvas de desempeño mapeos Gray y Binario, implementación sobre hardware reconfigurable (FPGA) con modulación 64 QAM.....	93
Figura 3.7. Curvas de desempeño para transmisión de 32768 y 1048576 bits, simulación en System Generator con modulación 16 QAM	94
Figura 3.8. Curvas de desempeño para transmisión de 32768 y 1048576 bits, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM	95

Figura 3.9. Curvas de desempeño para transmisión de 32768 y 1048576 bits, simulación en System Generator con modulación 16 QAM	97
Figura 3.10. Curvas de desempeño para transmisión de 32768 y 1048576 bits, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM	97
Figura 3.11. Comparación curvas de desempeño, simulación en System Generator e implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM y mapeo Gray.....	99
Figura 3.12. Comparación curvas de desempeño, simulación en System Generator e implementación sobre hardware reconfigurable (FPGA) con modulación 64 QAM y mapeo Gray.....	99
Figura 3.13. Comparación curvas de desempeño, implementación sobre hardware reconfigurable (FPGA) y teórica con modulación 16 QAM y mapeo Gray.....	100
Figura 3.14. Comparación curvas de desempeño, implementación sobre hardware reconfigurable (FPGA) y teórica con modulación 64 QAM y mapeo Gray.....	100
Figura 3.15. Comparación curvas de desempeño, implementación sobre hardware reconfigurable (FPGA) con modulación 16/64 QAM y mapeo Gray	101

LISTADO DE ACRÓNIMOS

ASK	: Modulación por Desplazamiento de Amplitud, <i>Amplitude Shift Keying</i>
AWGN	: Ruido Blanco Gaussiano Aditivo, <i>Additive White Gaussian Noise</i>
BER	: Tasa de Error de Bit, <i>Bit Error Rate</i>
CDMA	: Acceso Múltiple Por División De Código, <i>Code Division Multiple Access</i>
CPU	: Unidad Central de Procesamiento, <i>Central Processing Unit</i>
DSP	: Procesamiento Digitales de Señales, <i>Digital Signal Processing</i>
EVM	: Magnitud del Vector de Error, <i>Error Vector Magnitude</i>
FIR	: Respuesta Finita al Impulso, <i>Finite Impulse Response</i>
FPGA	: Arreglos de Compuertas de Campo Programable, <i>Field Programmable Gate Arrays</i>
FSK	: Modulación por Desplazamiento de Frecuencia, <i>Frequency Shift Keying</i>
GUI	: Interfaz Gráfica de Usuario, <i>Graphical User Interface</i>
HDL	: Lenguaje de Descripción de Hardware, <i>Hardware Description Language</i>
IID	: Independiente e Idénticamente Distribuido, <i>Independent and Identically Distributed</i>
ISE	: Ambiente de Software Integrado, <i>Integrated Software Environment</i>
ISI	: Interferencia Intersimbolo, <i>Intersymbol Interference</i>
LFSR	: Registro de Desplazamiento con Realimentación Lineal, <i>Linear Feedback Shift Register</i>
LSB	: Bit Menos Significativo, <i>Least Significant Bit</i>
m-QAM	: Modulación de Amplitud en Cuadratura m-aria, <i>m-ary Quadrature Amplitude Modulation</i> .
MSB	: Bit Más Significativo, <i>Most Significant Bit</i>
OFDM	: Multiplexación por División de Frecuencia Ortogonal, <i>Orthogonal Frequency Division Multiplexing</i>
PDF	: Función de Densidad de Probabilidad, <i>Probability Density Function</i>

PLD	: Dispositivos Lógicos Programables, <i>Programmable Logic Device</i>
PSD	: Densidad Espectral de Potencia, <i>Power Spectral Density</i>
PSK	: Modulación por Desplazamiento de Fase, <i>Phase Shift Keying</i>
QAM	: Modulación de Amplitud en Cuadratura, <i>Quadrature Amplitude Modulation</i>
RAM	: Memoria de Acceso Aleatorio, <i>Random Access Memory</i>
ROM	: Memorias de Solo Lectura, <i>Read Only Memory</i>
SNR	: Relación Señal a Ruido, <i>Signal to Noise Ratio</i>
SoC	: Sistemas en Chip, <i>Systems on Chip</i>
UBE	: Estimación por Fronteras de Unión, <i>Union Bound Estimation</i>
VHDL	: Lenguaje de Descripción Hardware de Circuitos Integrados de Muy Alta Velocidad, <i>Very High Speed Integrated Circuit Hardware Description Language</i>
WGNG	: Generador de Ruido Blanco Gaussiano, <i>White Gaussian Noise Generator</i>
XST	: Tecnología de Síntesis de Xilinx, <i>Xilinx Synthesis Technology</i>

INTRODUCCIÓN

Ante los elevados costos de implementación de un sistema de comunicaciones sobre hardware, por tratarse de dispositivos específicos con limitado número de aplicaciones y servicios, existe la necesidad de contar con un sistema hardware flexible y reconfigurable que permita una disminución significativa de los costos de implementación y que a su vez posibilite la evaluación del desempeño a nivel físico de los diversos esquemas de modulación en un sistema de comunicaciones real.

Aunque se ha desarrollado un sinnúmero de tecnologías de hardware reconfigurable, la tecnología de Arreglos de Compuertas Programables de Campo (FPGAs, *Field Programmable Gate Array*) representa hoy en día la tecnología de mayor acogida, pues aunque es una tecnología ya madura sus avances continúan. Actualmente los fabricantes de FPGAs proporcionan, además del hardware, herramientas software que facilitan el diseño y síntesis¹ de modelos de sistemas de comunicaciones, mejorando los procesos de desarrollo a través de plataformas de simulación. Estas herramientas ofrecen modelos gráficos de programación a partir de bloques predefinidos cuyas combinaciones permiten la implementación de una gran variedad de diseños.

De acuerdo a lo anterior, considerando el gran potencial que representa el uso de FPGAs como hardware reconfigurable para la implementación de sistemas de comunicaciones, y por lo tanto su aplicación en esquemas de modulación, en el presente trabajo de grado se evaluó y analizó el desempeño de un sistema de comunicaciones con Modulación de Amplitud en Cuadratura (QAM, *Quadrature Amplitude Modulation*) implementado sobre hardware reconfigurable (FPGA). Así, este estudio, se considera un avance como referente en materia de implementación, evaluación y análisis del desempeño de esquemas de modulación sobre hardware reconfigurable que permite la exploración de otros campos de aplicación, como alternativa de solución ante los altos costos de implementación hardware.

¹ Síntesis: proceso que inicia desde un nivel de abstracción lógico alto y genera automáticamente un nivel de abstracción lógico bajo.

1. GENERALIDADES

Como una antesala a todas las consideraciones que han de tenerse en cuenta a lo largo del desarrollo del presente trabajo de grado, este capítulo se enfoca en abordar un marco conceptual general que busca abarcar en mayor medida todos aquellos conceptos indispensables que permitan conseguir el cumplimiento de los objetivos trazados. Los capítulos siguientes documentan el modelado, simulación e implementación además de la validación, plan de pruebas, resultados, análisis y conclusiones.

1.1 SISTEMAS DE COMUNICACIONES DIGITALES

Los sistemas de comunicaciones han estado desde siempre inmersos en las relaciones del ser humano, desde la pronunciación de los primeros vocablos hasta los avanzados sistemas de comunicaciones digitales e inalámbricos de hoy, caracterizándose en todo momento por contar con tres componentes fundamentales a saber: Emisor o Transmisor, Medio y Receptor; encargados de generar, transportar y recibir la información respectivamente. La figura 1.1 muestra el diagrama en bloques de un sistema de comunicaciones general, donde se identifican claramente las tres etapas mencionadas.

De este modo, se define un sistema de comunicaciones como el conjunto de elementos que interactúan entre sí manipulando una o varias señales para llevar un mensaje de un lugar (transmisor) a otro (receptor) de manera exitosa, atendiendo criterios de confiabilidad y calidad, entre otros, de la señal transmitida[1].

Figura 1.1. Diagrama en bloques sistema de comunicaciones general



Aunque los inicios de los sistemas digitales se sustentan a partir de los estudios de Claude Shannon en 1948, especialmente con la Teoría de la Información², un primer sistema digital se encuentra en el telégrafo de Samuel Morse en 1837. Las señales discretas transmitidas son de hecho señales digitales por definición y la codificación se asemeja en mecánica a la codificación binaria de los sistemas digitales de la actualidad, evaluando la probabilidad de ocurrencia (la clave morse da menos caracteres a las letras más usadas).

Un sistema de comunicaciones digital se caracteriza por tener una fuente de información discreta donde la señal de información ya no cambia en forma continua sino que presenta variaciones por pasos finitos. La ventaja está en que al tener valores finitos, se reduce el requerimiento de recursos para su procesamiento y almacenamiento, en comparación con los sistemas analógicos; por otro lado, la mayoría de los sistemas de comunicaciones actuales pueden modelarse e implementarse a través de software, hecho que da mucha más versatilidad al diseño ya que obvia el condicionamiento a recursos comercialmente disponibles, dando al diseñador la capacidad de satisfacer los requerimientos de su

² Se ocupa de la medición de la información y de la representación de la misma, así como también de la capacidad de los sistemas de comunicación para transmitir y procesar información.

sistema de forma mucho más completa. En contraparte, todas las señales generadas por el hombre y su entorno son básicamente analógicas, con lo cual, siempre será necesaria una interfaz analógico-digital, digital-analógico que alterará el desempeño del sistema de comunicaciones, ya que este tipo de procesos introducen alteraciones en la calidad.

1.1.1 Componentes básicos de un sistema de comunicaciones digital. A partir del diagrama en bloques planteado en la figura 1.1, los componentes básicos de todo sistema de comunicaciones digital se ilustran en la figura 1.2 [2].

Figura 1.2. Diagrama en bloques sistema de comunicaciones digital



Bloque 1. Fuente. Encargada de generar la información a transmitir, entrega al resto del sistema de comunicaciones la secuencia bits a la cual se le aplica todo tipo de proceso que le permita llegar al destino con las mejores condiciones y así, ser entendida.

Bloque 2. Modulador / Transmisor. Encargado de realizar adecuaciones sobre la secuencia de bits para ser transmitida exitosamente, modifica reversiblemente dicha secuencia en señales continuas en el tiempo (formas de onda).

Bloque 3. Canal de transmisión. Encargado de proveer la conexión entre el transmisor y el receptor, corresponde al medio físico a través del cual se envía la información a lo largo de la comunicación. Este canal físico puede ser un par de cables que transporten señales eléctricas, o una línea de fibra óptica que transporte la información sobre un haz de luz modulada, o espacio libre sobre el cual una portadora modulada por información es radiada por medio de una antena.

Como consecuencia del paso de la información a través de canal, esta se ve afectada por contaminaciones tales como: ruido, interferencias, desvanecimientos, retardos, entre otros.

Bloque 4. Receptor/Demodulador. Responsable tanto de la correcta recuperación de la información proveniente del canal de transmisión como de reproducirla en el destino. Aplica procesos inversos a los realizados en el transmisor además de mecanismos de decisión.

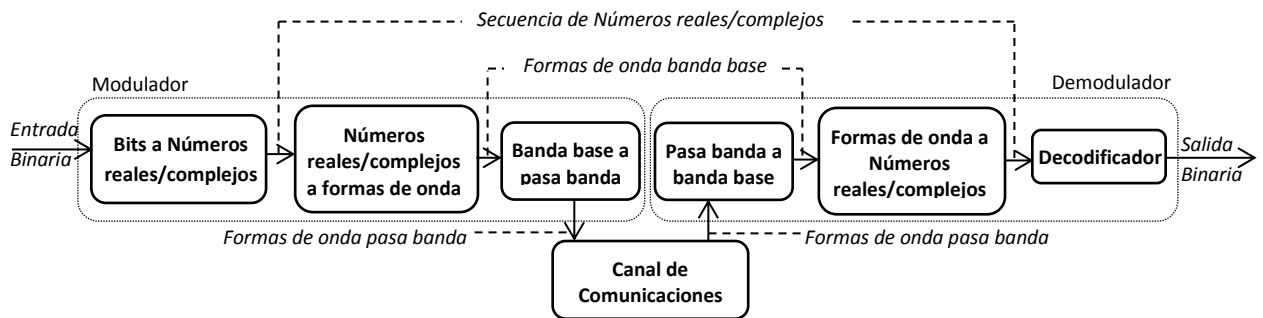
Bloque 5. Destino. Entendido como el último paso en la comunicación, corresponde al componente final con la capacidad de entender la información digital.

En términos generales, los pasos básicos llevados a cabo en el bloque modulador/transmisor resultan ser: un procedimiento para representar la secuencia de bits en una secuencia de número reales o complejos dependiendo del esquema de modulación, seguida de la asignación de formas de onda banda base. De este modo el demodulador, aplicando operaciones inversas, representa las formas de onda recibidas como una secuencia de números reales o complejos, los cuales a su vez son representadas como secuencias de bits, probablemente igual a las secuencias de entrada.

Ahora bien, los moduladores usualmente incluyen un paso adicional, consistente en trasladar las formas de onda banda base a una frecuencia portadora f_c , obteniendo así las correspondientes formas de onda pasa banda. En el demodulador las formas de onda pasa banda son representadas nuevamente como formas de onda banda base y posteriormente se recupera la secuencia de bits transmitida.

A la operación de conversión de frecuencia (traslación hacia y desde f_c) en transmisor y receptor, respectivamente, se le suele denominar modulación y demodulación, pero hoy en día es mucho más común referirse como modulación al proceso completo de asignar secuencias de números reales o complejos y la posterior asignación de formas de onda banda base y pasa banda, como se muestra en la figura 1.3 [3].

Figura 1.3. Etapas de modulación y demodulación



Con lo anterior, los sistemas de comunicaciones pueden ser clasificados en dos grupos dependiendo del rango de frecuencias utilizadas para la transmisión de información: banda base y pasa banda. La transmisión banda base envía la secuencia de bits en su forma original en términos de frecuencia (sin traslación en el dominio de la frecuencia), mientras que pasa banda desplaza la secuencia de bits a una frecuencia mucho más alta (frecuencia portadora que resulta ser mucho más grande que la máxima componente de la señal banda base), para ser transmitida de modo tal que del lado del receptor pueda ser desplazada nuevamente a su frecuencia original.

Para transmitir una señal banda base a través de un sistema de comunicaciones digital, la secuencia de bits que representa la información se debe representar por un conjunto finito y discreto de números reales o complejos, a cada uno de los cuales se les asigna formas de onda que repercutirán en potencia de transmisión, ancho de banda, y detección-corrección de errores, entre otros. En el receptor, dadas las contaminaciones que se adicionan en el canal de transmisión principalmente, es necesario tomar una decisión acerca de la forma de onda que se transmitió, hecho en el cual la energía de cada forma de onda (área bajo la curva) desempeña un papel de alta importancia afectando la probabilidad de detección correcta. Adicionalmente, ya que la información no es trasladada a una frecuencia superior, no se hacen necesarios circuitos de detección y enganche de portadora.

Para transmitir una señal pasa banda se recurre a la traslación de la señal banda base hacia una frecuencia central diferente de cero (por lo general del orden de los megahertz e incluso gigahertz) con el fin de brindarle mejores condiciones al momento de propagarse por canal de comunicaciones. De este modo, la secuencia banda base es multiplicada por

una senoide con frecuencia f_c dando como resultado el corrimiento espectral de la figura 1.4, con formas de onda dependientes de la secuencia de números reales o complejos; números que, vistos en su forma polar, dadas sus características de fase y amplitud, definen cómo cambia la fase y amplitud de la senoide.

Como resultado de la traslación espectral se tiene que el ancho de banda de la señal de información se duplica y se hace necesario recurrir a un circuito tanto de recuperación como de enganche de portadora.

Figura 1.4. Traslación lineal espectral



1.1.2 Modulación Digital. La modulación digital corresponde al proceso mediante el cual una secuencia de bits, que representa una señal de información, es convertida en una forma de onda apropiada para ser transmitida sobre un canal de comunicaciones [3]. Este proceso consiste en la variación sistemática de uno o más parámetros (amplitud, frecuencia o fase) de una señal portadora analógica sinusoidal de acuerdo a una señal de información.

Teniendo en cuenta el parámetro de variación de la señal portadora, se tienen tres tipos de esquemas de modulación primarios: Modulación por Desplazamiento de Amplitud (ASK, *Amplitude Shift Keying*), Modulación por Desplazamiento de Frecuencia (FSK, *Frequency Shift Keying*) y Modulación por Desplazamiento de Fase (PSK, *Phase Shift Keying*); a partir de los cuales se pueden obtener esquemas derivados como es el caso de QAM, donde, además de la variación de la amplitud de la señal portadora, se presenta también variación de la fase.

Las modulaciones digitales se clasifican en modulaciones binarias y modulaciones multinivel de acuerdo al número de posibles estados (símbolos) que pueda tomar la señal portadora para representar una señal de información. En el caso de las modulaciones binarias el número de símbolos es igual a dos, lo que implica el uso de un solo bit por símbolo; en el caso de las modulaciones multinivel el número de símbolos es igual a m , lo que implica el uso de $\log_2(m)$ bits por símbolo, tal como lo presenta la ecuación 1.1. Ya que la cantidad de bits por símbolo transmitido es mayor en la modulación multinivel, esta

modulación supone un incremento en la velocidad de transmisión de datos frente a la modulación binaria

$$M = \log_2(m), \quad (1.1)$$

donde,

M , número de bits por símbolo.

m , número de estados (o símbolos) de la modulación.

1.1.2.1 Modulación QAM. Técnica de modulación digital multinivel cuya información está contenida tanto en la amplitud como en la fase de la señal que se transmite. Esencialmente se transmiten dos señales de información independientes por un único camino. Esto se consigue modulando una misma portadora, desfasada 90° entre una señal de información y la otra [4]. La modulación QAM consiste en dos etapas básicas:

- **Etapla banda base.** En esta etapa una secuencia de bits es convertida en una secuencia de números complejos y la secuencia de números complejos es transformada en formas de onda que atraviesan apropiadamente el canal de comunicaciones permitiendo la correcta recuperación de la información en el demodulador [3]. La salida de la etapa banda base es ilustrada en la ecuación 1.2.

$$u(t) = \sum_{n=-\infty}^{\infty} a[n]p(t - nT), \quad (1.2)$$

donde,

$u(t)$, salida de la etapa banda base.

$a[n]$, secuencia de símbolos complejos.

$p(t - nT)$, pulsos desplazados entre si T segundos, representan las formas de onda.

- **Etapla pasa banda.** En esta etapa, la salida banda base $u(t)$ es trasladada a una frecuencia f_c , para lograrlo, $u(t)$ es multiplicada por $e^{2\pi j f_c t}$ originando una forma de onda $u(t)e^{2\pi j f_c t}$, ya que esta forma de onda es compleja, se adiciona su conjugado complejo para convertirla en una forma de onda real, el resultado corresponde a la salida de la etapa pasa banda, tal como lo muestra la ecuación 1.3 [3].

$$\begin{aligned} x(t) &= u(t)e^{2\pi j f_c t} + u(t)^*e^{-2\pi j f_c t} \\ &= 2\Re\{u(t)\}\cos(2\pi f_c t) - 2\Im\{u(t)\}\sen(2\pi f_c t), \end{aligned} \quad (1.3)$$

donde,

$x(t)$, salida de la etapa pasa banda.

$u(t)$, salida de la etapa banda base.

$u(t)^*$, conjugado complejo de $u(t)$.

$\Re\{ \}$, operador parte real.

$\Im\{ \}$, operador parte imaginaria.

Como lo presenta la ecuación 1.3, las componentes real e imaginaria de la salida banda base son multiplicadas por una misma portadora desfasada 90° entre una componente y la otra, obteniéndose dos señales ortogonales entre sí que pueden ser mezcladas sin temor a perderse la capacidad de distinguir entre una y otra.

1.1.2.2 Sistema básico de comunicaciones banda base con modulación m-QAM. La figura 1.5 ilustra el diagrama en bloques de un sistema básico de comunicaciones banda base con modulación m-QAM y un canal no limitado en banda que genera Ruido Blanco Gaussiano Aditivo (AWGN, *Additive White Gaussian Noise*).

Figura 1.5. Sistema básico de comunicaciones banda base con modulación m-QAM



Bloque 1. Fuente de información. Genera una secuencia aleatoria de bits $b[n]$ con periodo de bit T_b (tiempo necesario para transmitir un bit) y velocidad de transmisión de datos R_b (número de bits que se transmiten por unidad de tiempo). Se asume que las probabilidades de que cada uno de los bits, sea cero o uno, son iguales y no dependen de los bits anteriores ni condicionan los siguientes, por lo tanto, estos bits son considerados equiprobables³ y estadísticamente independientes⁴.

Bloque 2. Modulador QAM banda base. Agrupa los bits de entrada en conjuntos de M bits (ver ecuación 1.1), genera un símbolo complejo por cada conjunto de bits, y representa cada símbolo con una forma de onda rectangular. El resultado, es una secuencia de símbolos complejos $a[n]$ representada por una señal multinivel compleja $u(t)$, cuyos niveles de amplitud corresponden con los valores de los símbolos complejos.

El conjunto de todos los posibles valores $\{a_1, a_2, \dots, a_m\}$ que puede tomar la secuencia de símbolos complejos $a[n]$ es denominado alfabeto y la manera como se asocia a cada grupo de M bits de entrada un símbolo complejo de un alfabeto \mathcal{A} , es denominado mapeo. La señal multinivel compleja $u(t)$ es caracterizada por un periodo de símbolo T_s y una velocidad de transmisión de símbolos R_s , tal como lo presentan las ecuaciones 1.4 y 1.5.

$$T_s = MT_b = T_b \log_2 m, \quad (1.4)$$

donde,

T_s , periodo de símbolo [s].

M , número de bits por símbolo.

T_b , periodo de bit [s].

m , número de estados de la modulación QAM.

³ Los bits de una secuencia se consideran equiprobables cuando la probabilidad de ocurrencia de cada uno ellos es igual.

⁴ Los bits de una secuencia se consideran estadísticamente independientes, cuando un bit emitido en un instante determinado no depende de sus antecesores ni condiciona los siguientes.

$$R_s = MR_b = R_b \log_2 m, \quad (1.5)$$

donde,

R_s , velocidad de transmisión de símbolos [bps].

M , número de bits por símbolo.

R_b , velocidad de transmisión de datos [bps].

m , número de estados de la modulación QAM.

La ecuación 1.6 presenta la señal multinivel compleja $u(t)$ generada por el modulador; esta señal es representada por ondas rectangulares separadas entre sí T_s segundos con amplitudes iguales a los valores de la secuencia de símbolos complejos $a[n]$.

$$u(t) = \sum_{n=-\infty}^{\infty} a[n] \text{rect}(t - nT_s), \quad (1.6)$$

donde,

$u(t)$, señal multinivel compleja (señal de salida del modulador QAM banda base).

$a[n]$, secuencia de símbolos complejos.

T_s , periodo de símbolo [s].

$\text{rect}(t)$, denota una forma de onda rectangular, definida como: $\text{rect}(t) = \begin{cases} 1 & |t| < T_s/2 \\ 0 & |t| > T_s/2 \end{cases}$

Dado que se considera un canal de comunicaciones AWGN no limitado en banda, es posible recuperar las formas de onda rectangulares (en el receptor) con ausencia de Interferencia Intersimbolo (ISI, *Intersymbol Interference*)⁵ sin la necesidad de representar la secuencia de símbolos complejos $a[n]$ como formas de onda con espectro limitado en banda⁶.

- **Tipos de mapeo.** Generalmente el mapeo utilizado en un modulador es el mapeo Gray, sin embargo, resulta interesante compararlo con el mapeo Binario, ya que el uso de un mapeo u otro altera directamente la probabilidad de error (P_e)⁷ de un sistema de comunicaciones digital.

El mapeo Gray garantiza que la diferencia entre símbolos adyacentes⁸ corresponda únicamente a un bit, caso contrario al mapeo Binario, donde la diferencia entre

⁵ Fenómeno indeseable en comunicaciones digitales en el cual los pulsos que representan a dos símbolos distintos se traslapan entre sí, provocando confusiones y errores en la detección. Entre las múltiples causales se encuentran: los efectos dispersivos del canal y una elección inadecuada de la forma de onda que representa los símbolos.

⁶ En el caso de un canal limitado en banda los pulsos que representan los diferentes símbolos deben tener un espectro limitado en banda. Si el espectro de los pulsos no es limitado en banda, como en el caso de las formas de onda rectangulares, los pulsos presentan ISI. Para evitar la ISI en un canal limitado en banda usualmente se utilizan formas de onda como coseno roll off o coseno alzado, estas formas de onda son estrictamente limitadas en banda y garantizan ausencia de ISI.

⁷ La probabilidad de error (P_e) es una expectativa teórica (matemática) de la frecuencia promedio con que se presenta un error en la transmisión, entendiéndose como error al fenómeno de detectar incorrectamente un símbolo o un bit transmitido.

⁸ Tomando un símbolo cualquiera como referencia, se consideran símbolos adyacentes o vecinos, a aquellos que se encuentran en las posiciones arriba, abajo, a la izquierda y a la derecha, obviando ubicaciones diagonales.

símbolos adyacentes puede corresponder a más de un bit. Las figuras 1.6 y 1.7 presentan los diagramas de constelación de las modulaciones 16 QAM y 64 QAM, con mapeos Binario y Gray, donde claramente se puede apreciar lo anteriormente expuesto.

Teniendo en cuenta la diferencia entre los mapeos Binario y Gray y considerando que en caso de producirse un error la situación más habitual sería aquella en que el receptor detecte un símbolo como su símbolo adyacente [5]: en el caso del mapeo Gray un símbolo errado correspondería a un bit errado y en el caso del mapeo Binario un símbolo errado podría corresponder a más de un bit errado. Por lo tanto el mapeo Gray reduce la probabilidad de error del sistema de comunicaciones y permite considerar que la probabilidad de error de un bit es igual a la probabilidad de error de un símbolo.

Figura 1.6. Diagramas de constelación, modulaciones 16 QAM y 64 QAM, mapeo Binario

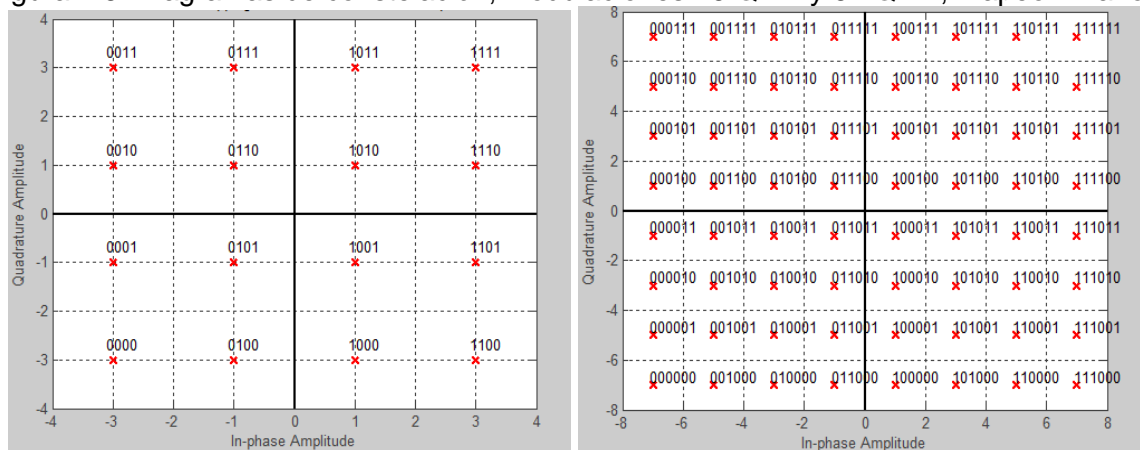
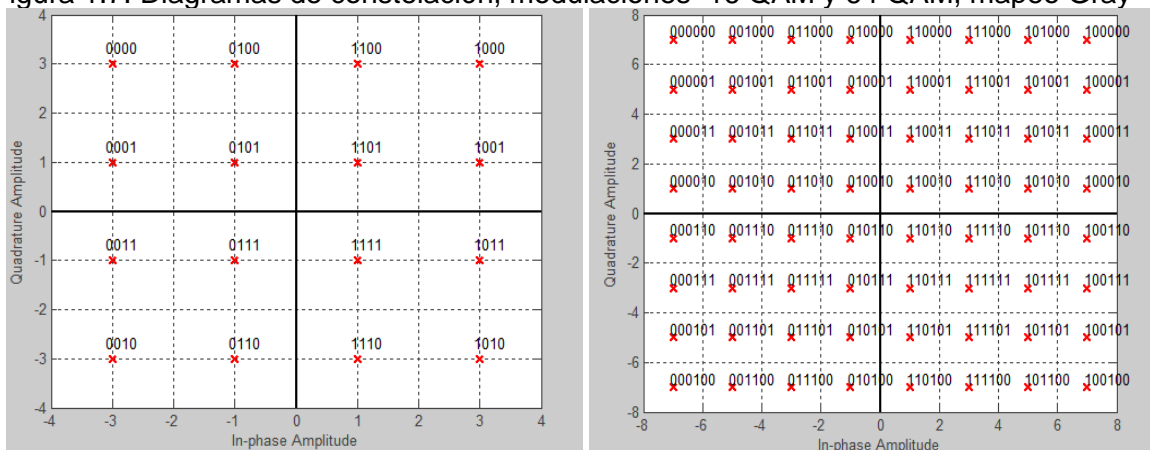


Figura 1.7. Diagramas de constelación, modulaciones 16 QAM y 64 QAM, mapeo Gray



- **Potencia promedio asociada a la constelación de la modulación QAM.** Asumiendo que los bits de entrada al modulador son equiprobables y estadísticamente independientes, cada grupo de M bits, representado por un símbolo, es igualmente probable y asimismo independiente de los otros símbolos. Por lo tanto la secuencia de símbolos complejos $a[n]$ puede ser modelada como un proceso estocástico Independiente e Idénticamente Distribuido (IID, *Independent and Identically Distributed*)⁹ [6].

Bajo estas circunstancias, la potencia promedio de la secuencia de símbolos complejos $a[n]$ es representada como lo muestra la ecuación 1.7.

$$P_A = \sigma_A^2 = \sum_{n=-\infty}^{\infty} |a_n|^2 p_a(a_n), \quad (1.7)$$

donde,

P_A , potencia promedio asociada a la secuencia de símbolos complejos $a[n]$.

σ_A^2 , varianza del proceso estocástico que modela la secuencia de símbolos complejos $a[n]$.

$|a_n|$, magnitud del símbolo n -ésimo de la secuencia de símbolos complejos a_n .

$p_a(a_n)$, probabilidad de ocurrencia del símbolo a_n .

Asimismo, es posible obtener la potencia promedio asociada a la constelación de la modulación m -QAM, calculando el segundo momento de cada uno de los puntos de la constelación, tal como lo muestra la ecuación 1.8.

$$P_a = \sum_{k=1}^n |a_k|^2 p_a(a), \quad (1.8)$$

donde,

P_a , potencia promedio asociada a la constelación m -QAM.

$|a_k|$, magnitud del símbolo k de un alfabeto \mathcal{A} .

p_a , probabilidad de ocurrencia del símbolo a_k .

Debido a la simetría que guardan los cuatro cuadrantes de la constelación m -QAM, es posible calcular la potencia promedio de los puntos de un solo cuadrante de la constelación y multiplicar el resultado por un factor igual cuatro. Asumiendo que los puntos de la constelación se encuentran uniformemente distribuidos, la probabilidad de ocurrencia de un símbolo es igual a $1/m$, siendo m el número de símbolos de la modulación.

De acuerdo a la ecuación 1.8 y teniendo en cuenta las consideraciones antes mencionadas, las potencias promedio asociadas a las constelaciones de las modulaciones 16 QAM y 64 QAM presentadas en las figuras 1.6 y 1.7 son planteadas en las ecuaciones 1.9 y 1.10.

⁹ Un proceso estocástico IID $\{X(t); t \in \mathbb{R}\}$, se caracteriza porque para todo $k > 0$, en todos los instantes t_1, \dots, t_k las variables aleatorias $X(t_1), \dots, X(t_k)$ son IID, toda vez que cuenten con una misma función de probabilidad para todas y además, con una distribución de probabilidad conjunta que sea igual al producto de las funciones marginales.

$$P_{a_{16 QAM}} = 4 \left\{ \frac{1}{16} \left[(\sqrt{2})^2 + 2(\sqrt{10})^2 + (\sqrt{18})^2 \right] \right\} = 10, \quad (1.9)$$

donde,

$P_{a_{16 QAM}}$, potencia promedio asociada a la constelación de la modulación 16 QAM.

$$P_{a_{64 QAM}} = 4 \left\{ \frac{1}{64} \left[(\sqrt{2})^2 + (\sqrt{18})^2 + 2(\sqrt{10})^2 + 3(\sqrt{50})^2 + 2(\sqrt{34})^2 \right. \right. \\ \left. \left. + 2(\sqrt{26})^2 + (\sqrt{98})^2 + 2(\sqrt{74})^2 + 2(\sqrt{58})^2 \right] \right\} = 42, \quad (1.10)$$

donde,

$P_{a_{64 QAM}}$, potencia promedio asociada a la constelación de la modulación 64 QAM.

Generalmente resulta conveniente la normalización en potencia de la señal de salida del modulador QAM banda base con el fin de lograr una potencia promedio igual a uno; para ello, la señal a normalizar es multiplicada por un factor de normalización, este factor corresponde a la raíz cuadrada del valor inverso de la potencia promedio asociada a cada una de las constelaciones de los esquemas de modulación; para los casos de las modulaciones 16 QAM y 64 QAM los factores de normalización corresponden a $1/\sqrt{10}$ y $1/\sqrt{42}$ respectivamente. El proceso de desnormalización en el receptor corresponde a dividir la señal por el factor de normalización aplicado en el transmisor.

Bloque 3. Canal de comunicaciones. Se asume un canal de comunicaciones no limitado en banda que introduce ruido AWGN complejo a la señal de entrada. La salida del canal es modelada como un proceso aleatorio $\{Y(t); t \in \mathbb{R}\}$ representado por la ecuación 1.11.

$$Y(t) = U(t) + W(t), \quad (1.11)$$

donde,

$\{Y(t); t \in \mathbb{R}\}$, proceso aleatorio que modela la salida del canal de comunicaciones.

$\{U(t); t \in \mathbb{R}\}$, proceso aleatorio que modela la entrada al canal de comunicaciones.

$\{W(t); t \in \mathbb{R}\}$, proceso aleatorio de ruido AWGN complejo introducido por el canal de comunicaciones.

La noción de ruido aditivo considera que los procesos $\{U(t); t \in \mathbb{R}\}$ y $\{W(t); t \in \mathbb{R}\}$ son estadísticamente independientes y asume que la señal de entrada no es alterada por ninguna otra perturbación diferente a $\{W(t); t \in \mathbb{R}\}$ [7].

- **Ruido AWGN.** El ruido AWGN es un ruido que siempre aparece, en ausencia o presencia de una señal, es denominado blanco por ser modelado como un proceso aleatorio estacionario¹⁰ con Densidad Espectral de Potencia (PSD, *Power Spectral Density*) constante para toda frecuencia, como lo presenta la ecuación 1.12; su amplitud obedece a la Función de Densidad de Probabilidad (PDF, *Probability Density Function*) gaussiana de media (μ) cero presentada en la ecuación 1.13¹¹.

¹⁰ Los procesos estacionarios se caracterizan por una localización temporal irrelevante ya que se comportan de la misma manera en diferentes instantes, manteniendo las mismas propiedades estadísticas.

¹¹ Cuando el valor de la varianza (σ^2) es igual a uno, se considera que el ruido tiene un nivel de potencia normalizado.

$$W(f) = \frac{N_o}{2} = \sigma^2, \quad (1.12)$$

donde,
 $W(f)$, PSD asociada al proceso aleatorio de ruido AWGN.
 N_o [W/Hz], densidad constante en la notación estándar¹².
 σ^2 , varianza asociada al proceso aleatorio de ruido AWGN.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}, \quad (1.13)$$

donde,
 $p(x)$, PDF gaussiana.
 σ^2 , varianza asociada a la PDF gaussiana.

Dado que la función de autocorrelación del ruido AWGN corresponde a una función delta de Dirac centrada en el origen, como lo presenta la ecuación 1.14, cualquier par de muestras de ruido AWGN tomadas en instantes distintos son no correlacionadas y por lo tanto estadísticamente independientes.

$$R_W(\tau) = \int_{-\infty}^{\infty} \frac{N_o}{2} e^{j2\pi f\tau} df = \frac{N_o}{2} \delta(\tau), \quad (1.14)$$

donde,
 $R_W(\tau)$, función de autocorrelación asociada al proceso aleatorio de ruido AWGN.
 $\frac{N_o}{2}$, PSD asociada al proceso aleatorio de ruido AWGN.
 $\delta(\tau)$, función delta de Dirac.

- **Ruido AWGN complejo.** Este ruido es modelado como un proceso aleatorio complejo (ver ecuación 1.15), cuyas componentes real e imaginaria, corresponden a procesos con PDF gaussiana, de media (μ) cero y varianza (σ^2) $N_o/2$, caracterizadas por ser ortogonales entre sí, lo que es equivalente a tener un par de procesos AWGN independientes, uno que representa la componente real y otro que representa la componente imaginaria del ruido AWGN complejo.

$$W(t) = W_I(t) + W_Q(t), \quad (1.15)$$

donde,
 $W(t)$, proceso de ruido AWGN complejo.
 $W_I(t)$, componente real del proceso de ruido AWGN complejo.
 $W_Q(t)$, componente imaginaria del proceso de ruido AWGN complejo.

¹² N_o : definida por la temperatura de ruido (T_N [K]) y la constante de Boltzmann ($C=1,38 \times 10^{-23}$ [J/K]), de la siguiente manera: $N_o = KT_N$.

Bloque 4. Demodulador QAM banda base. Representa la señal recibida como una secuencia de símbolos complejos $\hat{a}[n]$ y genera los bits correspondientes a cada símbolo complejo de la secuencia. El resultado es una secuencia de bits $\hat{b}[n]$ que representa la información recuperada por el demodulador.

- **Criterio de decisión.** Ya que la señal recibida por el demodulador corresponde a una versión ruidosa de la señal transmitida por el modulador, el demodulador compara los niveles de la señal recibida con un conjunto de valores umbrales preestablecidos y decide con base en un criterio de decisión a qué valores pertenece cada uno de los niveles de dicha señal, originando una nueva señal lo más parecida posible a la señal transmitida por el modulador.

El criterio de decisión permite determinar a qué símbolo de un alfabeto \mathcal{A} corresponde cada uno de los símbolos recibidos por el demodulador. En el caso de una fuente que emite símbolos independientes y equiprobables, el criterio de decisión de mínima probabilidad de error corresponde al criterio de decisión de distancia mínima, el cual consiste en determinar el símbolo de referencia que se encuentra más cercano geoméricamente al símbolo obtenido después del canal, la ecuación 1.16 presenta la condición que garantiza este hecho [2].

$$\min\{|c_k - a_k|\} = d_{\min}, \quad (1.16)$$

donde,

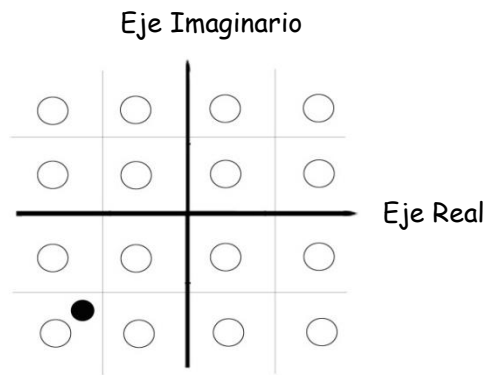
c_k , valor de un símbolo recibido por el demodulador.

a_k , valor del símbolo de referencia más cercano geoméricamente a c_k .

d_{\min} , distancia mínima, garantizada por el símbolo de referencia a_k .

En razón a lo anterior, el plano de una constelación puede dividirse en regiones de decisión centradas alrededor de los símbolos de un alfabeto \mathcal{A} ; por lo tanto, si un símbolo c_k recibido por el demodulador es ubicado en la región de decisión de un símbolo de referencia de valor a_k , el símbolo recibido es representado por el valor de a_k . La figura 1.8 ilustra las regiones de decisión de una constelación, donde el círculo oscuro representa un símbolo recibido en el demodulador y los círculos claros los símbolos de referencia.

Figura 1.8. Regiones de decisión de una constelación 16 QAM



- **Tasa de error de bit.** Ya que el demodulador QAM proporciona la señal de información recuperada, es posible evaluar la calidad de transmisión del sistema de comunicaciones digital a través de la Tasa de Error de Bit (BER, *Bit Error Rate*); esta tasa determina la relación entre el número de bits errados recibidos y el número total de bits transmitidos, tal como lo presenta la ecuación 1.17.

$$BER = \frac{\text{Numero de bits errados}}{\text{Numero de bits enviados}} \quad (1.17)$$

- **Probabilidad de error.** La probabilidad de error (P_e) es una expectativa teórica (matemática) de la frecuencia promedio con que se presenta un error en la transmisión, la ecuación 1.18 permite observar la probabilidad de error obtenida a partir de la Estimación por Frontera de Unión (UBE, *Union Bound Estimate*)[8].

$$P_e\{e_s\} = KQ\left(\frac{d}{2\sigma}\right), \quad (1.18)$$

donde,

e_s , evento error en un símbolo.

K, número promedio de vecinos de cada símbolo.

Q(), función de error complementaria.

d, distancia mínima entre símbolos vecinos de la constelación.

σ , desviación estándar.

Con el fin de obtener la probabilidad de error asociada a las modulaciones 16 QAM y 64 QAM, se halla el valor de K y la correspondiente expresión del argumento de la función de error complementaria en términos de la relación energía de bit a densidad espectral de potencia de ruido (E_b/N_o), para cada una de las modulaciones.

Partiendo de la ecuación 1.8 y asumiendo que las constelaciones de las modulaciones 16 QAM y 64 QAM se encuentran uniformemente distribuidas con una distancia mínima entre símbolos vecinos igual a 2α , se obtiene la energía de símbolo correspondiente a cada una de las modulaciones, tal como lo presentan las ecuaciones 1.19 y 1.20.

$$E_{s16QAM} = 4 \left\{ \frac{1}{16} \left[(\sqrt{2}\alpha)^2 + 2(\sqrt{10}\alpha)^2 + (\sqrt{18}\alpha)^2 \right] \right\} = 10\alpha^2, \quad (1.19)$$

donde,

E_{s16QAM} , energía de símbolo asociada a la modulación 16QAM.

$$E_{s64QAM} = 4 \left\{ \frac{1}{64} \left[(\sqrt{2}\alpha)^2 + (\sqrt{18}\alpha)^2 + 2(\sqrt{10}\alpha)^2 + 3(\sqrt{50}\alpha)^2 + 2(\sqrt{34}\alpha)^2 \right. \right. \\ \left. \left. + 2(\sqrt{26}\alpha)^2 + (\sqrt{98}\alpha)^2 + 2(\sqrt{74}\alpha)^2 + 2(\sqrt{58}\alpha)^2 \right] \right\} = 42\alpha^2, \quad (1.20)$$

donde,

E_{s64QAM} , energía de símbolo asociada a la modulación 64QAM.

Para el caso específico del mapeo Gray se tiene que la relación entre la energía de símbolo y la energía de bit está dada por la ecuación 1.21

$$E_b = \frac{E_s}{\log_2 M'} \quad (1.21)$$

donde,

E_b , energía de bit

E_s , energía de símbolo

M, número de estados de la modulación

De este modo, considerando la relación presentada en la ecuación 1.21 y la energía de símbolo obtenida en las ecuaciones 1.19 y 1.20, se halla el valor de la energía de bit a partir de la energía de símbolo, para cada una de las modulaciones, tal como lo presentan las ecuaciones 1.22 y 1.23.

$$E_{b16QAM} = \frac{10\alpha^2}{4} = \frac{5\alpha^2}{2}, \quad (1.22)$$

donde,

E_{b16QAM} , energía de bit asociada a la modulación 16QAM.

$$E_{b64QAM} = \frac{42\alpha^2}{6} = 7\alpha^2, \quad (1.23)$$

donde,

E_{b64QAM} , energía de bit asociada a la modulación 64QAM.

Despejando α de las ecuaciones 1.22 y 1.23 se obtienen las ecuaciones 1.24 y 1.25.

$$\alpha^2 = \frac{2E_{b16QAM}}{5} \quad (1.24)$$

$$\alpha^2 = \frac{E_{b64QAM}}{7} \quad (1.25)$$

Así, considerando las ecuaciones 1.24 y 1.25, además de la varianza de ruido definida en la ecuación 1.12, se obtiene el argumento de la función de error complementaria planteada en la ecuación 1.18, tal como lo presentan las ecuaciones 1.26 y 1.27 para las modulaciones 16 QAM y 64 QAM respectivamente.

$$\frac{d}{2\sigma} = \sqrt{\frac{d^2}{4\sigma^2}} = \sqrt{\frac{4\alpha^2}{4\frac{N_o}{2}}} = \sqrt{\frac{2\alpha^2}{N_o}} = \sqrt{\frac{2}{N_o} \left(\frac{2E_{b16QAM}}{5} \right)} = \sqrt{\frac{4 E_{b16QAM}}{5 N_o}} \quad (1.26)$$

$$\frac{d}{2\sigma} = \sqrt{\frac{2\alpha^2}{N_o}} = \sqrt{\frac{2}{N_o} \left(\frac{E_{b64QAM}}{7} \right)} = \sqrt{\frac{2 E_{b64QAM}}{7 N_o}} \quad (1.27)$$

Finalmente, el valor del factor K de la ecuación 1.18, se obtiene calculando el número promedio de vecinos de cada símbolo mediante la ecuación 1.28, para cada una de las constelaciones de las modulaciones 16 QAM y 64 QAM.

$$K = \frac{1}{M} \sum_{j=2}^4 j * s, \quad (1.28)$$

donde,

K, número promedio de vecinos de cada símbolo.

M, número de estados de la modulación.

j, número de vecinos por símbolo (para el caso de constelaciones cuadradas un símbolo puede tener mínimo dos y máximo cuatro símbolos vecinos).

s, número de símbolos que cuentan con j símbolos vecinos.

Así, a partir de la ecuación 1.28, el factor K correspondiente a cada una de las modulaciones, está dado por las ecuaciones 1.29 y 1.30, respectivamente.

$$K_{16QAM} = \frac{1}{16} [(2x4) + (3x8) + (4x4)] = \frac{48}{16} = 3, \quad (1.29)$$

donde,

K_{16QAM} , número promedio de vecinos de cada símbolo asociado a la modulación 16QAM.

$$K_{64QAM} = \frac{1}{64} [(2x4) + (3x24) + (4x36)] = \frac{224}{64} = 3.5, \quad (1.30)$$

donde,

K_{64QAM} , número promedio de vecinos de cada símbolo asociado a la modulación 64QAM.

Con lo anterior, reemplazando la ecuación 1.26 y 1.29 en la ecuación 1.18, para el caso de la modulación 16 QAM y reemplazando la ecuación 1.27 y 1.30 en la ecuación 1.18, para el caso de la modulación 64 QAM, se obtiene la probabilidad de error para cada una de las modulaciones, tal como se presenta en las ecuaciones 1.31 y 1.32.

$$P_{e16QAM}\{e_s\} \cong 3Q\left(\sqrt{\frac{4 E_{b16QAM}}{5 N_o}}\right), \quad (1.31)$$

donde,

P_{e16QAM} , probabilidad de error asociada a la modulación 16QAM

$$P_{e64QAM}\{e_s\} \cong 3,5Q\left(\sqrt{\frac{2 E_{b64QAM}}{7 N_o}}\right), \quad (1.32)$$

donde,

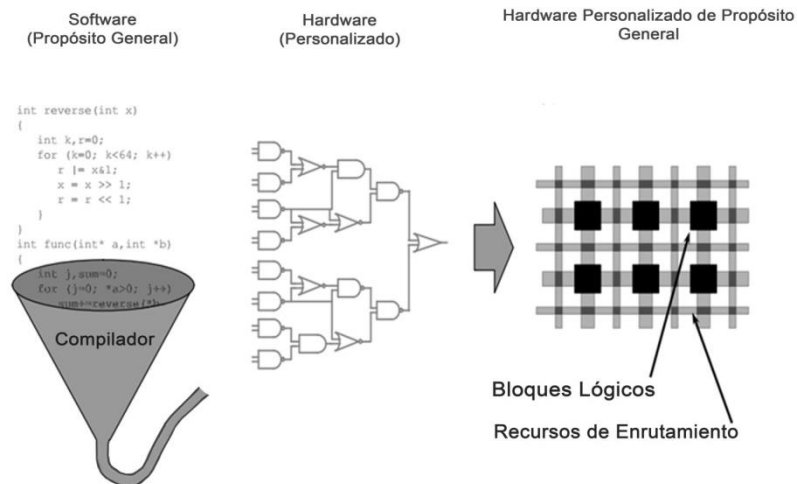
P_{e64QAM} , probabilidad de error asociada a la modulación 64QAM

1.2 HERRAMIENTAS HARDWARE Y SOFTWARE

Como una respuesta y aporte a la versatilidad de los sistemas de comunicaciones digitales de hoy en día, herramientas hardware y software se han venido desarrollando de manera conjunta con el fin de brindar suficientes y mejores mecanismos tanto para el desarrollo como para la implementación a nivel físico de dichos sistemas, con la principal característica de permitir reconfiguración y por ende, reutilización de recursos hardware y software.

1.2.1 Computación reconfigurable. Como se menciona en [9], una de las tecnologías emergentes desde hace ya varios años es la computación reconfigurable, la cual presenta una creciente incursión en las soluciones digitales dadas sus características principales de desempeño y flexibilidad; mismas sustentadas por el avance en los Dispositivos Lógicos Programables (PLD, *Programmable Logic Device*), que permiten un mayor nivel de integración a nivel hardware/software. De este modo, la computación reconfigurable se define como la combinación hardware/software que proporciona un método de implementación de circuitos sobre hardware reconfigurable (ver figura 1.9), en el cual la funcionalidad de los componentes lógicos es adaptable en tiempo de funcionamiento y las conexiones entre dichos componentes es también reconfigurable; consiguiendo así un mejor desempeño del software, como resultado de pasar del ciclo de Buscar-Decodificar-Ejecutar de los microprocesadores tradicionales, a una ejecución en paralelo de múltiples operaciones [10]. Hoy en día se destaca su reconocimiento como solución rentable en la creación rápida de prototipos y algunas aplicaciones de bajo volumen [11].

Figura 1.9. Conformación del Hardware Reconfigurable



Fuente. S. Goldstein, Developers Forum

1.2.1.1 Tecnologías de hardware reconfigurable. Aunque se han desarrollado un sinnúmero de tecnologías de hardware reconfigurable, con su respectivo software, a nivel académico y comercial, de manera general se pueden clasificar en dos categorías:

FPGAs y dispositivos de circuitos integrados con recursos reconfigurables embebidos¹³ [12], [13], [14].

- **Tecnología FPGA.** El FPGA representa hoy en día la tecnología hardware reconfigurable de mayor acogida, pues aunque es una tecnología ya madura, sus avances continúan aumentando la cantidad de compuertas lógicas y memoria disponible, hecho que brinda una mayor posibilidad de integración de funciones que van desde simples multiplicadores cableados, pasando por transmisores-receptores de alta velocidad hasta llegar a núcleos de Unidades Centrales de Procesamiento (CPU, *Central Processing Unit*) con periféricos asociados. Así, la implementación de sistemas de comunicaciones completos sobre un único FPGA se hace posible y además de todo, sencillo con las mejores condiciones en consumo de potencia, desempeño, fiabilidad y costo, con un muy corto tiempo de configuración (del orden de los segundos). Por otro lado, la reconfiguración se puede llevar a cabo en cualquier momento. Las características como posibilidad de correcciones y mejoras en corto tiempo al prototipo final, responder ante cambios de diseño de último momento, soportar múltiples implementaciones (por separado) y disminuir los tiempos de entrega le dan su bien conocida característica de flexibilidad.

La implementación sobre FPGAs se realiza a través de la programación basada en el Lenguaje de Descripción Hardware de Circuitos Integrados de Muy Alta Velocidad (VHDL, *Very High Speed Integrated Circuit Hardware Description Language*) y lenguajes Verilog para la descripción de líneas de entrada; ello sin hacer distinción del fabricante, que entre otros se encuentran: Quicklogic, Atmel, Lattice Semiconductor, Actel, Altera y Xilinx. Altera y Xilinx son los líderes del mercado, por su especial enfoque hacia estudios académicos universitarios.

Dentro de los FPGAs de Altera más conocidos están la familia Stratix cuya arquitectura ha sido diseñada primordialmente para optimizar el desempeño, la densidad de componentes lógicos en un área dada y la implementación de procesos de funciones intensivas tales como filtrado, transformadas y modulación, con una frecuencia de funcionamiento alrededor de 370 MHz. La familia Cyclone ofrece una arquitectura diseñada desde el principio para tener el más bajo costo además de ofrecer un conjunto de características definidas por el usuario para un alto desempeño y bajo consumo de energía a partir de lo cual es posible implementar sistemas digitales complejos sobre un único FPGA con una frecuencia de funcionamiento de alrededor de 250 MHz.

Dentro de los FPGAs de Xilinx, la familia de alto nivel más asequible es Virtex cuya arquitectura diseñada para trabajar con una frecuencia de alrededor de los 500 MHz; resulta ser un dispositivo altamente recomendado para soluciones de alto desempeño, con todas las funciones para aplicaciones de plataformas embebidas; además, es una buena solución para obtener un alto desempeño en aplicaciones de Procesamiento Digitales de Señales (DSP, *Digital Signal Processing*). La familia Spartan, aunque es una versión menos potente que la Virtex, dado su enfoque de satisfacer las

¹³ Como una tercer categoría podrían considerarse los núcleos reconfigurables integrados para Sistemas en Chip (SoC, *Systems on Chip*) pero de manera general, confluyen con la segunda categoría.

necesidades de alto volumen para aplicaciones electrónicas de consumo sensibles al costo, es ideal para un amplio rango de aplicaciones incluyendo acceso de banda ancha, redes telemáticas y televisión digital.

- **Tecnología de dispositivos de circuitos integrados con recursos reconfigurables embebidos.** Estas arquitecturas están basadas principalmente en la combinación de una CPU programable y un arreglo reconfigurable de rutas de datos, dando un enfoque específico hacia aplicaciones de DSP que le permite convertirse tanto en una alternativa frente a FPGAs en este tipo de tareas, pese a su poca madurez, como en un competidor directo de los procesadores de instrucciones DSP convencionales. Como principales ventajas frente a los FPGA se observa el ahorro de energía y la reducción del tamaño, aunque presenta también desventaja en materia de reconfiguración.

Dentro de los dispositivos más reconocidos se pueden citar: circuitos integrados a nivel de sistemas de campo programable y sistemas de máquina de computación.

Hoy en día, un nuevo hardware reconfigurable ha empezado a surgir, entre los cuales sobresalen Arduino y Raspberry Pi (ver figura 1.10), dos herramientas muy similares a los FPGA, pero con microprocesadores de diferentes fabricantes y una tecnología aún no muy madura, en el caso de Arduino; en el caso de Raspberry Pi, el 15 de Septiembre de 2013 lanzó su primer modelo compatible directamente con Simulink® valiéndose del paquete de soporte de Simulink® para hardware Raspberry Pi.

Figura 1.10. Primer modelo de Raspberry para Simulink®



Fuente. Mathworks, MAKERZONE

1.2.1.2 Herramientas software. Aunque pueden encontrarse un gran número de herramientas software, tantas como tecnologías de herramientas hardware, la atención del presente trabajo de grado se centra en dos herramientas específicas dado su enfoque hacia estudios universitarios y en especial, por su articulación con otra herramienta de alto nivel para el diseño de sistemas de comunicaciones (Simulink® de Matlab®), con lo cual el diseñador más que preocuparse por el código de Lenguaje de Descripción de Hardware (HDL, *Hardware Description Language*), tiene que enfocarse únicamente en el modelado de su sistema, ya que el código de programación del FPGA es generado automáticamente por dicha herramienta software a partir de un modelo en Simulink®. Estas herramientas son DSP Builder, para el caso de Altera y System Generator para Xilinx.

- **System Generator.** 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 realistas 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 extremadamente 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®.

Todos los pasos del flujo de descarga para la implementación sobre FPGAs, incluyendo síntesis y ubicación/enrutamiento se realizan automáticamente para generar el archivo de programación.

- **DSP Builder.** DSP Builder de Altera integra dos herramientas: la primera, de alto nivel para desarrollo de algoritmos y la segunda, de HDL a través de la combinación del desarrollo de algoritmos, simulación y verificación de las capacidades de las herramientas de diseño de Mathworks.

DSP Builder consta de un conjunto de bloques de construcción que acorta 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 y verificar las implementaciones de 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.

Finalmente, considerando las herramientas hardware y software anteriormente descritas, en el marco de la computación reconfigurable, para el desarrollo del presente trabajo de grado se seleccionó la tecnología FPGA como hardware reconfigurable y System Generator como herramienta de diseño y síntesis de alto nivel de sistemas de comunicaciones sobre FPGAs, dadas las características y ventajas mencionadas. Su escogencia se detalla en mayor medida en la sección 2.1.3.1.

2. MODELADO, SIMULACIÓN E IMPLEMENTACIÓN

Con el propósito de presentar el sistema básico de comunicaciones banda base con modulación 16/64 QAM, implementado sobre hardware reconfigurable (FPGA), este capítulo expone la metodología de trabajo, la cual se compone de actividades de modelado, simulación e implementación sobre hardware reconfigurable (FPGA) del sistema de comunicaciones, brindando las herramientas necesarias para la evaluación y análisis del desempeño de este sistema en el capítulo 3.

2.1 METODOLOGÍA DE TRABAJO

En razón al cumplimiento de los objetivos del presente trabajo de grado, para el desarrollo del modelado, la simulación y la implementación sobre hardware reconfigurable (FPGA) se empleó el Modelo Lineal Secuencial [15]. La figura 2.1 permite observar la estructura del modelo, en el cual se trabajan las etapas relacionadas a continuación:

Etapa 1. Análisis de requerimientos. Plantea los requisitos del sistema de comunicaciones a modelar de acuerdo a los objetivos del presente trabajo de grado.

Etapa 2. Modelado. Expone el modelo conceptual del sistema de comunicaciones con base en el análisis realizado en la etapa 1.

Etapa 3. Simulación. Considera la selección de herramientas y a partir de ellas presenta el diseño y la simulación del sistema de comunicaciones modelado en la etapa 2. Finalmente realiza la validación del diseño propuesto.

Etapa 4. Implementación. Presenta la creación y descarga del archivo de programación sobre hardware reconfigurable (FPGA) del sistema de comunicaciones de la etapa 3.

Figura 2.1. Modelo Lineal Secuencial



2.1.1 Análisis de requerimientos. Los requisitos que enmarcan el presente trabajo se plantean a continuación:

- Modelar el sistema básico de comunicaciones banda base con modulación 16/64 QAM y un canal AWGN.

- Implementar sobre hardware reconfigurable (FPGA) el sistema de comunicaciones modelado.
- Evaluar el desempeño del sistema de comunicaciones implementado, a partir de la BER y del comportamiento del hardware reconfigurable (FPGA), ante variaciones en las condiciones del canal.

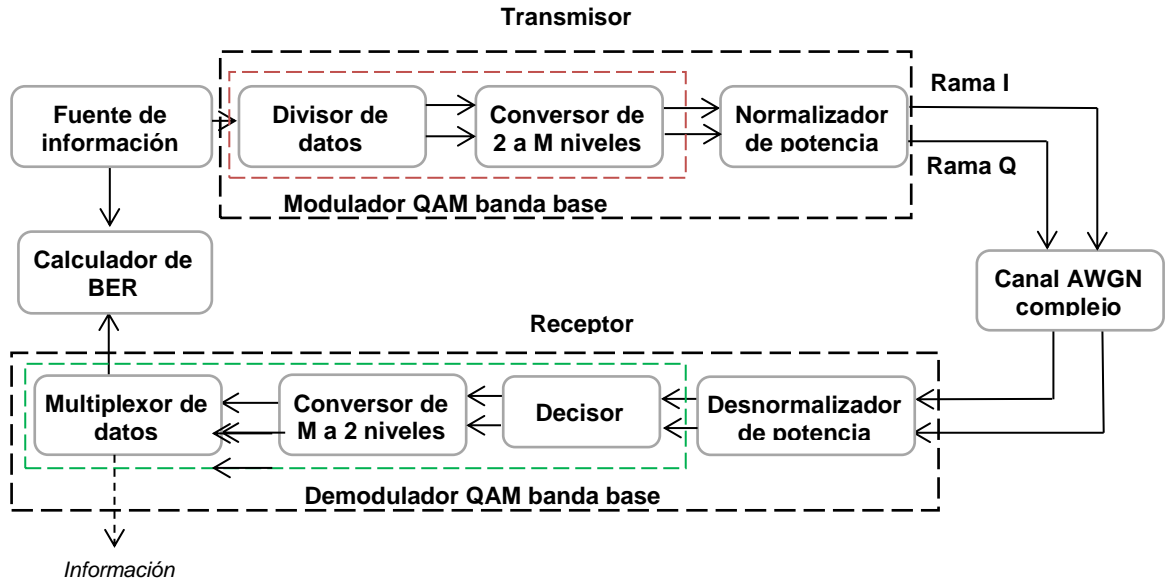
2.1.2 Modelado. Con el objeto de establecer las bases para el desarrollo apropiado del presente trabajo de grado, de acuerdo a los requerimientos planteados en el sección 2.1.1, esta sección explora conceptualmente la obtención de un modelo para el sistema básico de comunicaciones banda base con modulación 16/64 QAM y un canal AWGN, analizando paso a paso el procesamiento que debe de aplicarse sobre la señal proveniente de la fuente de información.

2.1.2.1 Modelo de referencia. Tomando el sistema de comunicaciones definido y descrito en la figura 1.5, como el punto de partida para el modelado del sistema básico de comunicaciones banda base con modulación 16/64 QAM, se inició el estudio pertinente con el fin de determinar los procesos internos de cada bloque.

El sistema de comunicaciones de la figura 1.5 cuenta con cuatro componentes básicos: una fuente de información que genera la señal de información a transmitir; un modulador QAM banda base que convierte la señal de entrada en una señal multinivel compleja adecuada para ser transmitida sobre el canal de comunicaciones; un canal de comunicaciones que adiciona ruido AWGN complejo a la señal multinivel compleja; y un demodulador QAM banda base que convierte la señal afectada por ruido AWGN complejo en una réplica lo más parecida posible a la señal emitida por la fuente de información.

Teniendo en cuenta las funciones de los bloques anteriormente descritos y asumiendo que el nivel de potencia de la señal de salida del modulador QAM banda base es normalizado (con el fin de facilitar la comparación y el análisis del desempeño del sistema básico de comunicaciones), la figura 2.2 presenta el modelo de referencia, para el sistema básico de comunicaciones banda base con modulación 16/64 QAM.

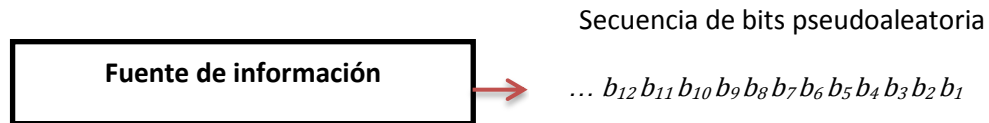
Figura 2.2. Modelo de referencia



Para conseguir las características específicas del sistema básico de comunicaciones a modelar, se determinaron y asumieron aspectos importantes relacionados con el funcionamiento de cada uno de los bloques del modelo de referencia; estos aspectos se detallan a continuación.

Bloque 1. Fuente de información. Encargada de generar una secuencia de bits pseudoaleatoria (ver figura 2.3) con una velocidad de transmisión de datos determinada (R_b). Se asume que los bits generados son equiprobables y estadísticamente independientes entre sí.

Figura 2.3. Fuente de información



Bloque 2. Divisor de datos. Toma los bits generados por la fuente de información y los distribuye por las ramas I y Q.

Teniendo en cuenta la ecuación 1.1, donde se relaciona el número de estados de la modulación (m) con el número de bits por símbolo (M); para garantizar el número de estados de la modulación, el divisor de datos agrupa los bits en conjuntos de M bits y los distribuye por las ramas I y Q, donde los primeros $M/2$ bits de cada conjunto son enviados por la rama I y los siguientes $M/2$ bits por la rama Q.

A efectos de ilustrar la forma como el divisor de datos agrupa y distribuye los bits, estos se denotan con las letras I y Q de acuerdo al esquema de modulación (ver tabla 2.1). Las figuras 2.4 y 2.5 permiten apreciar la forma como el divisor de datos procesa los bits.

Tabla 2.1. Representación de los bits de entrada al divisor de datos

Secuencia de bits	...	b ₁₂	b ₁₁	b ₁₀	b ₉	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
Representación de bits Modulación 16 QAM	...	Q ₆	Q ₅	I ₆	I ₅	Q ₄	Q ₃	I ₄	I ₃	Q ₂	Q ₁	I ₂	I ₁
Representación de bits Modulación 64 QAM	...	Q ₆	Q ₅	Q ₄	I ₆	I ₅	I ₄	Q ₃	Q ₂	Q ₁	I ₃	I ₂	I ₁

Figura 2.4. Divisor de datos, modulación 16 QAM

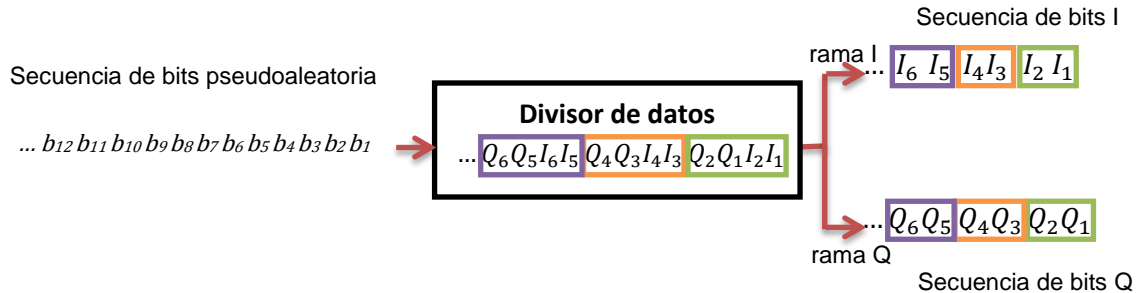
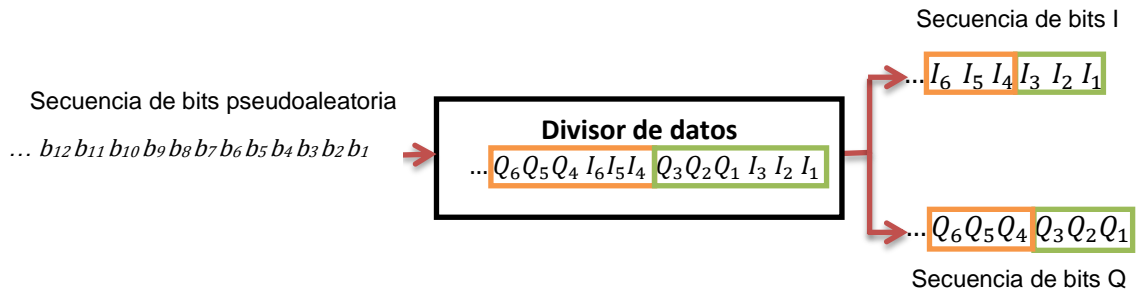


Figura 2.5. Divisor de datos, modulación 64 QAM



Bloque 3. Conversor de 2 a M niveles. Agrupa los bits de cada rama (I y Q) en conjuntos de $M/2$ bits, asocia a cada par de conjuntos de las ramas I y Q el valor de un símbolo complejo de acuerdo al mapeo establecido, y genera una forma de onda rectangular, tanto en la rama I como en la rama Q, por cada símbolo complejo; generando así las señales bipolares multinivel I y Q que componen la señal multinivel compleja resultado de la modulación. Los niveles de amplitud de la señal multinivel compleja corresponden con los valores de cada símbolo complejo (ver figuras 2.6¹⁴ y 2.7).

¹⁴ Un símbolo complejo consta de una parte real y una parte imaginaria; en la figura 2.6, los valores $A_1, A_2, A_3, A...$ denotan la parte real de los símbolos complejos y los valores $B_1, B_2, B_3, B...$ denotan la parte imaginaria de los símbolos complejos, siendo así: $A_1 + jB_1, A_2 + jB_2, A_3 + jB_3...$ los símbolos complejos asociados a cada par de conjuntos de $M/2$ bits de las ramas I y Q. En la figura 2.7 el proceso es análogo al planteado para la figura 2.6.

Figura 2.6. Conversor de 2 a M niveles, modulación 16 QAM

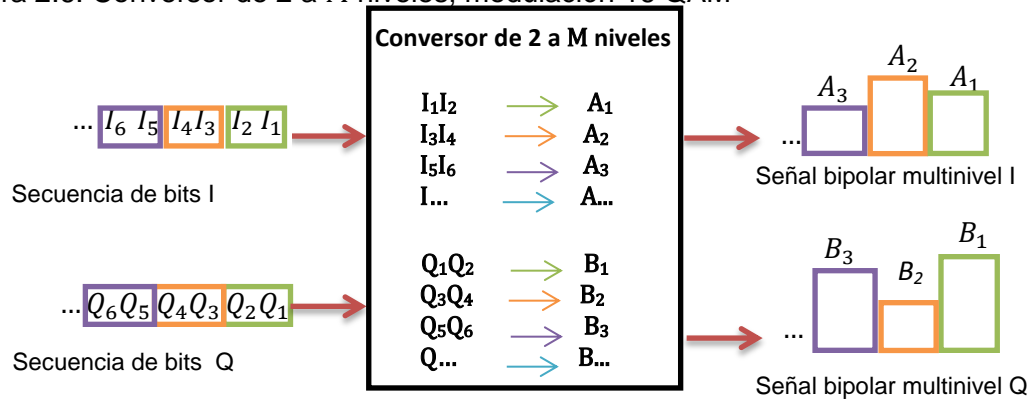
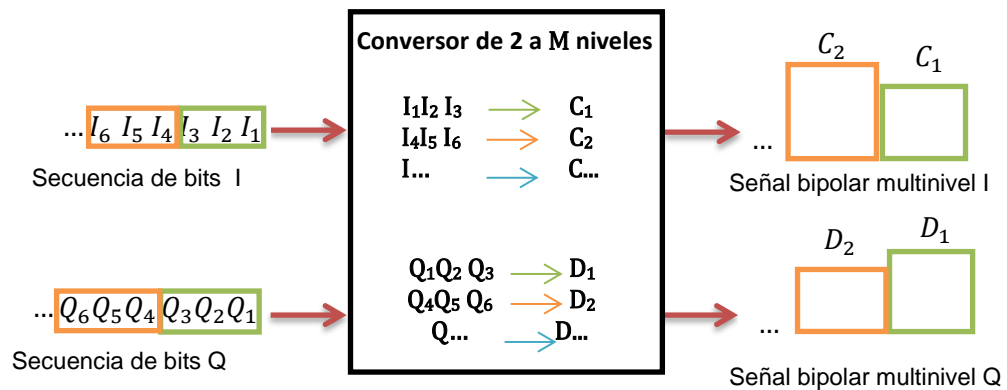


Figura 2.7. Conversor de 2 a M niveles, modulación 64 QAM



Bloque 4. Normalizador de potencia. Multiplica el valor de cada nivel de las señales bipolares multinivel I y Q recibidas, por el correspondiente factor de normalización¹⁵ con el fin de obtener una potencia promedio igual a uno. En cuanto se obtienen las señales de salida del normalizador de potencia, la etapa del transmisor finaliza (ver figuras 2.8 y 2.9).

¹⁵ Como se planteó en la sección 1.1.2.2, para el caso de la modulación 16 QAM el factor de normalización corresponde a $1/\sqrt{10}$ y para el caso de la modulación 64 QAM el factor de normalización corresponde a $1/\sqrt{42}$.

Figura 2.8. Normalizador de potencia, modulación 16 QAM

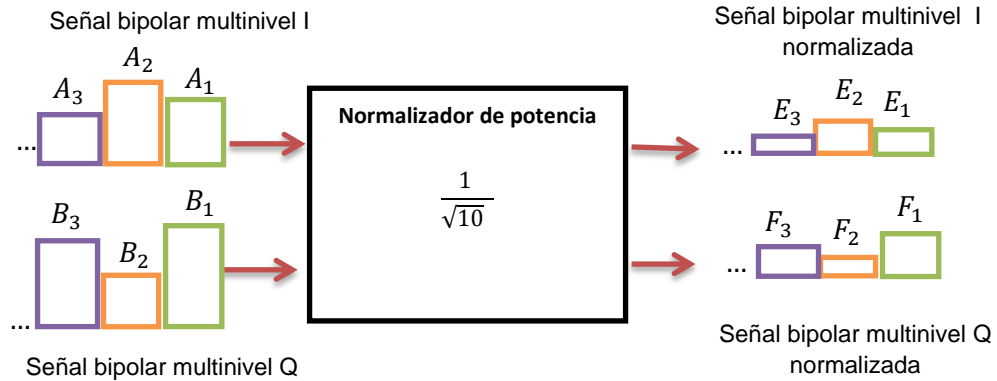
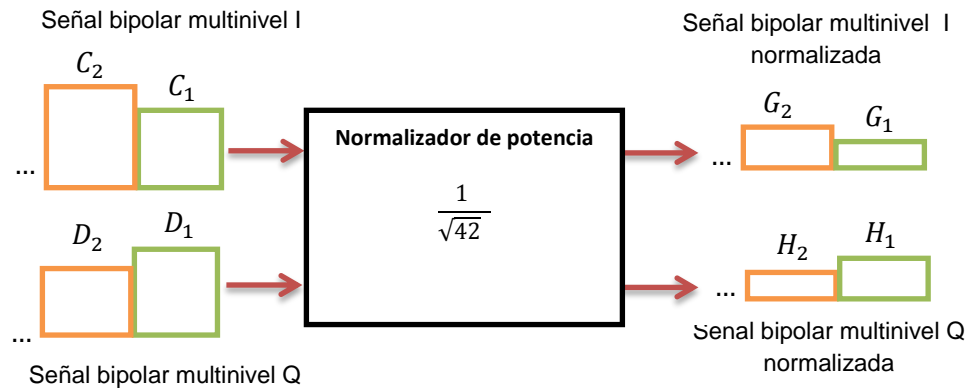


Figura 2.9. Normalizador de potencia, modulación 64 QAM



Bloque 5. Canal AWGN complejo. Se asume un canal de comunicaciones no limitado en banda que adiciona ruido AWGN complejo a las señales provenientes del normalizador de potencia. Las señales de salida del canal corresponden a las señales bipolares multinivel I y Q normalizadas afectadas por el correspondiente ruido (ver figuras 2.10 y 2.11).

Figura 2.10. Canal AWGN complejo, modulación 16 QAM

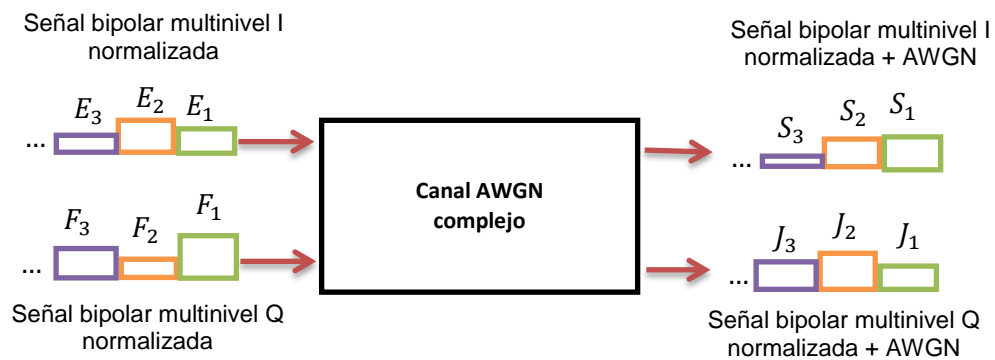
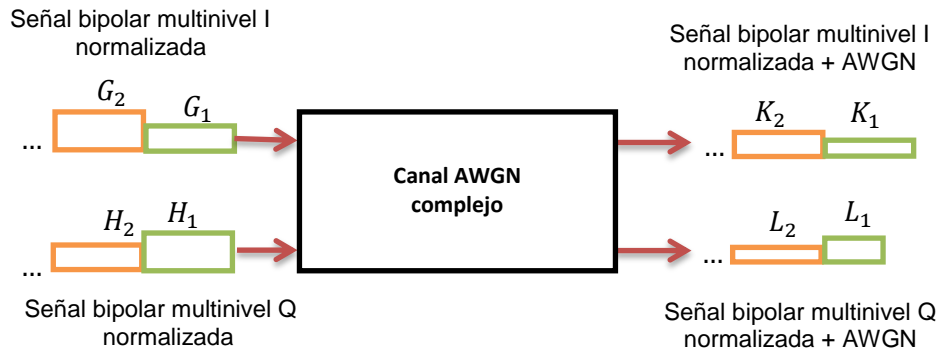


Figura 2.11. Canal AWGN complejo, modulación 64 QAM



Bloque 6. Desnormalizador de potencia. Para revertir la normalización de potencia hecha en el transmisor, el desnormalizador de potencia divide las señales después del canal por el correspondiente factor de normalización (ver figuras 2.12 y 2.13).

Figura 2.12. Desnormalizador de potencia, modulación 16 QAM

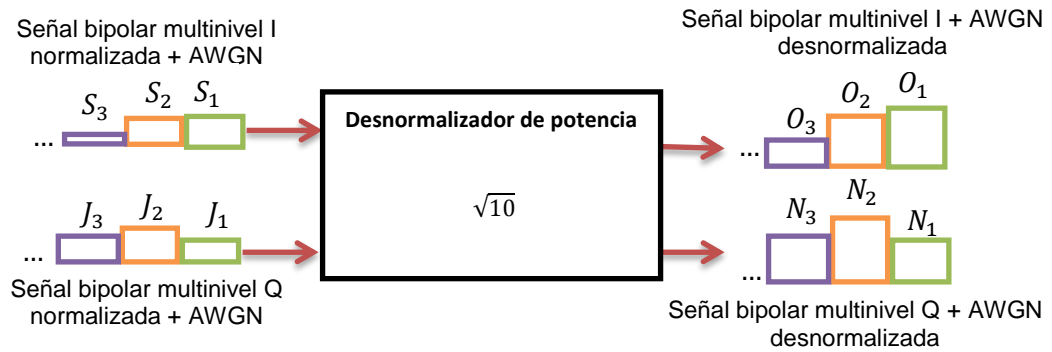
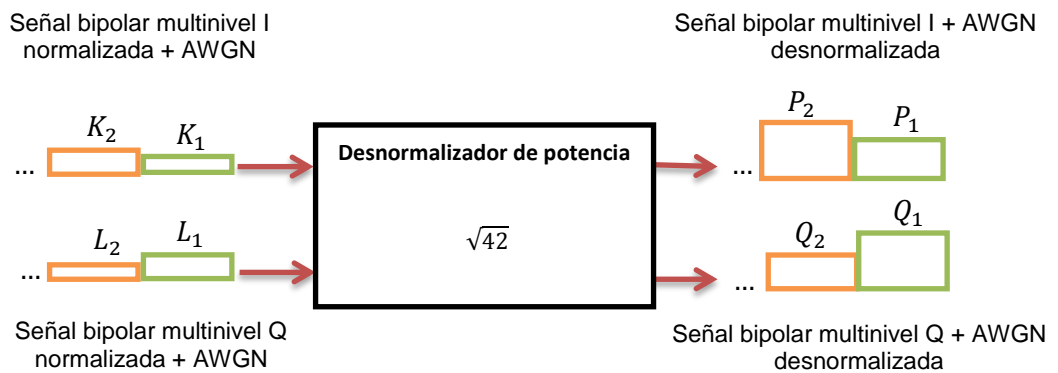


Figura 2.13. Desnormalizador de potencia, modulación 64 QAM



Bloque 7. Decisor. A partir de un criterio de decisión determinado establece el valor del símbolo complejo que le corresponde a cada par de niveles de las señales de las ramas I y Q proporcionadas por el normalizador de potencia (ver figuras 2.14 y 2.15)

Figura 2.14. Decisor, modulación 16 QAM

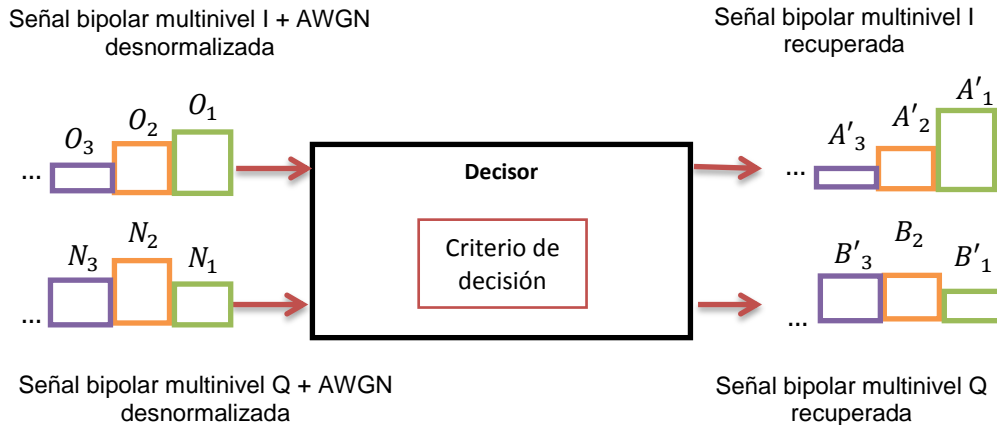
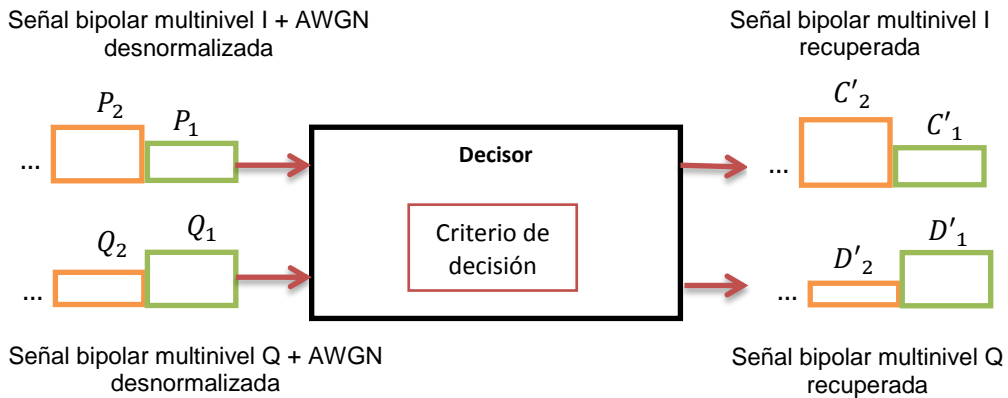


Figura 2.15. Decisor, modulación 64 QAM



Bloque 8. Conversor de M a 2 niveles. Identifica cada uno de los símbolos complejos que representan conjuntamente las señales bipolares multinivel I y Q recuperadas y genera por cada símbolo un conjunto de $M/2$ bits tanto en la rama I como en la rama Q, de acuerdo al mapeo utilizado en el modulador (ver figuras 2.16 y 2.17).

Figura 2.16. Conversor de M a 2 niveles, modulación 16 QAM

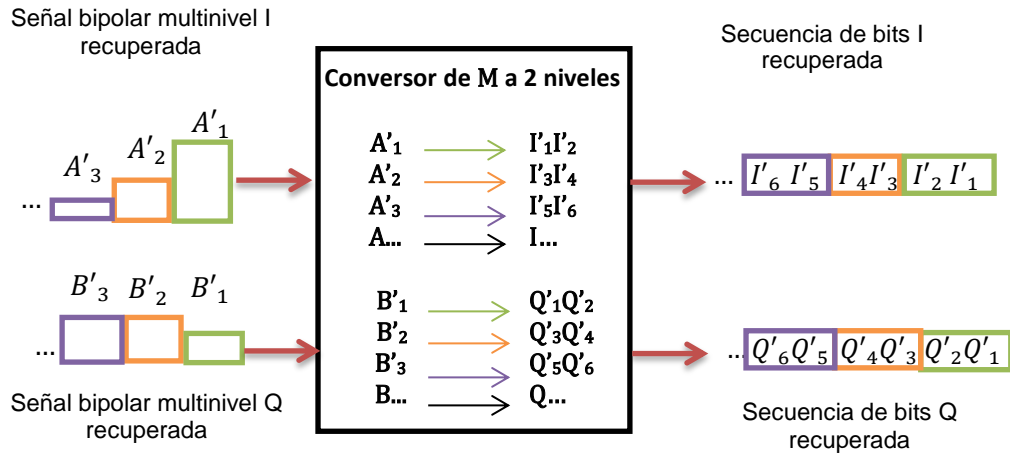
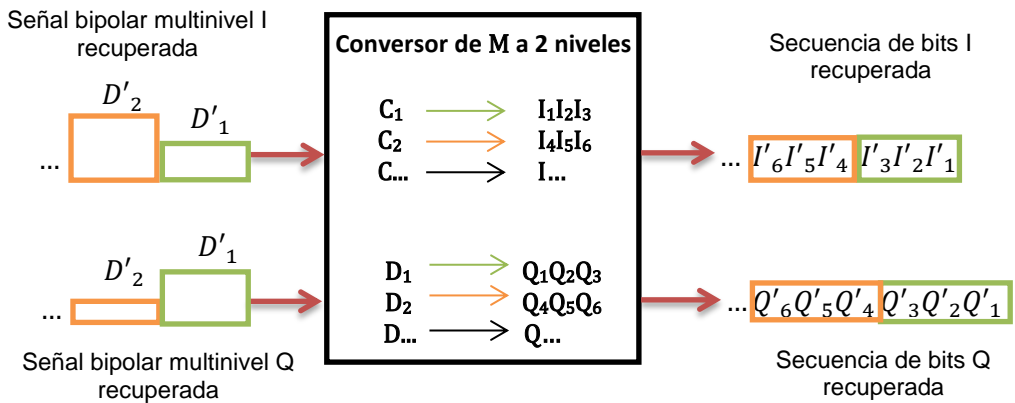


Figura 2.17. Conversor de M a 2 niveles, modulación 64 QAM



Bloque 9. Multiplexor de datos. Agrupa los conjuntos de $M/2$ bits de cada una de las ramas I y Q de manera alternada, empezando por el primer conjunto de la rama I (ver figuras 2.18 y 2.19). La señal de salida del Multiplexor de datos corresponde a la secuencia de bits recuperada o también denominada señal demodulada.

Figura 2.18. Multiplexor de datos, modulación 16 QAM

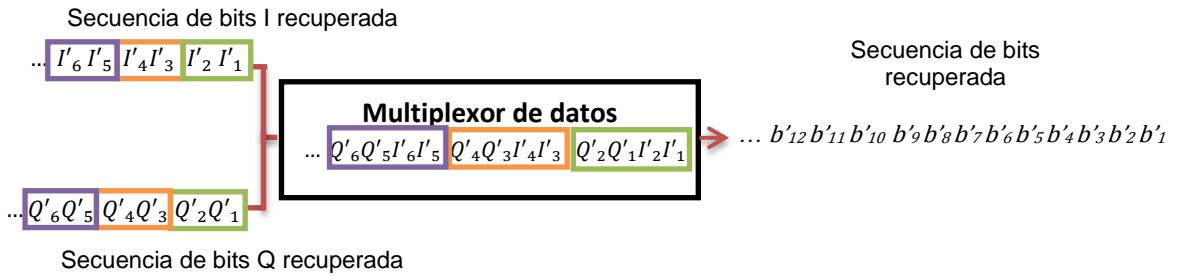
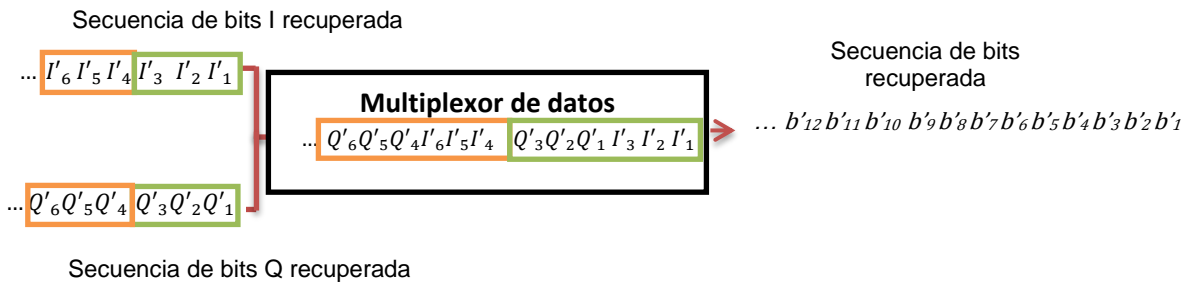
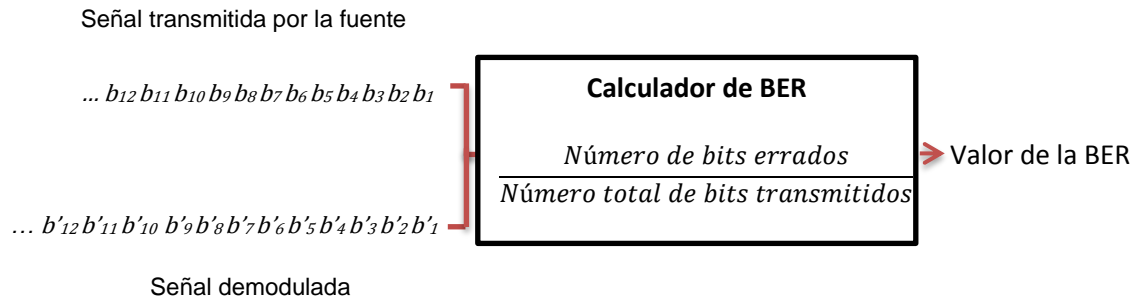


Figura 2.19. Multiplexor de datos, modulación 64 QAM



Bloque 10. Calculador de BER. Recibe la señal transmitida por la fuente de información y la señal demodulada, sincroniza estas señales, las compara a nivel de bit, calcula el número de bits errados de la señal demodulada y lo divide entre la cantidad total de bits transmitidos, obteniendo como resultado la BER del sistema básico de comunicaciones para un nivel de ruido determinado por el valor de la relación energía de bit a densidad espectral de potencia de ruido (E_b/N_o) (ver figura 2.20).

Figura 2.20. Calculador de BER



2.1.3 Simulación. Con las consideraciones conceptuales de la sección 2.1.2 y tras una breve elección de las herramientas pertinentes hardware y software, en esta sección se presenta tanto el diseño del modelo del sistema básico de comunicaciones banda base con modulación 16/64 QAM para simulación como la validación del correcto funcionamiento del diseño simulado, por medio de la comparación con las curvas de desempeño ofrecidas por un diseño en Simulink® utilizado como referencia.

2.1.3.1 Selección de herramientas. Las herramientas software y hardware seleccionadas para la elaboración del presente trabajo de grado se relacionan a continuación:

- **Herramientas Software.** De las herramientas software mencionadas en la sección 1.2.1.2, la más adecuada para el desarrollo del presente trabajo de grado fue System Generator del paquete de diseño del Ambiente de Software Integrado¹⁶ (ISE, *Integrated Software Environment*), su elección se debió a que: no se requiere experiencia previa con metodologías de diseño sobre FPGAs de Xilinx, se habilita el uso del entorno de modelado de Simulink® para el diseño basado en FPGA y los diseños son procesados por el entorno de Simulink® utilizando un conjunto de bloques específico de Xilinx.
- **Herramientas Hardware.** Teniendo en cuenta que System Generator realiza diseños sobre FPGAs de Xilinx, la herramienta hardware escogida fue el kit de entrenamiento Nexys 3 de Digilent, el cual cuenta con un FPGA Spartan 6 de Xilinx. Adicionalmente, se consideró el equilibrio entre cuatro factores concretos: disponibilidad en el mercado colombiano, precio, capacidad de procesamiento y periféricos de salida.

2.1.3.2 Elementos utilizados en el modelo en System Generator. A partir del modelo de referencia planteado en la sección 2.1.2.1 (ver figura 2.2) se construyó el modelo en System Generator del sistema básico de comunicaciones banda base con modulación 16/64 QAM. A continuación se detallan los elementos utilizados en el modelo en System Generator, los cuales cuentan con los siguientes parámetros comunes:

Arithmetic Type. Permite escoger el tipo de dato de salida. Los tipos de datos disponibles son: booleano, sin signo y con signo en complemento a dos.

Number of bits. Especifica el número de bits con el que se representa el valor de salida del elemento.

Binary point. Indica la posición del punto decimal del valor de salida.

Quantization and Overflow. *Quantization* cuenta con dos opciones, truncar y redondeo. La primera para eliminar los bits que se encuentran a la derecha del Bit Menos Significativo (LSB, *Least Significant Bit*) representable y la segunda que permite redondear al valor más cercano representable. Para *Overflow*, System Generator ofrece las opciones recortar y saturar. Recortar para descartar los bits que están a la izquierda del Bit Más Significativo (MSB, *Most Significant Bit*) representable y saturar que permite tomar el valor positivo más grande y el valor negativo más pequeño.

¹⁶ Paquete de diseño ISE: Herramienta software de Xilinx para el análisis y compilación (síntesis) de diseños HDL permitiendo realizar también análisis de tiempos, simular la reacción de un diseño a diferentes estímulos, y configurar el hardware a través de una interfaz de programación. De los componentes disponibles en este paquete, los más relevantes para el presente trabajo de grado fueron: System Generator, para el diseño de alto nivel soportado por Simulink®; Project Navigator para el análisis de restricciones del diseño y cálculo de recursos hardware requeridos e iMPACT para la programación del FPGA.

Sample Period. Permite especificar un período de muestreo particular, aunque por lo general, cada elemento detecta la tasa de muestreo de entrada para generar la tasa de muestro de salida.

Latency. Es el número de ciclos de reloj en que se retarda la señal de salida, por lo cual debe ser un valor entero positivo.

Elemento 1. System Generator Token. Mostrado en la figura 2.21, funciona como un panel de control para administrar los parámetros del sistema modelado y la simulación. También es usado para invocar el generador de código de programación del FPGA.

Todo modelo de Simulink® que contenga algún elemento de Xilinx debe contar por lo menos con un *System Generator Token*, a partir del cual es posible especificar cómo debe ser configurado el generador de código y la simulación.

Los parámetros específicos a configurar son:

- *Compilation.* Especifica el tipo de resultado de compilación que debe producirse cuando se invoca el generador de código.
- *Part.* Define el FPGA a usarse.
- *Synthesis tool.* Especifica la herramienta a usarse para sintetizar el diseño. Las posibilidades son Synplicity's Synplify Pro, Synplify, y Xilinx's XST.
- *Hardware Description Language.* Especifica el lenguaje HDL a usarse para la compilación del diseño. Las posibilidades son VHDL and Verilog.
- *Target directory.* Especifica dónde debe guardar System Generator los resultados de compilación.
- *Project type.* Selecciona el tipo de archivo de proyecto que debe generarse, si *Project Navigator* o *PlanAhead*.
- *FPGA clock period(ns).* Dado en nanosegundos, define el periodo del reloj del sistema. El valor debe ser entero. El período es pasado a una herramienta de implementación de Xilinx a través de un archivo de restricciones, donde es usado como el periodo de restricción global.
- *Clock pin location.* Define el pin de ubicación del reloj del hardware. Esta información es pasada a la herramienta de implementación de Xilinx a través de un archivo de restricción. Esta opción no debe ser especificada si el diseño en System Generator va a ser incluido como parte de un diseño más grande.
- *Simulink system period(sec).* Define el período del sistema de Simulink® en unidades de segundos. El período de sistema de Simulink® es el máximo común divisor de los periodos de muestreo que aparecen en el modelo. Estos periodos de muestreo se

establecen explícitamente en los cuadros de diálogo de cada uno de los elementos, y se heredan de un elemento a otro.

Figura 2.21. *System Generator Token*

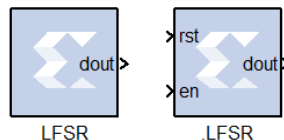


Elemento 2. Registro de desplazamiento con realimentación lineal. Mostrado en la figura 2.22, implementa un Registro de Desplazamiento con Realimentación Lineal (LFSR, *Linear Feedback Shift Register*), compuesto por una cadena de biestables, el cual puede generar largas secuencias pseudoaleatorias de unos y ceros (secuencias m).

Los parámetros específicos a configurar son:

- *Type* Fibonacci o Galois. Este campo especifica la estructura de realimentación. Fibonacci tiene una compuerta XOR (o XNOR) al inicio del registro de desplazamiento que opera los derivadores¹⁷ de realimentación junto con el valor de entrada del primer biestable. Galois cuenta con una compuerta XOR (o XNOR) por cada derivador, la cual realiza operaciones a partir de los datos provenientes del derivador y datos provenientes del último biestable, ubicado en la salida del registro de desplazamiento.
- *Gate type XOR o XNOR*. Este campo especifica la compuerta usada por las señales de realimentación.
- *Number of bits in LFSR*. Especifica el número de biestables del registro de desplazamiento.
- *Feedback polynomial*. Este campo especifica el polinomio generador de la secuencia pseudoaleatoria, y con ello, los puntos de ubicación de cada derivador en el registro de desplazamiento, el valor debe ingresarse en hexadecimal con comillas sencillas.
- *Initial Value*. Especifica el valor de semilla a partir del cual el *LFSR* inicia la secuencia pseudoaleatoria. El valor inicial no puede ser ceros cuando se ha escogido la compuerta tipo XOR y no puede ser unos cuando se ha escogido XNOR, de ser así el *LFSR* no funciona.

Figura 2.22. Elemento *LFSR*

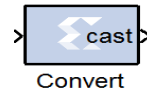


¹⁷ Las salidas de cada biestable que influyen en la entrada del registro de desplazamiento (realimentación) se denominan derivadores y se constituyen en cada uno de los valores del polinomio generador.

Elemento 3. Convert. Mostrado en la figura 2.23, cambia la cuantificación de un número pero no el valor. Este elemento puede alterar el número de bits usado para representar un número y puede ser utilizado para convertir del tipo sin signo al tipo con signo y viceversa. Frecuentemente es usado para limitar el número de bits de la parte decimal del número de salida de una operación de multiplicación.

Este elemento no cuenta con parámetros específicos.

Figura 2.23. Elemento *Convert*

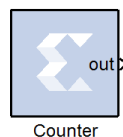


Elemento 4. Counter. Mostrado en la figura 2.24, implementa un contador que puede correr libremente a lo largo del rango establecido para tal efecto y después del cual, reinicia el conteo. La salida del elemento puede ser del tipo sin signo o con signo, y se le puede dar un valor inicial o bien proporcionarle un puerto por el cual cargar un valor determinado. También se puede configurar el tamaño de paso que incrementa el contador.

Los parámetros específicos a configurar son:

- *Counter type.* Especifica el modo de conteo, ya sea libre o limitado.
- *Initial value.* Indica el valor en el que comienza a contar.
- *Count to value.* En el modo limitado indica hasta qué valor debe contar. Si se configura con *Inf* cuenta hasta el máximo valor posible. Este número no puede ser el mismo que aparece en *Initial value*.
- *Step.* Tamaño del paso con el que se cambia de un valor a otro en el conteo.
- *Count direction.* Indica si es un contador ascendente o descendente, o bien se habilita un puerto para que se indique si va a ser ascendente o descendente.

Figura 2.24. Elemento *Counter*



Elemento 5. Mux. Mostrado en la figura 2.25, implementa un multiplexor donde la entrada de selección es del tipo sin signo y el número de entradas de datos va desde 2 hasta 32.

El parámetro específico a configurar es:

- *Number of inputs.* Especifica el número de buses de datos de entradas.

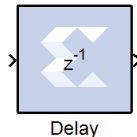
Figura 2.25. Elemento *Mux*



Elemento 6. Delay. Mostrado en la figura 2.26, es un registro de desplazamiento, de longitud configurable, que induce un retardo sobre la señal. Permite añadir latencia al diseño. Los datos de la entrada aparecerán a la salida después de un determinado número de períodos de la señal de reloj.

Este elemento no cuenta con parámetros específicos.

Figura 2.26. Elemento *Delay*

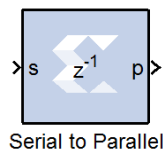


Elemento 7. Serial to Parallel. Mostrado en la figura 2.27, toma los datos en serie de entrada y los convierte a datos en paralelo a la salida, creando un único dato de salida para cada determinada cantidad de datos de entrada. Cuenta con un retardo mínimo de un periodo de muestreo, con respecto al periodo de muestreo de la señal de salida. Se produce un error cuando el número de bits de salida no se puede dividir por el número de bits de entrada.

El parámetro específico a configurar es:

- *Input order.* Asignación del primer bit de salida al LSB o MSB de la secuencia.

Figura 2.27. Elemento *Serial to Parallel*



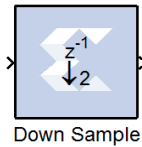
Elemento 8. Down Sample. Mostrado en la figura 2.28, reduce la tasa de muestreo en el sitio donde se coloque, para lo cual muestrea cada cierto intervalo de tiempo, dando la posibilidad de elegir si se desea la muestra del inicio o del final del intervalo; el valor de cada muestra es mantenido hasta que la próxima muestra es tomada.

Los parámetros específicos a configurar son:

- *Sampling Rate.* Tiene que ser un entero mayor o igual que dos. Es la relación entre la tasa de entrada y la de salida, es esencialmente un divisor de tasa de muestreo.

- **Sample.** Puede ser Primer Valor de la Muestra o Último Valor de la Muestra, indicando si se desea la muestra inicial o la muestra final de la ventana de muestreo respectivamente.

Figura 2.28. Elemento *Down Sample*



Elemento 9. Mcode. Mostrado en la figura 2.29, permite el modelado de estructuras de control hardware a partir del lenguaje de programación de Matlab®; para este fin se genera un *script* que es ejecutado dentro del *Mcode*.

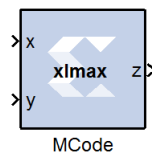
Este elemento no es adecuado para códigos que describen una operación algorítmica tal como un filtro de Respuesta Finita al Impulso (FIR, *Finite Impulse Response*) o matriz inversa, pero en cambio proporciona un método conveniente y eficiente para la aplicación de máquinas de estado y condiciones complejas de multiplexación.

El Mcode soporta las siguientes operaciones en lenguaje de programación Matlab®:

- Asignación de sentencias.
- Sentencias simples y compuestas *if/else/eselseif*.
- Sentencias *switch*.
- Expresiones aritméticas que involucran solo suma y resta.
- Multiplicación.
- División por una potencia de dos.

Este elemento no cuenta con parámetros específicos.

Figura 2.29. Elemento *Mcode*



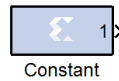
Elemento 10. Constant. Mostrado en la figura 2.30, genera un valor constante que puede ser sin signo, con signo o booleano.

Los parámetros específicos a configurar son:

- *Constant Value.* Indica el valor de la constante.

- *Sampled Constant*. Permite asociar un período determinado a la salida constante del elemento. Esta opción suele ser usada para interconectar el elemento *Constant* con algunos otros que requieren esta característica.

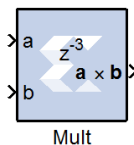
Figura 2.30. Elemento *Constant*



Elemento 11. *Mult*. Mostrado en la figura 2.31, implementa un multiplicador de dos entradas.

Este elemento no cuenta con parámetros específicos.

Figura 2.31. Elemento *Mult*



Elemento 12. *White Gaussian Noise Generator*. El elemento Generador de Ruido Blanco Gaussiano (WGNG, *White Gaussian Noise Generator*), mostrado en la figura 2.32, genera ruido AWGN usando una combinación del algoritmo de Box-Muller y el teorema del límite central¹⁸.

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 en (0,1). Posteriormente dichos valores se almacenan en Memorias de Solo Lectura (ROM, *Read Only Memory*) direccionados con variables aleatorias uniformes.

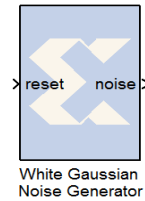
Las salida de cuatro subsistemas paralelos Box-Muller son promediados para obtener una PDF gaussiana, generado así ruido normalizado $N(0,1)$. La latencia total del generador es de diez ciclos de reloj y el puerto de salida del ruido es un número de doce bits con signo y siete bits después del punto binario.

El parámetro específico a configurar es:

- *Seed*. Define el patrón de ruido del WGNG.

¹⁸ 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.

Figura 2.32. Elemento *WGNG*

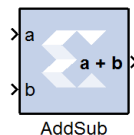


Elemento 13. *AddSub*. Mostrado en la figura 2.33, implementa un sumador o un restador a la vez o, de acuerdo a la configuración, permite implementar las dos operaciones de manera secuencial de acuerdo a alguna condición externa.

El parámetro específico a configurar es:

- *Operation*. Suma, resta o suma/resta. Cuando la operación suma/resta está seleccionada aparece un puerto de entrada que controla la operación a realizar; la señal que controla este puerto debe ser de tipo booleano. Si está a nivel alto el elemento actúa como restador, en caso contrario como sumador.

Figura 2.33. Elemento *AddSub*

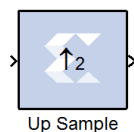


Elemento 14. *Up Sample*. Mostrado en la figura 2.34, incrementa la tasa de muestreo en el punto en el que se coloque. La entrada es sobremuestreada L veces en un período de muestreo de la señal de entrada. La salida es constituida ya sea por L muestras de la señal de entrada, o por una muestra de la señal de entrada y $L-1$ ceros.

Los parámetros específicos a configurar son:

- *Sampling rate*. Tiene que ser un entero mayor o igual a dos. Este valor es la relación entre el período de muestreo a la salida y a la entrada del elemento. Por ejemplo, si este valor es dos, indica que la tasa de muestreo de salida es dos veces la de entrada.
- *Copy samples*. Permite elegir qué hacer con las muestras adicionales debidas al incremento de la tasa de muestreo. Si se selecciona esta casilla se copian las muestras de entradas, en caso contrario se insertan ceros.

Figura 2.34. Elemento *Up Sample*

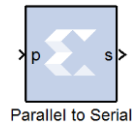


Elemento 15. *Parallel to Serial*. Mostrado en la figura 2.35, toma los datos de entrada en paralelo y los separa a la salida para transmitirlos en serie, generando por cada dato de entrada un conjunto de datos de salida en serie. Se puede configurar el orden de salida para que empiece por el LSB o por el MSB.

El parámetro específico a configurar es:

- *Output order*. Indica si se empieza a transmitir primero el LSB o el MSB.

Figura 2.35. Elemento *Parallel to Serial*

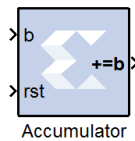


Elemento 16. *Accumulator*. Mostrado en la figura 2.36, implementa un acumulador de suma o de resta, donde el tamaño de la salida y la entrada deben tener la misma cantidad de bits de la parte entera y decimal y deben ser del mismo tipo.

El parámetro específico a configurar es:

- *Operation*. Suma o resta.

Figura 2.36. Elemento *Accumulator*



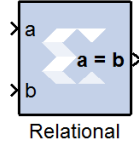
Elemento 17. *Relational*. Mostrado en la figura 2.37, implementa un operador relacional lógico. Las comparaciones permitidas son:

- Igual ($r = t$).
- Distinto ($r \neq t$).
- Menor que ($r < t$).
- Mayor que ($r > t$).
- Menor o igual que ($r \leq t$).
- Mayor o igual que ($r \geq t$).

El parámetro específico a configurar es:

- *Comparison*. Indica la comparación a realizar.

Figura 2.37. Elemento *Relational*

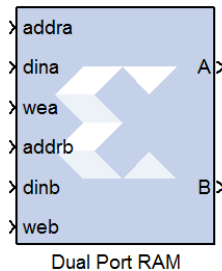


Elemento 18. Dual Port Ram. Mostrado en la figura 2.38, implementa una Memoria de Acceso Aleatorio (RAM, *Random Access Memory*). Cuenta con dos conjuntos independientes de puertos para la lectura y escritura simultánea. Puertos independientes de habilitación para dirección, datos y escritura permiten el acceso compartido a un único espacio de memoria. Por defecto, cada conjunto de puertos tiene un puerto de salida y tres puertos de entrada: dirección, datos de entrada, y habilitación de escritura. Opcionalmente, se puede agregar un puerto de habilitación y uno de señal de reseteo síncrono a cada conjunto de puertos de entrada.

Los parámetros específicos a configurar son:

- *Depth*. Indica el tamaño de la memoria (el número de palabras que se pueden guardar), debe ser un entero positivo.
- *Initial value vector*. Indica el contenido inicial de la memoria. Cuando la longitud del vector excede el tamaño de la memoria, los valores con índice mayor se ignoran. Cuando es menor, las posiciones superiores de la memoria se inicializan en cero.

Figura 2.38. Elemento *Dual Port Ram*

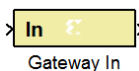


Elemento 19. Gateway In. Mostrado en la figura 2.39, son las entradas de la parte Xilinx del diseño en Simulink®. Convierten los tipos de datos enteros, dobles y de punto fijo de Simulink® en los tipos de datos de punto fijo para System Generator. Cada elemento define un puerto de entrada en el código de HDL generado por System Generator.

El parámetro específico a configurar es:

- *IOB Pad Location*, e.g. ('MSB',..., 'LSB'). En este campo se indican, de más significativo a menos significativo, la localización de los pines (hardware) de entrada asociados a este elemento.

Figura 2.39. Elemento *Gateway In*



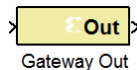
Elemento 20. Gateway Out. Mostrado en la figura 2.40, son las salidas de la parte Xilinx del diseño en Simulink®. Este elemento convierte los datos de punto fijo a datos de tipo doble.

Cada elemento define un puerto de salida en el código HDL generado, o bien se usa simplemente como un punto de prueba que después será eliminado del diseño hardware.

Los parámetros específicos a configurar son:

- *Translate into Output Port.* Representa el elemento como un puerto en la generación del código HDL.
- *Specify IOB Location Constraint.* Habilita un campo para la descripción de la localización de los pines de salida para estos puertos.

Figura 2.40. Elemento *Gateway Out*



2.1.3.3 Modelo en System Generator. El modelo planteado en System Generator para el sistema básico de comunicaciones banda base con modulación 16/64 QAM consideró dos configuraciones diferentes de parámetros: uno para la modulación 16 QAM y otro para la modulación 64 QAM; la figura 2.41 muestra el modelo en System Generator.

Las figuras 2.42 a 2.45 permiten observar con mayor detalle el diseño de cada uno de los bloques del modelo en *System Generator*, donde se representan cada uno de los bloques planteados en el modelo de referencia de la figura 2.2.

Figura 2.4.1. Modelo en System Generator

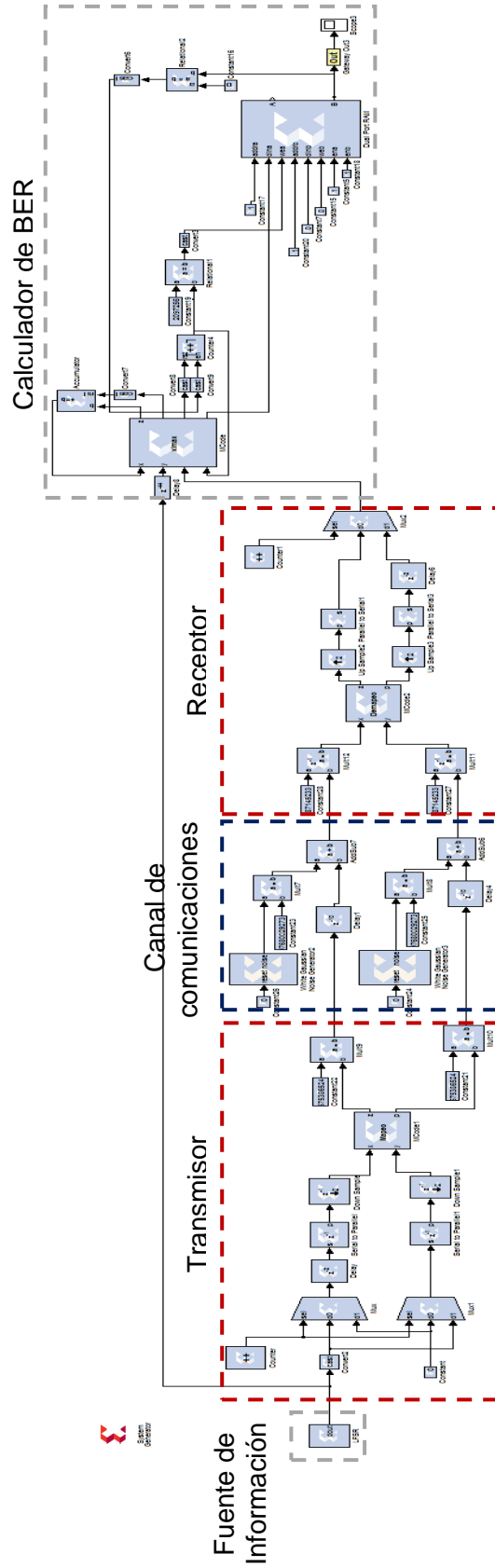


Figura 2.42. Fuente de información y transmisor, modelo en *System Generator*

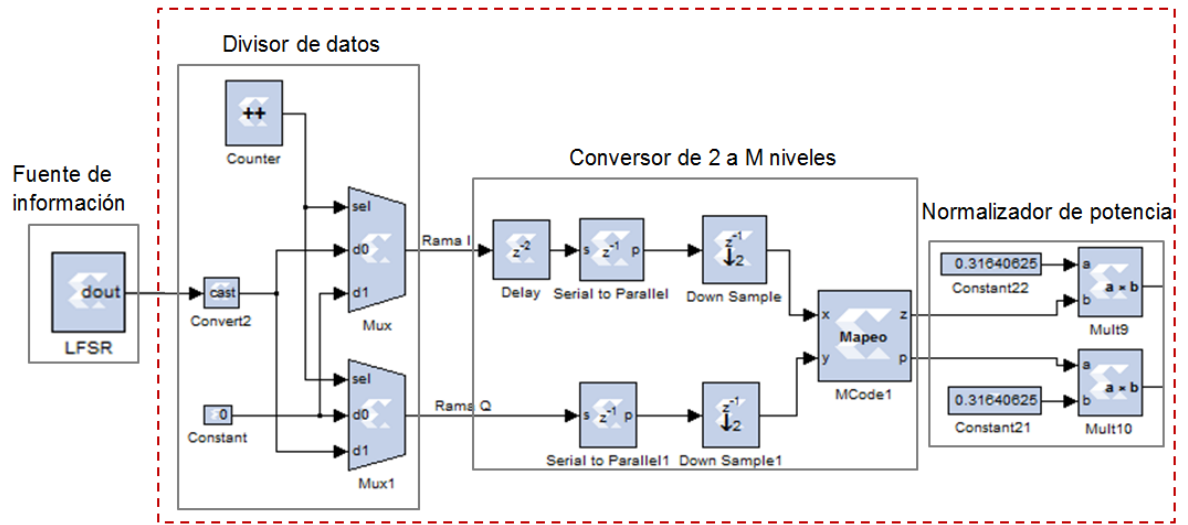


Figura 2.43. Canal de comunicaciones, modelo en *System Generator*

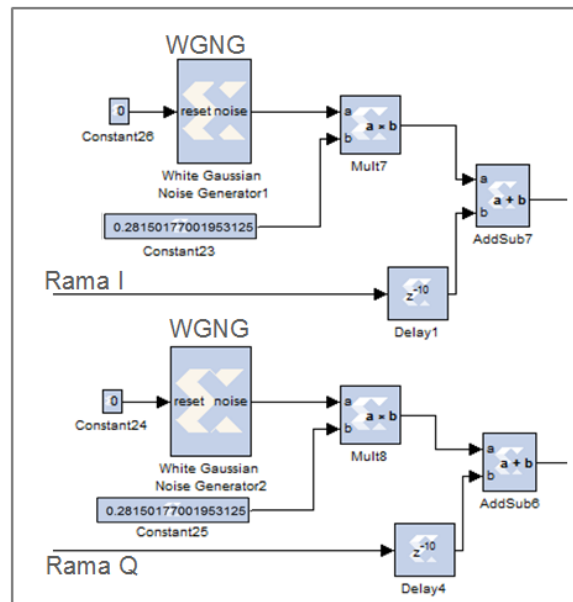


Figura 2.44. Receptor, modelo en System Generator

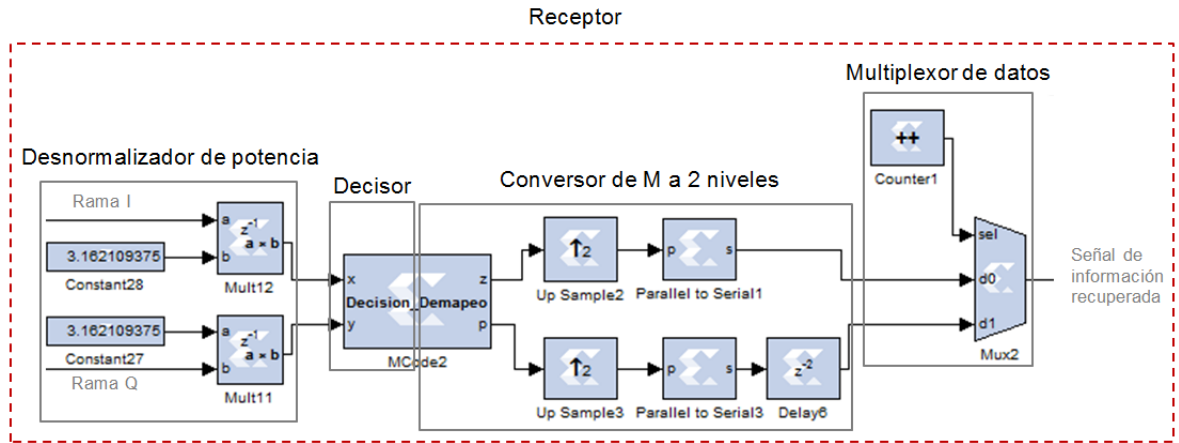
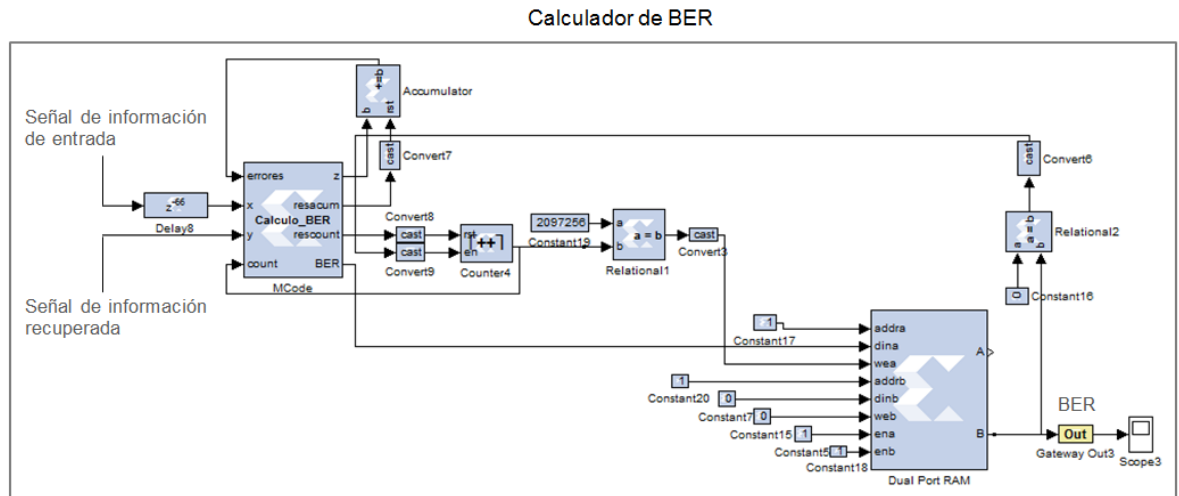


Figura 2.45. Calculador de BER, modelo en System Generator



Para ilustrar el proceso general del modelo en System Generator, se recurre a la representación gráfica de las señales de entrada y salida¹⁹ de cada uno de los componentes del sistema básico de comunicaciones banda base con modulación 16/64 QAM, a partir de una señal de información predefinida. A continuación se detalla el diseño de cada uno de los bloques del modelo:

Bloque 1. Fuente de información. Consta de un *LFSR* (ver figura 2.46.) encargado de generar una secuencia de bits pseudoaleatoria, que representa la señal de información a transmitir.

¹⁹ Para cada figura se consideraron las señales de entrada y salida del respectivo bloque; donde, la notación utilizada corresponde a: E para señal de entrada, S para señal de salida y R para la señal de referencia.

Las figuras 2.47 y 2.48 muestran la señal de información predefinida (generada por la fuente de información). Dicha señal corresponde a una secuencia de bits con periodo de bit (T_b) igual a un segundo para facilitar la visualización de los bits. Con el fin de ilustrar la manera como el sistema básico de comunicaciones procesa los bits, estos se nombran y representan de la misma forma que se hizo en el modelado (ver tabla 2.1).

Figura 2.46. Fuente de información

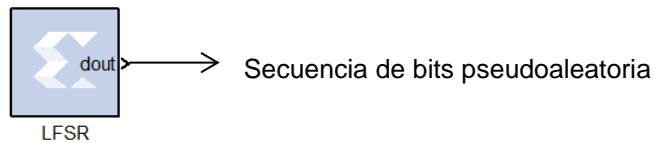


Figura 2.47. Secuencia de bits predefinida, modulación 16 QAM

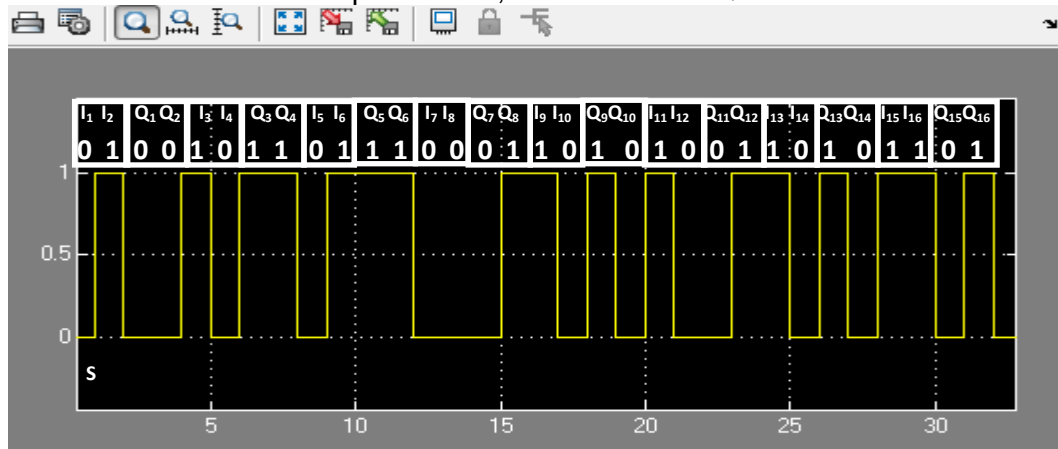
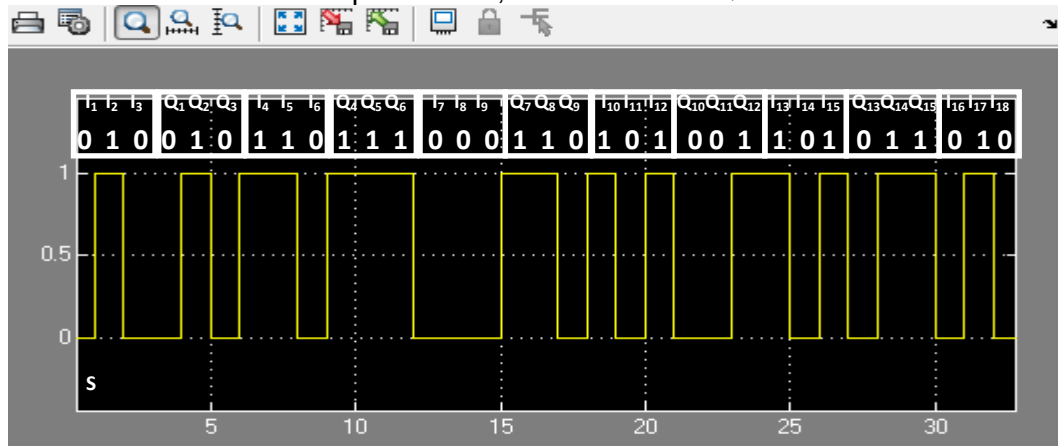


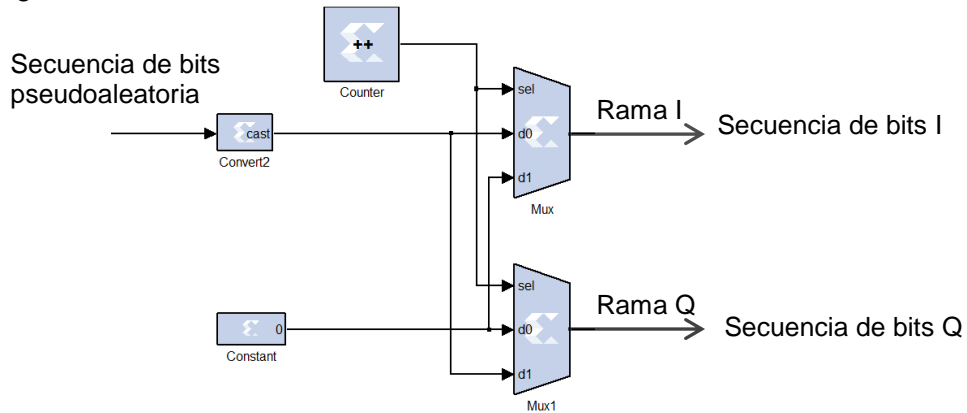
Figura 2.48. Secuencia de bits predefinida, modulación 64 QAM



Bloque 2. Divisor de datos. La construcción del divisor de datos requiere dos *Mux* para la generación de las ramas I y Q, un *Counter* que habilita las salidas del divisor de datos, un *Constant* con valor igual a cero en las entradas d1 y d0 de los *Mux* y *Mux1*

respectivamente y un *Convert* para el cambio del tipo de dato de entrada a los multiplexores dado que la fuente de información y los *Mux* manejan dos tipos de datos diferentes (ver figura 2.49).

Figura 2.49. Divisor de datos



En razón al diseño planteado, el divisor de datos se comporta como un multiplexor controlado por un *Counter*, con dos entradas (secuencia de bits pseudoaleatoria y constante igual a cero) y dos salidas (rama I y rama Q).

El *Counter* genera una señal periódica que varía entre los valores cero y uno partiendo de un valor inicial igual a cero, cuando la señal del *Counter* se encuentra en cero el divisor de datos envía $M/2$ bits de la secuencia pseudoaleatoria por la rama I y $M/2$ bits con valores iguales a cero por la rama Q, de la misma manera, cuando el la señal del *Counter* se encuentra en uno el divisor de datos envía $M/2$ bits de la secuencia pseudoaleatoria por la rama Q y $M/2$ bits con valores iguales a cero por la rama I. Así, solo una rama recibe bits de la fuente de información a la vez, mientras la otra rama recibe ceros (ver tabla 2.2).

Tabla 2.2. Tabla de verdad del divisor de datos

<i>Counter</i>	Modulación 16 QAM		Modulación 64 QAM	
	Secuencia de bits I	Secuencia de bits Q	Secuencia de bits I	Secuencia de bits Q
0	$I_1 I_2$	00	$I_1 I_2 I_3$	000
1	00	$Q_1 Q_2$	000	$Q_1 Q_2 Q_3$
0	$I_3 I_4$	00	$I_4 I_5 I_6$	000
1	00	$Q_3 Q_4$	000	$Q_4 Q_5 Q_6$
0	$I_5 I_6$	00	$I_7 I_8 I_9$	000
...

La tabla de verdad del divisor de datos permite observar claramente dos aspectos relevantes:

- El divisor de datos introduce secuencias de ceros en las ramas I y Q ajenas a la secuencia de bits pseudoaleatoria transmitida por la fuente de información. Esta cantidad de ceros, introducidos cada $M/2$ bits, corresponde a $M/2$ y la duración de cada cero es igual a T_b .

- La secuencia de bits Q posee un retardo igual a $T_b(M/2)$ respecto a la secuencia de bits I, debido a que la rama I recibe $M/2$ bits de la fuente de información primero.

Dado que los anteriores aspectos corresponden a efectos no deseados del diseño del divisor de datos, estos son eliminados por el conversor de 2 a M niveles. Las figuras 2.50 y 2.51 muestran las respectivas señales de entrada y salida del divisor de datos.

Figura 2.50. Señal de entrada y señales de salida de las ramas I y Q del divisor de datos, modulación 16 QAM

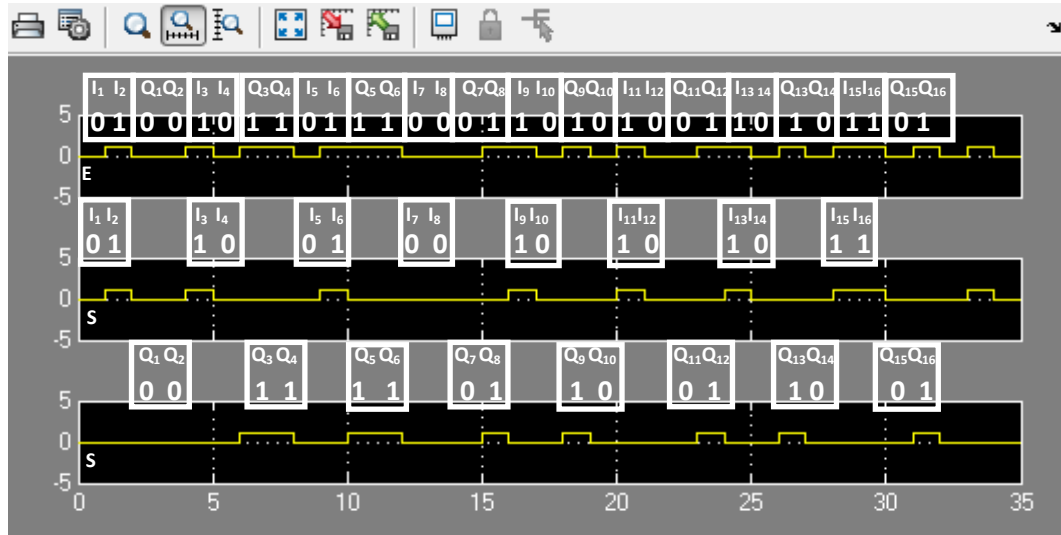
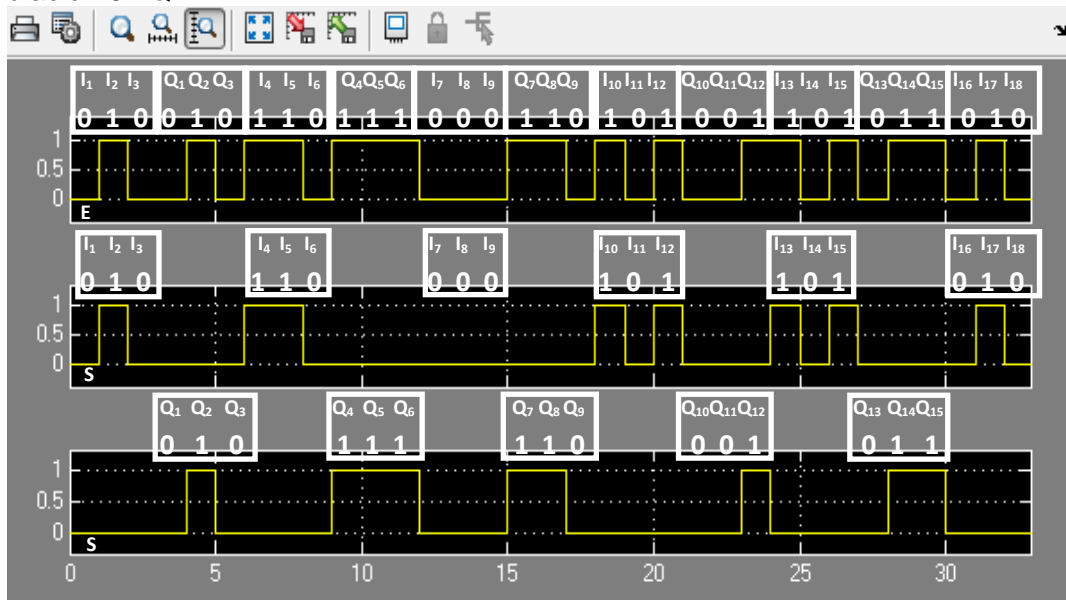
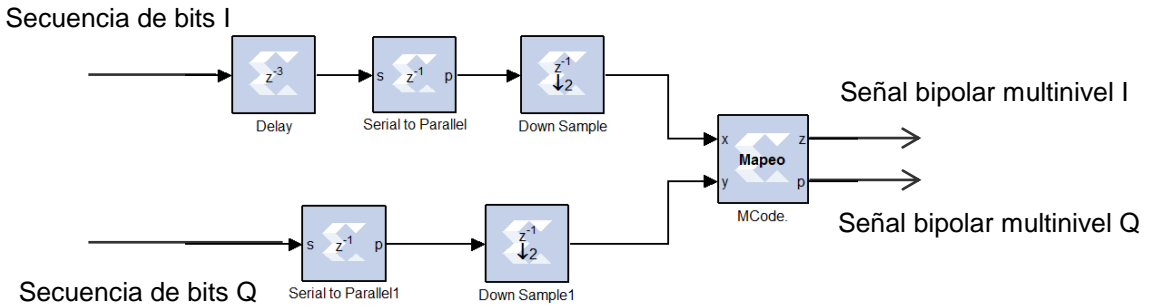


Figura 2.51. Señal de entrada y señales de salida de las ramas I y Q del divisor de datos, modulación 64 QAM



Bloque 3. Conversor de 2 a M niveles. El diseño del conversor de 2 a M niveles implica el uso de un *Delay* que sincroniza la secuencia de bits I con la secuencia de bits Q, un *Serial to Parallel* y un *Down Sample* en cada rama (I y Q) que eliminan los ceros introducidos por el divisor de datos, y un *Mcode* que posteriormente mapea las señales de las ramas I y Q generando las señales bipolares multinivel correspondientes (ver figura 2.52).

Figura 2.52. Conversor de 2 a M niveles



En cuanto se obtienen las señales de salida del divisor de datos, la secuencia de bits I es sincronizada respecto a la secuencia de bits Q a través de un *Delay* con valor igual a $T_b(M/2)$ en la rama I, permitiendo ubicar los bits de la rama I en la misma posición de tiempo de los correspondientes bits de la rama Q (ver figuras 2.53 y 2.54).

Figura 2.53. Señales de las ramas I y Q sincronizadas, modulación 16 QAM

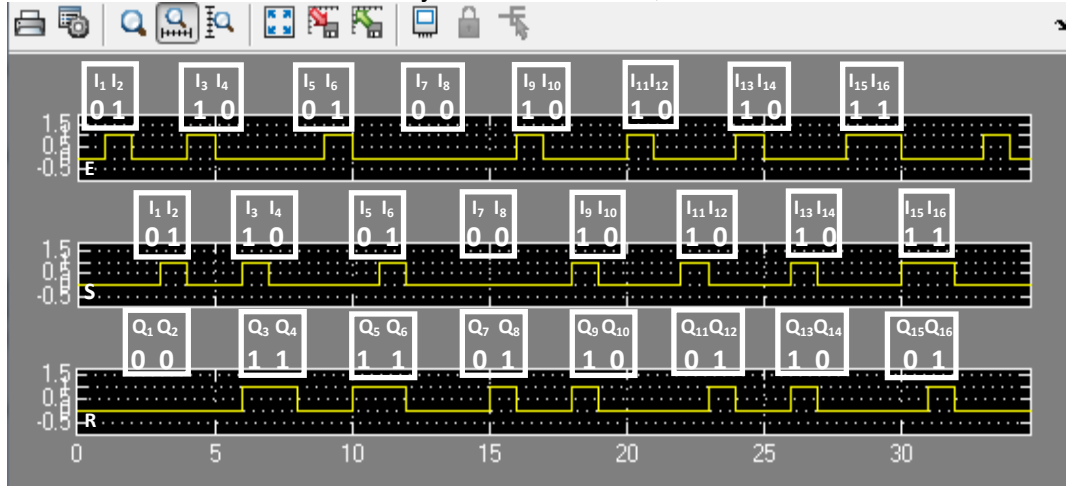
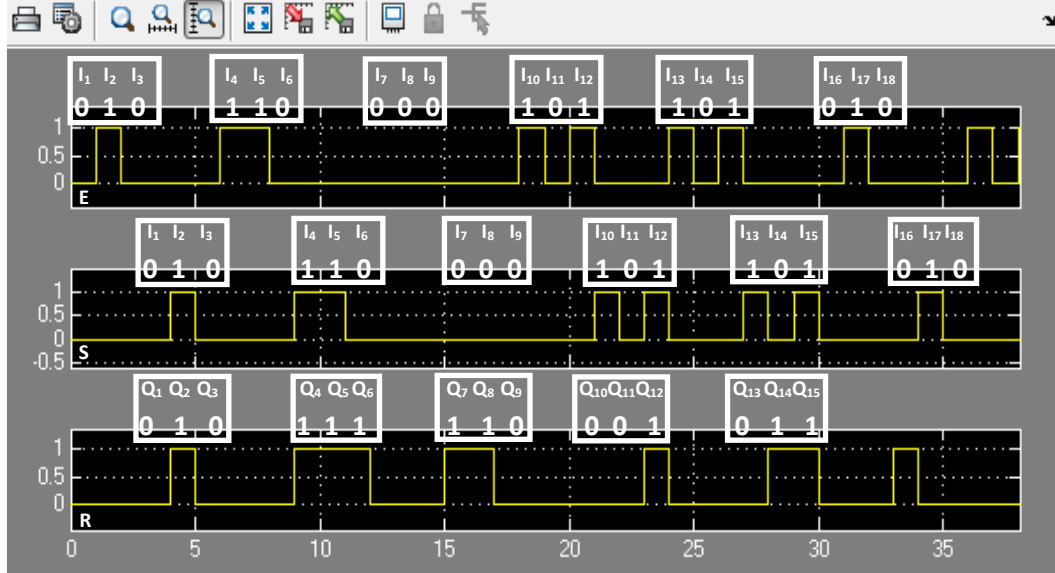


Figura 2.54. Señales de las ramas I y Q sincronizadas, modulación 64 QAM



Con el fin de eliminar los ceros introducidos por el divisor de datos, se utiliza un *Serial to Parallel* y un *Down Sample* tanto en la rama I como en la rama Q. El *Serial to Parallel* de cada rama (ver figura 2.52) recibe la correspondiente secuencia de bits, agrupa esta secuencia en conjuntos de $M/2$ bits y representa cada conjunto con un valor decimal que dura $T_b(M/2)$, donde el bit del extremo izquierdo corresponde al MSB. Como resultado se obtiene una señal unipolar multinivel con periodo de muestreo igual a $T_b(M/2)$. Dado que el *Serial to Parallel* genera un retardo de procesamiento igual a $T_b(M/2)$, la señal de salida se encuentra retrasada con respecto a la señal de entrada (ver figuras 2.55 a 2.58).

Figura 2.55. Señales de entrada y salida del *Serial to Parallel* de la rama I, modulación 16 QAM

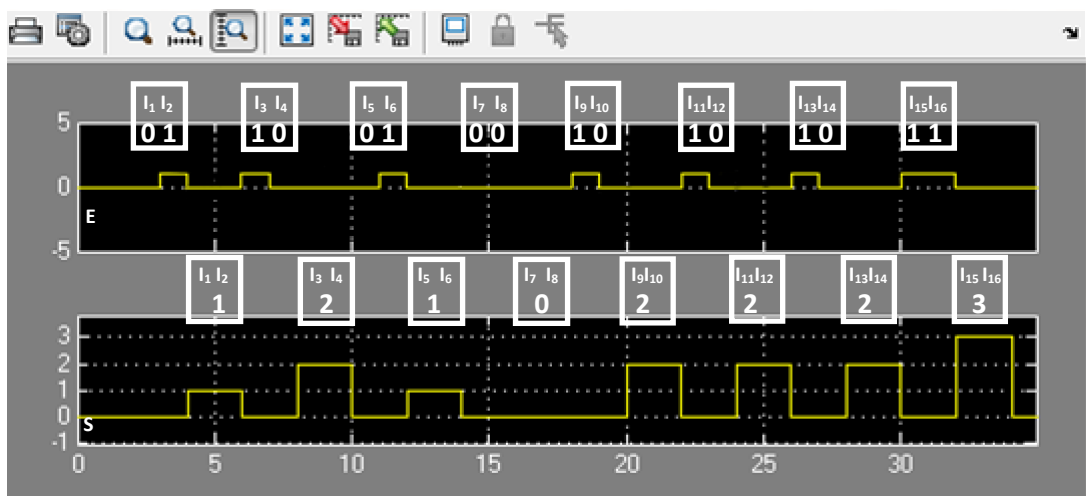


Figura 2.56. Señales de entrada y salida del *Serial to Parallel* de la rama Q, modulación 16 QAM

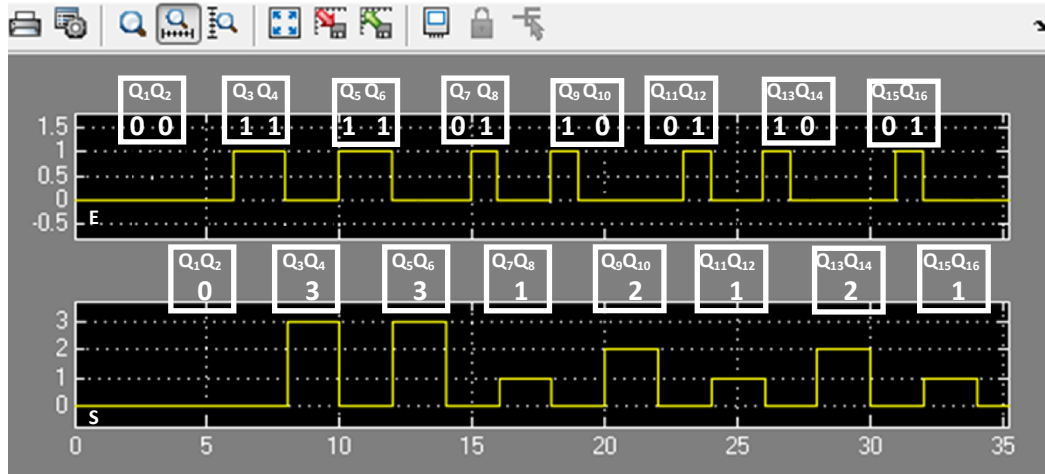


Figura 2.57. Señales de entrada y salida del *Serial to Parallel* de la rama I, modulación 64 QAM

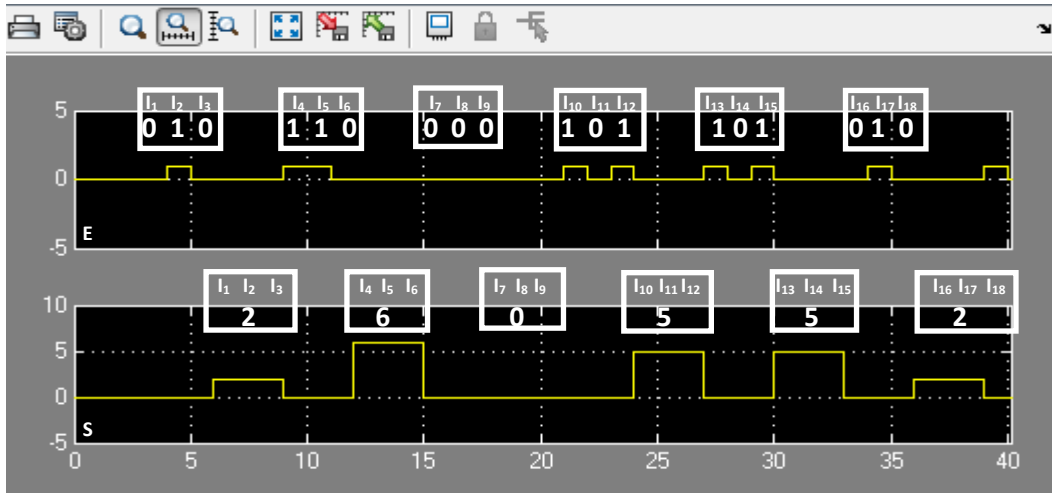
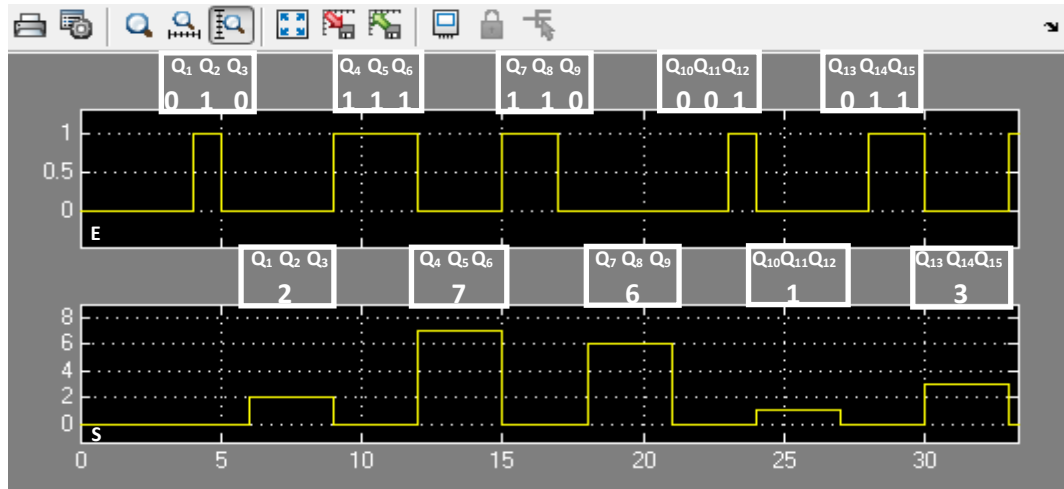


Figura 2.58. Señales de entrada y salida del *Serial to Parallel* de la rama Q, modulación 64 QAM



El *Down Sample* de cada rama (I y Q) (ver figura 2.52) toma la señal unipolar multinivel correspondiente y reduce su tasa de muestreo a la mitad; dado que el periodo de muestreo de la señal corresponde a $T_b(M/2)$, el *Down Sample* muestrea la señal con un periodo igual a T_bM y mantiene el valor de cada muestra hasta que la próxima muestra es tomada; de este modo, las muestras obtenidas por el *Down Sample* corresponden únicamente a los valores decimales que representan los bits de la secuencia generada por la fuente de información, obteniendo una señal unipolar multinivel que no contiene los ceros introducidos por el divisor de datos (ver figuras 2.59 a 2.62).

Figura 2.59. Señales de entrada y salida del *Down Sample* de la rama I, modulación 16 QAM

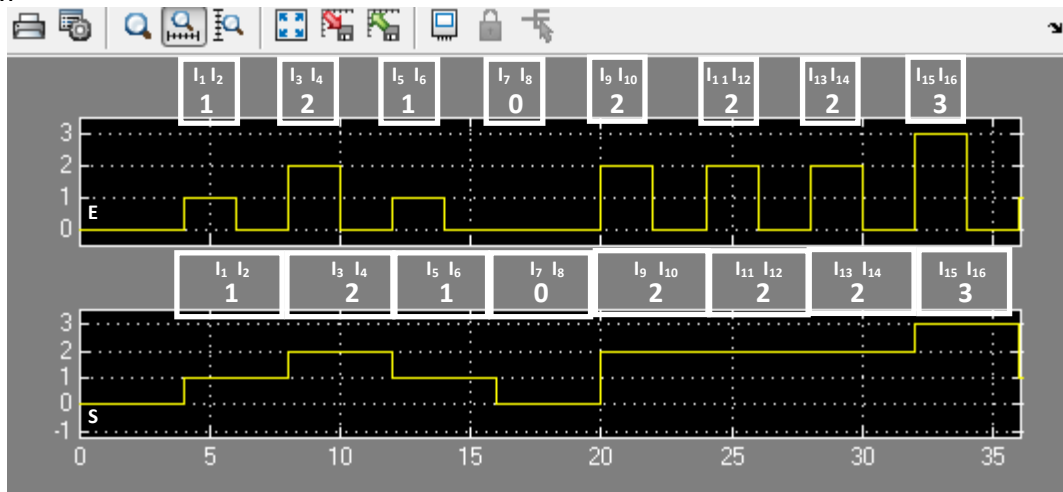


Figura 2.60. Señales de entrada y salida del *Down Sample* de la rama Q, modulación 16 QAM

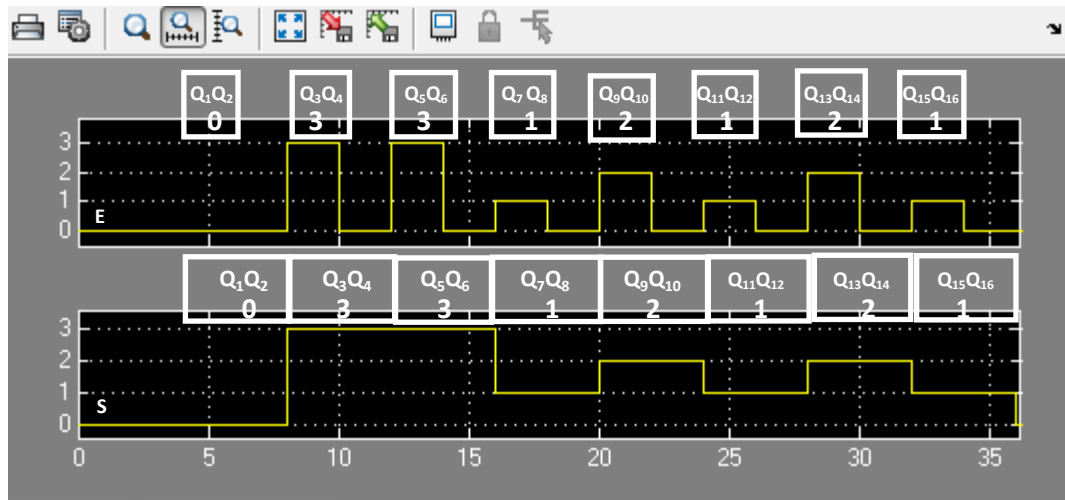


Figura 2.61. Señales de entrada y salida del *Down Sample* de la rama I, modulación 64 QAM

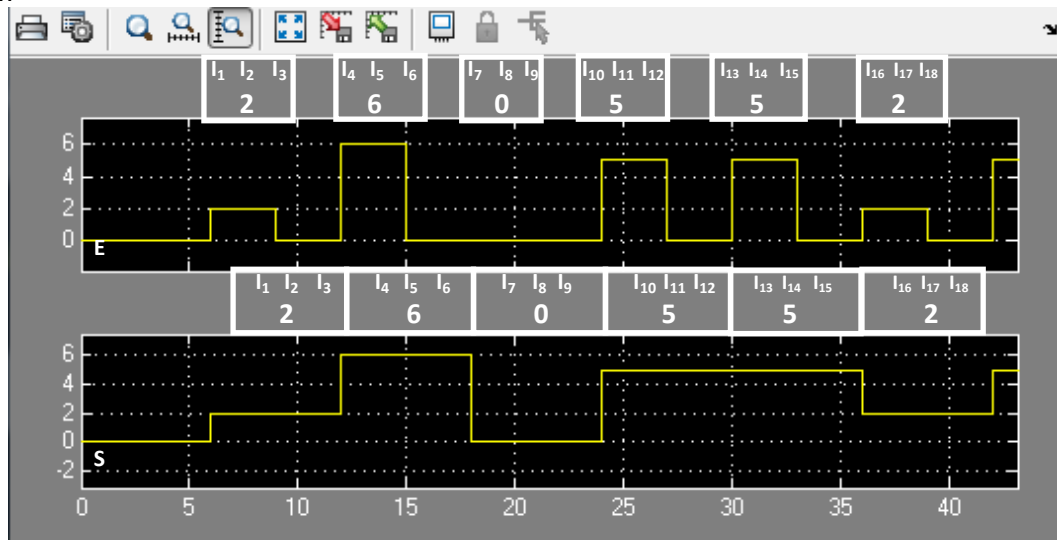
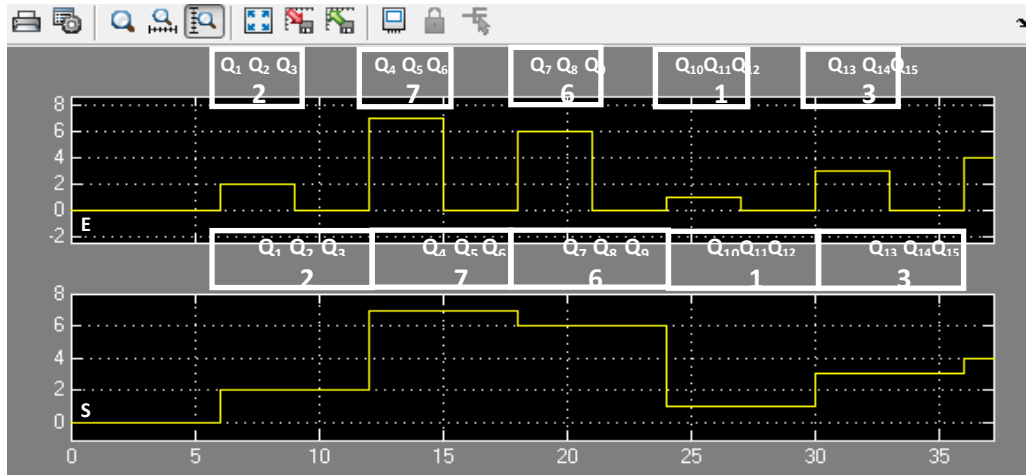


Figura 2.62. Señales de entrada y salida del *Down Sample* de la rama Q, modulación 64 QAM



Tan pronto como se obtienen las señales unipolares multinivel I y Q de los *Down Sample* de cada rama (I y Q), un *Mcode* (ver figura 2.52) identifica los bits que ingresan representados por estas señales, asocia a cada par de conjuntos de $M/2$ bits de las ramas I y Q el valor de un símbolo complejo de acuerdo al mapeo establecido (ver tablas 2.3 y 2.4) y genera una forma de onda rectangular tanto en la rama I como en la rama Q, por cada símbolo complejo; generando las señales bipolares multinivel I y Q que componen la señal compleja resultado de la modulación. Los niveles de amplitud de la señal compleja corresponden con los valores de cada símbolo complejo (ver figuras 2.63 a 2.66). Para realizar el anterior procedimiento, dentro del *Mcode* se desarrolla un código en lenguaje de programación Matlab®, basado en estructuras condicionales, que realiza las acciones correspondientes.

El mapeo establecido corresponde al mapeo Gray, sin embargo, en el plan de pruebas de la sección 3.1 se tiene en cuenta el mapeo Binario a efectos de comparar y analizar el desempeño del sistema básico de comunicaciones banda base con modulación 16/64 QAM ante variaciones en el tipo de mapeo.

Los diagramas de las constelaciones correspondientes al sistema básico de comunicaciones banda base con modulación 16/64 QAM, coinciden con los mostrados en las figuras 1.6 y 1.7.

Tabla 2.3. Mapeo Binario y Gray, modulación 16 QAM

Bits Identificados	Entrada	Mapeo Binario	Mapeo Gray	
$I_1 I_2 / Q_1 Q_2$	I/Q	I/Q	I	Q
11	3	3	1	-1
10	2	1	3	-3
01	1	-1	-1	1
00	0	-3	-3	3

Tabla 2.4. Mapeo Binario y Gray, modulación 64 QAM

Bits Identificados	Entrada	Mapeo Binario	Mapeo Gray	
$I_1 I_2 I_3 / Q_1 Q_2 Q_3$	I/Q	I/Q	I	Q
111	7	7	3	-3
110	6	5	1	-1
101	5	3	5	-5
100	4	1	7	-7
011	3	-1	-3	3
010	2	-3	-1	1
001	1	-5	-5	5
000	0	-7	-7	7

Figura 2.63. Señales de entrada y salida del *Mcode* de la rama I, mapeo Gray, modulación 16 QAM

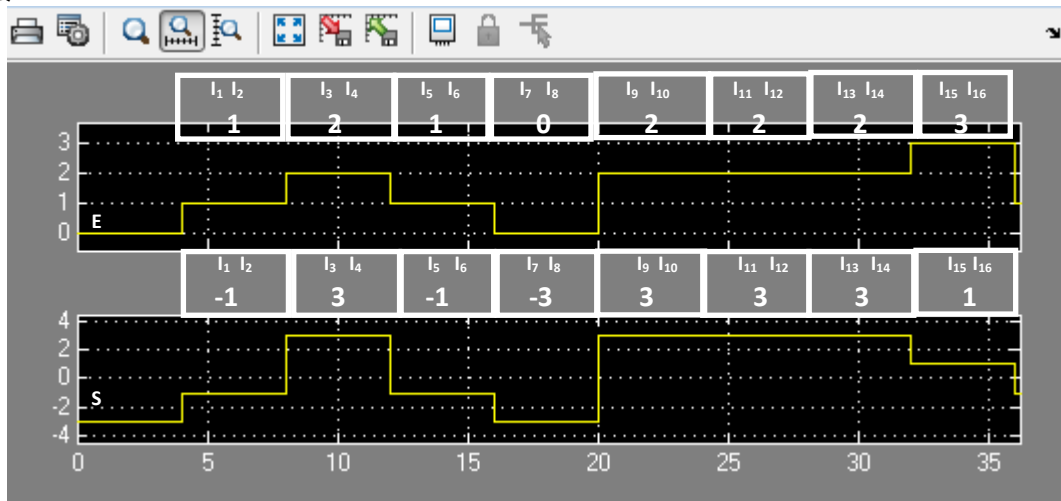


Figura 2.64. Señales de entrada y salida del *Mcode* de la rama Q, mapeo Gray, modulación 16 QAM

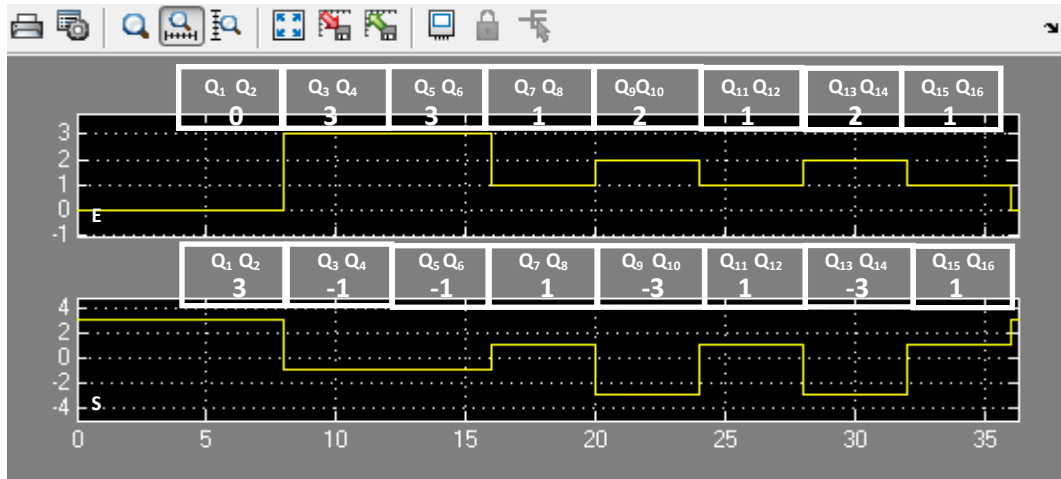


Figura 2.65. Señal de entrada y salida del *Mcode* de la rama I, mapeo Gray, modulación 64 QAM

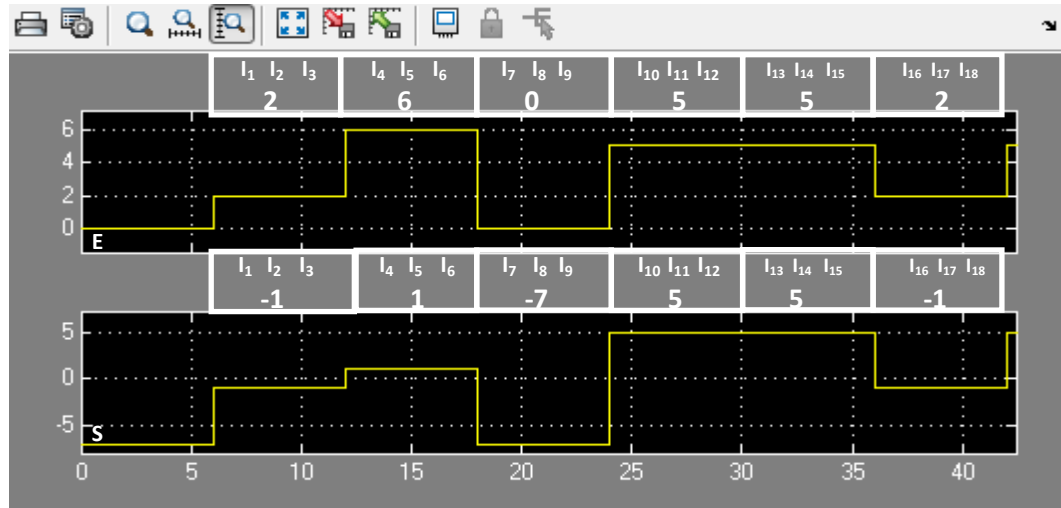
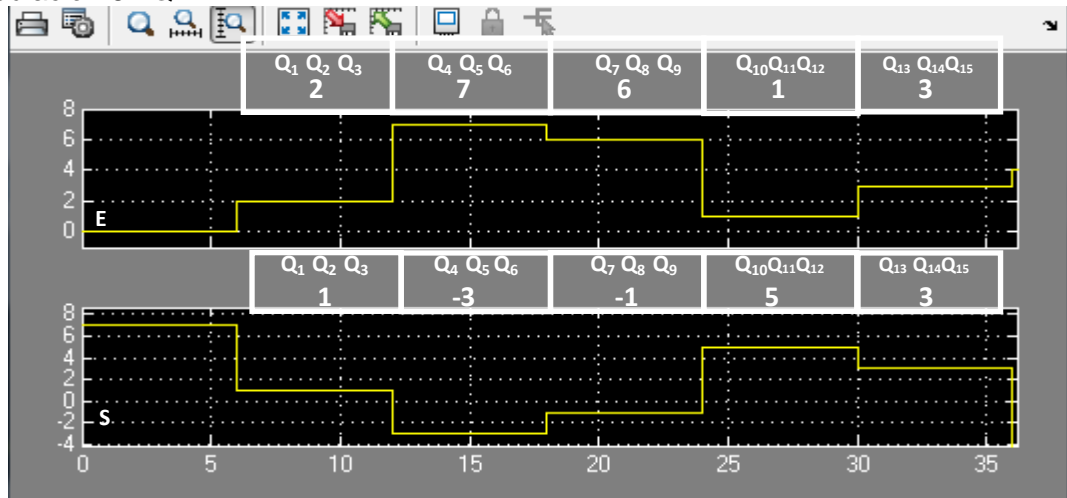
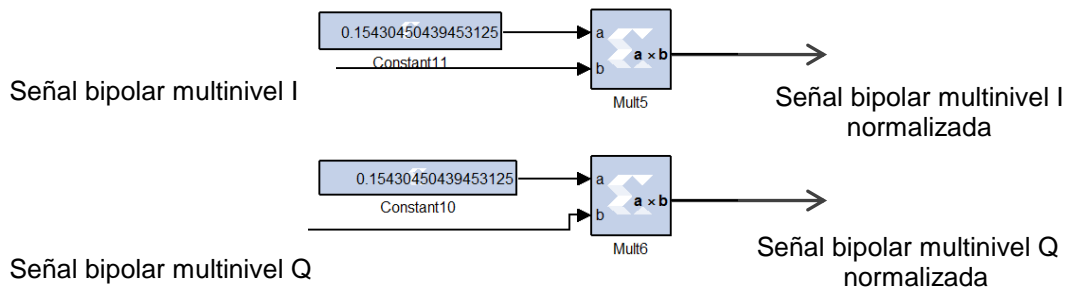


Figura 2.66. Señales de entrada y salida del *Mcode* de la rama Q, mapeo Gray, modulación 64 QAM



Bloque 4. Normalizador de potencia. La construcción de este bloque requiere en cada rama (I y Q) el uso de un *Constant* con un valor igual al factor de normalización y un *Mult* que realiza el producto entre este valor y la señal bipolar multinivel proporcionada por el *Mcode* del conversor de 2 a *M* niveles (ver figura 2.67).

Figura 2.67. Normalizador de potencia



En cuanto se obtienen las señales de salida del normalizador de potencia, la etapa del transmisor finaliza, obteniendo las señales bipolares multinivel I y Q con potencia normalizada (ver figuras 2.68 a 2.71).

Figura 2.68. Señales de entrada y salida del normalizador de potencia de la rama I, modulación 16 QAM

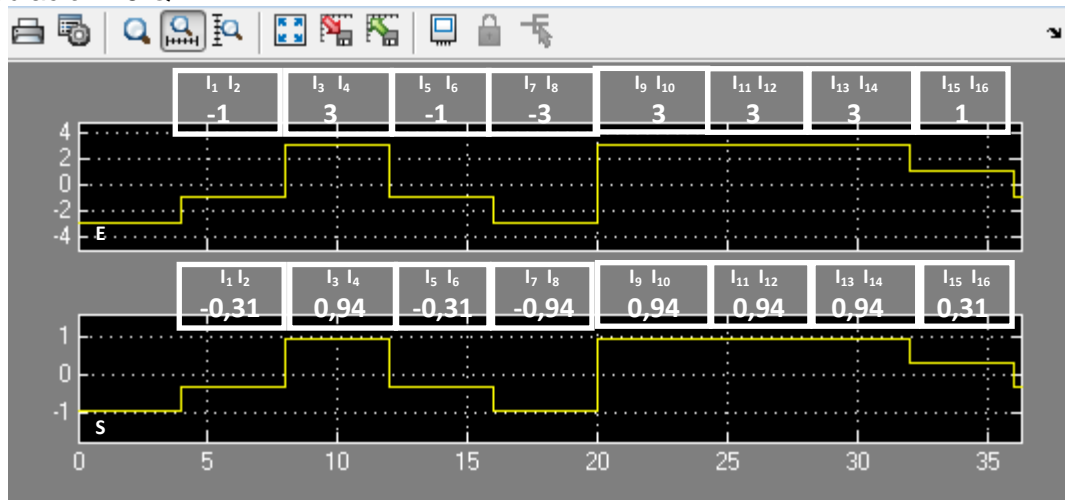


Figura 2.69. Señales de entrada y salida del normalizador de potencia de la rama Q, modulación 16 QAM

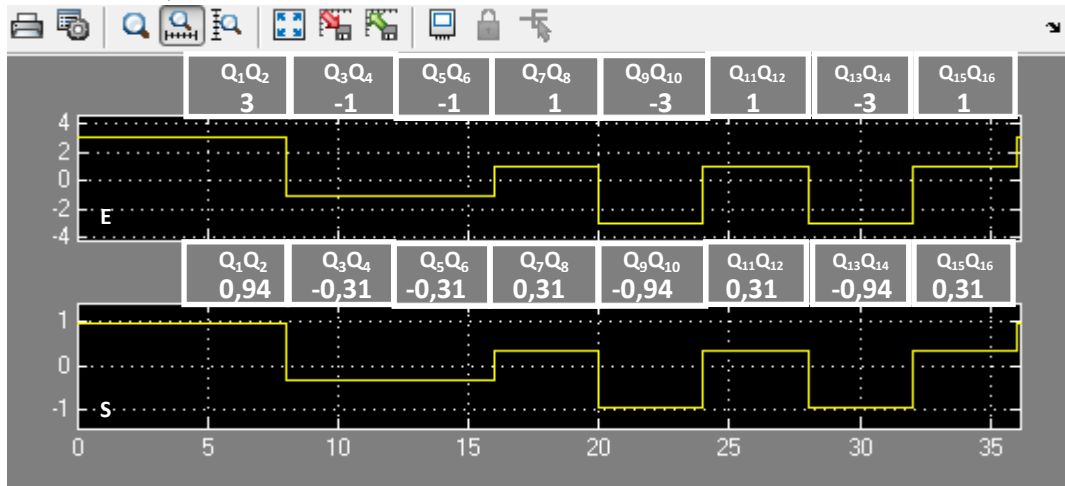


Figura 2.70. Señales de entrada y salida del normalizador de potencia de la rama I, modulación 64 QAM

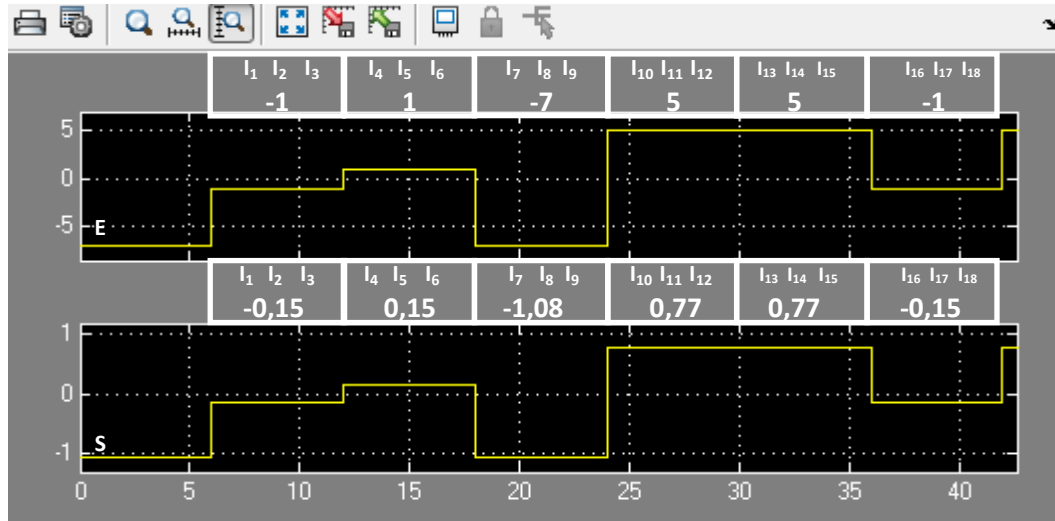
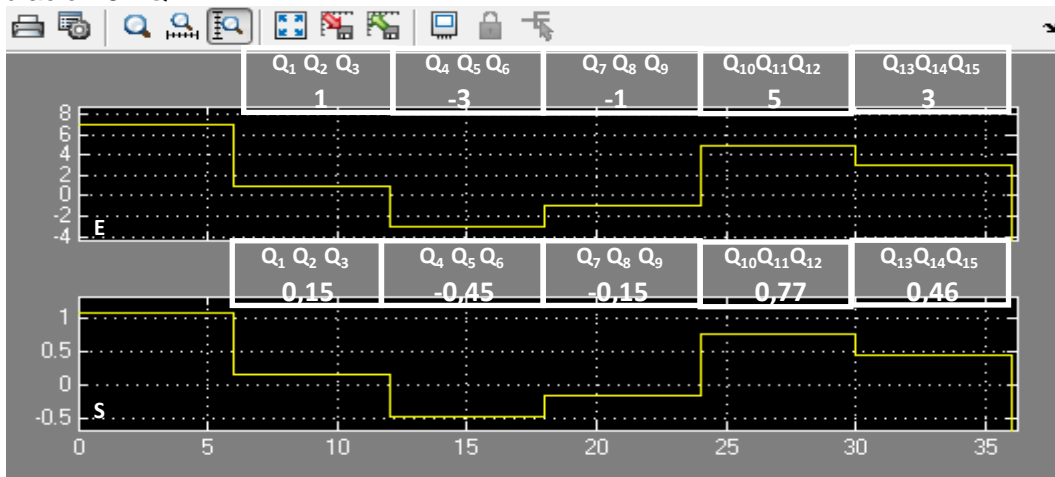


Figura 2.71. Señales de entrada y salida del normalizador de potencia de la rama Q, modulación 64 QAM

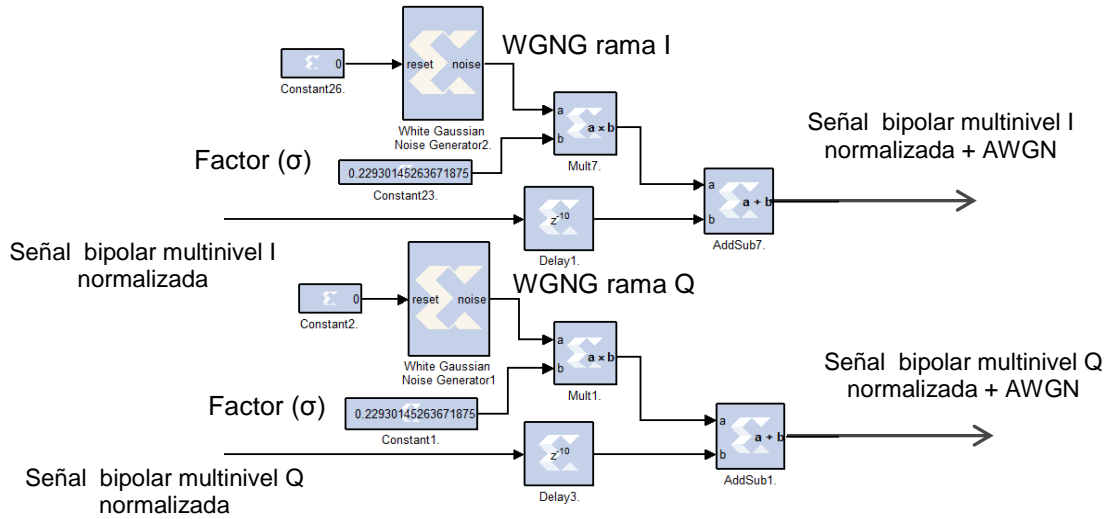


Bloque 5. Canal AWGN complejo. Ya que System Generator no ofrece un canal AWGN complejo, se diseñan dos canales AWGN no correlacionados, uno en la rama I y otro en la rama Q, que representan las componentes en fase (I) y cuadratura (Q) del ruido AWGN complejo.

El diseño de los dos canales AWGN no correlacionados requiere en cada rama (I y Q), un *WGNG* que genera ruido AWGN con un nivel de potencia normalizado, un *Constant* y un *Mult* que controlan la cantidad de ruido generada y un *AddSub* que adiciona el ruido generado a las señales bipolares multinivel I y Q normalizadas (ver figura 2.72)²⁰.

²⁰ Este diseño garantiza un canal AWGN complejo no limitado en banda.

Figura 2.72. Canal AWGN complejo



El único parámetro configurable en los WGNGs es el valor de una semilla (ver figura 2.32), este parámetro reviste suma importancia durante la simulación y posterior implementación, ya que para valores iguales de semilla, los WGNGs presentan comportamiento de correlación. Por lo tanto, si bien el ruido AWGN complejo es no correlacionado entre sus componentes real e imaginaria, el hecho de utilizar dos WGNGs separados para cada una de las ramas (I y Q) del canal no es suficiente para garantizar esta característica de no correlación.

Las figuras 2.73 y 2.74 muestran dos constelaciones a la salida del canal AWGN complejo: La primera donde las semillas de ambos WGNGs se configuraron con el mismo valor y la constelación presentó una dispersión con tendencia lineal causando que los símbolos transmitidos no se ubicaron alrededor del símbolo de referencia, sino sobre una línea recta; la segunda constelación corresponde a diferentes valores de semilla para cada WGNG, con lo cual la dispersión ubico los símbolos alrededor de cada símbolo de referencia.

Este comportamiento tiene lugar ya que al momento de configurar los mismos valores de semilla, los WGNGs generan los mismos patrones de ruido para las ramas I y Q dando como resultado la correlación entre las componentes del ruido. Por otro lado, cuando las semillas son diferentes, los patrones de ruido son diferentes en cada uno de los WGNGs, de modo que el ruido en las ramas es no correlacionado.

Figura 2.73. Diagrama de constelación señal modulada después del canal - semillas iguales

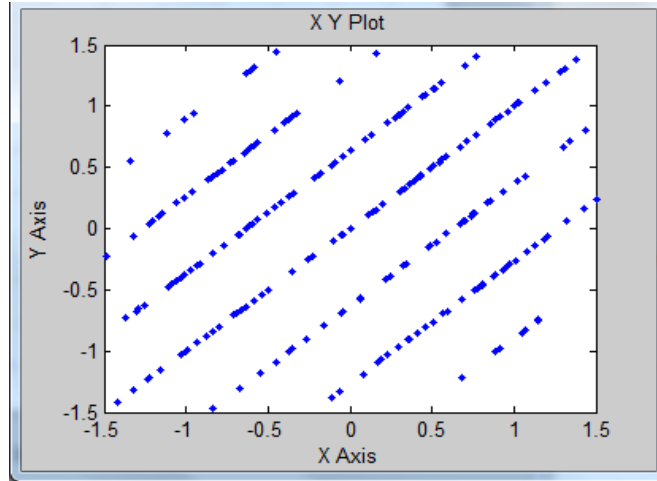
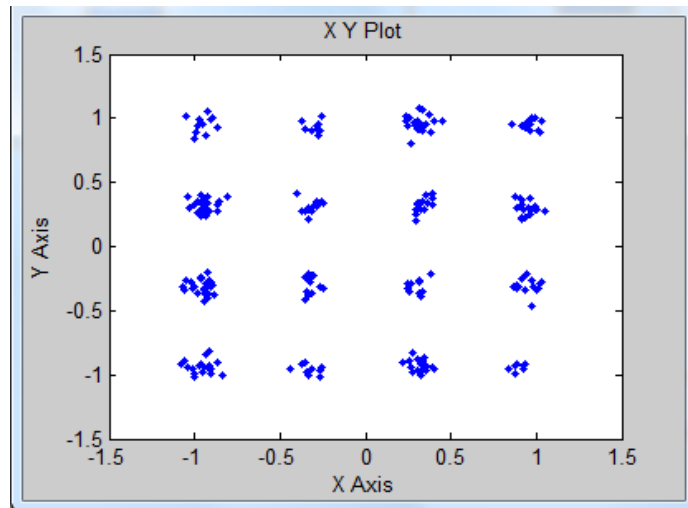


Figura 2.74. Diagrama de constelación señal modulada después del canal - semillas diferentes



Por lo anterior, en razón a que se debe garantizar la no correlación entre el ruido generado por los *WGNGs* de las ramas I y Q, las semillas de los *WGNGs* de cada rama (I y Q) se configuran con valores diferentes, garantizando patrones de ruido distintos tanto en la rama I como en la rama Q.

Ya que el *WGNG* posee un retardo de procesamiento de $T_b(10M)$ antes de comenzar a generar ruido, se hace necesario el uso de un *Delay* en cada rama (I y Q) que retarde las señales bipolares multinivel I y Q normalizadas, con el fin de que el ruido afecte toda la señal.

Con el propósito de controlar la cantidad de ruido suministrada por los *WGNGs* de las ramas I y Q, las señales de salida de estos elementos son multiplicadas por un factor que permite generar el valor de potencia de ruido normalizado correspondiente a un determinado valor de la relación energía de bit a densidad espectral de potencia de ruido (E_b/N_o). La multiplicación con el factor es realizada a través de un *Mult* y un *Constant* (con un valor igual al factor) en cada rama (I y Q).

Una vez obtenido el valor deseado de potencia de ruido normalizado, este ruido es adicionado a través de un *AddSub*, a las señales bipolares multinivel I y Q normalizadas.

El cálculo del factor que permite controlar la cantidad de ruido generada por los *WGNGs* de las ramas I y Q se realiza con base en los siguientes tres criterios:

Criterio 1. El ruido generado por un *WGNG* es modelado a través de una variable aleatoria normal estándar $X \sim N(\mu_x, \sigma_x^2)$, para cada instante, con valor medio (μ) igual a cero y varianza (σ^2) igual a uno; motivo por el cual el valor de potencia de ruido generado por este elemento corresponde al valor de la varianza (σ^2).

Criterio 2. Debido a que se utilizan dos *WGNGs* no correlacionados, en razón a la configuración de diferentes valores de semilla, uno en la rama I y otro en la rama Q, para representar las componentes en fase (I) y cuadratura (Q) del ruido AWGN complejo, el ruido generado por cada *WGNG* es modelado a través de una variable aleatoria diferente con distribución normal estándar, $X \sim N(\mu_x, \sigma_x^2)$ y $Y \sim N(\mu_y, \sigma_y^2)$, para cada instante, donde $X \sim N(\mu_x, \sigma_x^2)$ y $Y \sim N(\mu_y, \sigma_y^2)$ son no correlacionadas y por tanto independientes entre sí.

Ya que $X \sim N(\mu_x, \sigma_x^2)$ y $Y \sim N(\mu_y, \sigma_y^2)$ corresponden a variables aleatorias normales e independientes entre sí, la suma de las variables aleatorias $X \sim N(\mu_x, \sigma_x^2)$ y $Y \sim N(\mu_y, \sigma_y^2)$ genera una variable aleatoria $Z \sim N(\mu, \sigma^2)$ con distribución normal, tal como lo presenta la ecuación 2.1.

$$Z=X+Y \longrightarrow Z \sim N(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2) \quad (2.1)$$

Con base en lo anterior y dado que el valor de varianza (σ^2) de las variables aleatorias $X \sim N(\mu_x, \sigma_x^2)$ y $Y \sim N(\mu_y, \sigma_y^2)$ corresponde con el valor de la potencia de ruido generado por los *WGNGs* de las ramas I y Q; los *WGNGs* generan una potencia de ruido total como se presenta en la ecuación 2.2.

$$P_N = P_{WGNGI} + P_{WGNGQ} = \sigma_x^2 + \sigma_y^2, \quad (2.2)$$

donde,

P_N , potencia de ruido total.

P_{WGNGI} , potencia de ruido generada por el *WGNG* de la rama I.

P_{WGNGQ} , potencia de ruido generada por el *WGNG* de la rama Q.

σ_x^2 , varianza de la variable aleatoria que modela el ruido generado por el *WGNG* de la rama I, en un instante dado.

σ_y^2 , varianza de la variable aleatoria que modela el ruido generado por el *WGNG* de la rama Q, en un instante dado.

Criterio 3. Si una variable aleatoria normal estándar $U \sim N(0, 1)$ es multiplicada por una constante real arbitraria (a), dicha constante toma el lugar de la desviación estándar (σ), tal como lo presenta la ecuación 2.3.

$$V = aU \longrightarrow V \sim N(0, a^2) \quad (2.3)$$

Teniendo en cuenta la anterior propiedad y considerando que el ruido generado por un *WGNG* es modelado a través de una variable aleatoria normal estándar, para cada instante; si se multiplica un factor constante y real por la señal de salida de un *WGNG*, este factor toma el lugar de la desviación estándar (σ).

De acuerdo a lo anterior, se establece que el factor que permite generar el valor de potencia de ruido normalizado correspondiente a un determinado valor de la relación energía de bit a densidad espectral de potencia de ruido (E_b/N_o) corresponde a la desviación estándar (σ), por lo tanto los cálculos que se presentan a continuación relacionan los parámetros σ y E_b/N_o . Estos cálculos se plantean de forma general, tanto para la modulación 16 QAM como para la modulación 64 QAM.

Partiendo de la definición de la relación energía de bit a densidad espectral de potencia de ruido (E_b/N_o) presentada en la ecuación 2.4 y considerando el ancho de banda de la señal y la velocidad de transmisión de datos presentados en las ecuaciones 2.5 y 2.6 respectivamente, se obtiene la ecuación 2.7.

$$\left(\frac{E_b}{N_o}\right) = \frac{P_{señal}}{R_b} * \frac{Bw_{señal}}{P_{ruido}} \quad (2.4)$$

donde,

E_b , energía de bit [J/bit].

N_o , PSD de ruido [W/Hz].

$P_{señal}$, potencia de la señal [W].

R_b , velocidad de transmisión de datos [bps].

$BW_{señal}$, ancho de banda de la señal [Hz].

P_{ruido} , potencia de ruido [W].

$$Bw_{señal} = \frac{1}{T_s}, \quad (2.5)$$

donde,

T_s , periodo de símbolo [s].

$$R_b = \frac{1}{T_b}, \quad (2.6)$$

donde,

T_b , periodo de bit [s].

$$\left(\frac{E_b}{N_o}\right) = \frac{P_{señal}}{P_{ruido}} * \frac{T_b}{T_s} \quad (2.7)$$

Ya que el periodo de símbolo puede definirse como lo muestra la ecuación 1.4, esta variable es remplazada en la ecuación 2.7, originando la ecuación 2.8.

$$\left(\frac{E_b}{N_o}\right) = \frac{Pseñal}{Pruido} * \left(\frac{T_b}{T_b * \log_2(m)}\right) = \frac{Pseñal}{Pruido} * \left(\frac{1}{\log_2(m)}\right) \quad (2.8)$$

Realizando la conversión a dB de la ecuación 2.8, se obtiene la ecuación 2.9, de donde se despeja la relación Pseñal/Pruido, como lo presenta la ecuación 2.10.

$$\left(\frac{E_b}{N_o}\right) [dB] = \frac{Pseñal}{Pruido} [dB] + 10\log\left(\frac{1}{\log_2(m)}\right) [dB] \quad (2.9)$$

$$\frac{Pseñal}{Pruido} [dB] = \left(\frac{E_b}{N_o}\right) [dB] - 10\log\left(\frac{1}{\log_2(m)}\right) [dB] \quad (2.10)$$

Teniendo en cuenta que el nivel de potencia de la señal es normalizado, y considerando que la potencia de ruido generada por los dos WGNGs corresponde a la suma de las potencias individuales, dada su no correlación, como se estableció en el criterio 2, la relación Pseñal/Pruido es planteada en la ecuación 2.11.

$$\frac{Pseñal}{Pruido} = \frac{1}{\sigma_x^2 + \sigma_y^2} \quad (2.11)$$

donde,

σ_x^2 , varianza de la variable aleatoria X que modela el ruido generado por el WGNG de la rama I, en un instante dado.

σ_y^2 , varianza de la variable aleatoria Y que modela el ruido generado por el WGNG de la rama Q, en un instante dado.

Considerando valores de desviación estándar (σ) iguales en la ecuación 2.11, se logra la ecuación 2.12, donde se despeja la variable desviación estándar (σ), obteniendo como resultado la ecuación 2.13.

$$\frac{Pseñal}{Pruido} = \frac{1}{2\sigma^2} \quad (2.12)$$

$$\sigma = \frac{1}{\sqrt{2 * Psenal / Pruido}} \quad (2.13)$$

Finalmente, considerando la definición de la Relación Señal a Ruido (SNR, *Signal to Noise Ratio*) planteada en la ecuación 2.14, y remplazando esta variable en las ecuaciones 2.10 y 2.13, se logran obtener las ecuaciones 2.15 y 2.16, que relacionan el valor de la desviación estándar (σ) con un determinado valor de la relación energía de bit

a densidad espectral de potencia de ruido (E_b/N_o), para los esquemas de modulación 16 QAM y 64 QAM.

$$\frac{P_{señal}}{P_{ruido}} = SNR, \quad (2.14)$$

donde,
SNR, relación señal a ruido.

$$SNR [dB] = \left(\frac{Eb}{No}\right) [dB] - 10\log\left(\frac{1}{\log_2(m)}\right) [dB] \quad (2.15)$$

$$\sigma = \frac{1}{\sqrt{2 * SNR}} \quad (2.16)$$

Con el fin de ilustrar la manera como se realiza el cálculo del factor que permite generar el valor de potencia de ruido normalizado correspondiente a un valor de E_b/N_o , se considera un esquema de modulación 16 QAM y un valor de E_b/N_o igual a 2 dB.

Remplazando en la ecuación 2.15 el valor E_b/N_o igual a 2 dB y el número de estados m igual a 16, se obtiene el valor SNR de la ecuación 2.17.

$$SNR [dB] = 2[dB] - 10\log\left(\frac{1}{\log_2(16)}\right) [dB] = 2dB - 10\log 0.25 dB = 8.02 dB \quad (2.17)$$

Realizando la conversión a veces del valor SNR obtenido en la ecuación 2.17 y remplazando este valor en la ecuación 2.16, se halla el valor de la desviación estándar (σ) correspondiente a un valor de E_b/N_o igual a 2 dB, como lo presenta la ecuación 2.18.

$$\sigma = \frac{1}{\sqrt{2 * 6.34}} = 0.28 \quad (2.18)$$

Como se mencionó anteriormente, el valor de la desviación estándar (σ) obtenido corresponde al valor del factor que es multiplicado por las señales de salida de los *WGNs*, el cual es configurado en un *Constant*, tanto en la rama I como en la rama Q (ver figura 2.72).

En el apéndice A se encuentran los valores de desviación estándar (σ) correspondientes a determinados valores de la relación energía de bit a densidad espectral de potencia de ruido (E_b/N_o), calculados a partir de las ecuaciones 2.15 y 2.16, para los esquemas de modulación 16 QAM y 64 QAM.

Las figuras 2.75 a 2.78 muestran las señales de entrada y salida (señales bipolares multinivel normalizadas más AWGN) del canal AWGN complejo, correspondientes a una E_b/N_o igual a 2 dB, tanto para la modulación 16 QAM como para la modulación 64 QAM.

Figura 2.75. Señales de entrada y salida del canal AWGN de la rama I, modulación 16 QAM

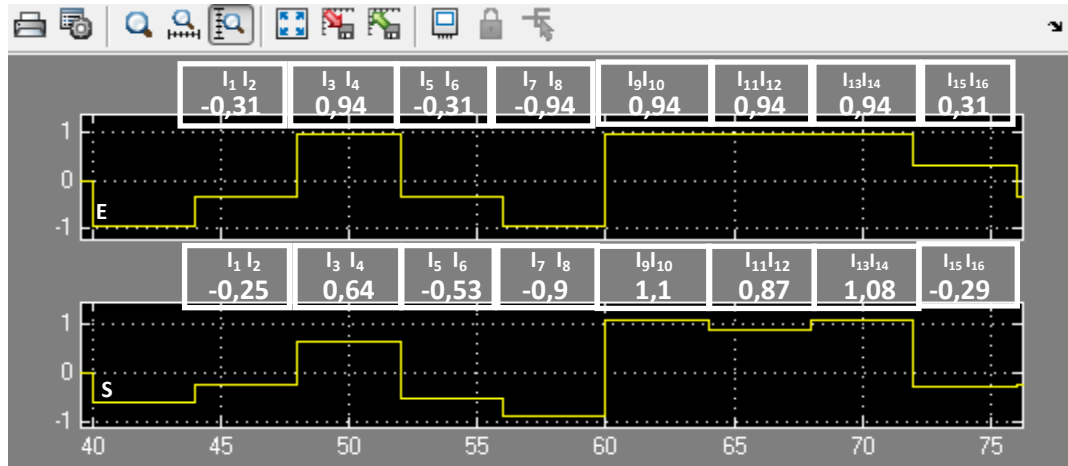


Figura 2.76. Señales de entrada y salida del canal AWGN de la rama Q, modulación 16 QAM

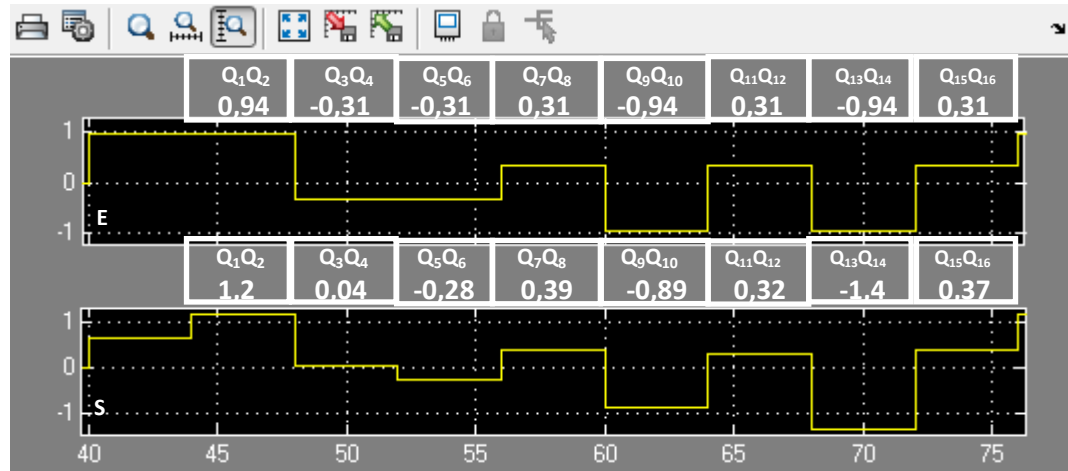


Figura 2.77. Señales de entrada y salida del canal AWGN de la rama I, modulación 64 QAM

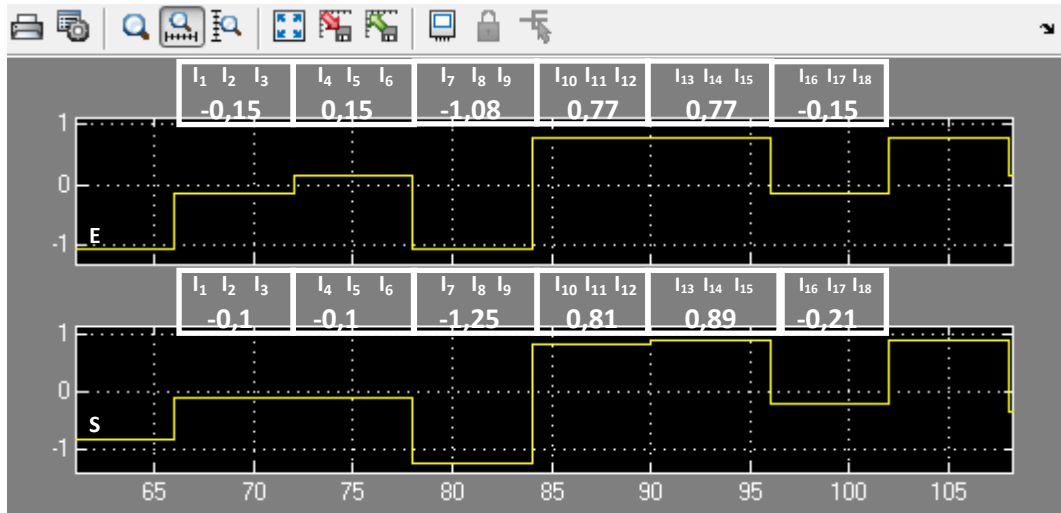
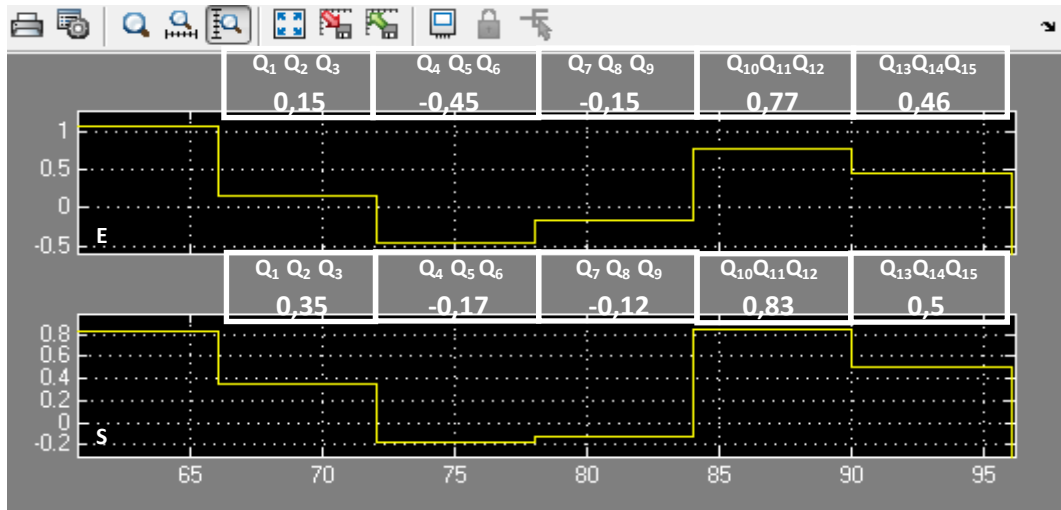


Figura 2.78. Señales de entrada y salida del canal AWGN de la rama Q, modulación 64 QAM



Bloque 6. Desnormalizador de potencia. La construcción de este bloque requiere en cada rama (I y Q) el uso de un *Constant* con un valor igual al inverso del factor de normalización aplicado en el normalizador de potencia y un *Mult* que realiza el producto entre este valor y la señal proporcionada por el canal AWGN complejo (ver figura 2.79). Las señales obtenidas por este bloque corresponden a las señales bipolares multinivel I y Q afectadas por ruido AWGN complejo, pero con potencia desnormalizada (ver figuras 2.80 a 2.83).

Figura 2.79. Desnormalizador de potencia

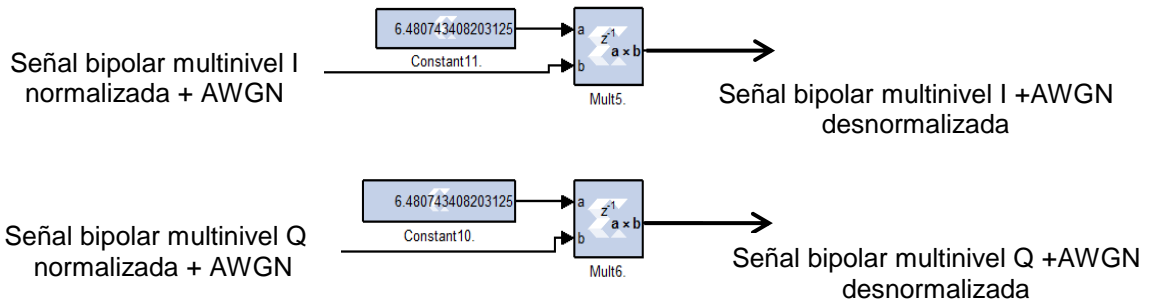


Figura 2.80. Señales de entrada y salida del desnormalizador de potencia de la rama I, modulación 16 QAM

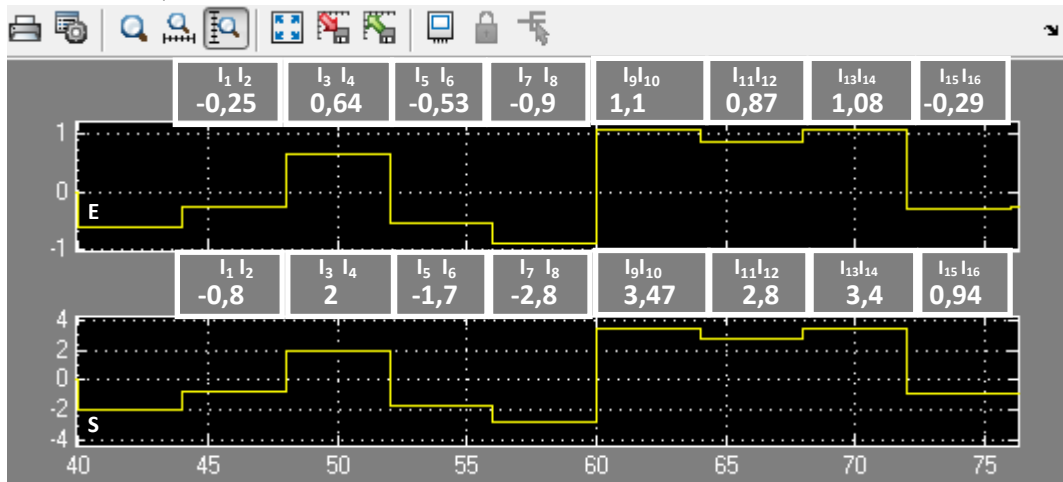


Figura 2.81. Señales de entrada y salida del desnormalizador de potencia de la rama Q, modulación 16 QAM

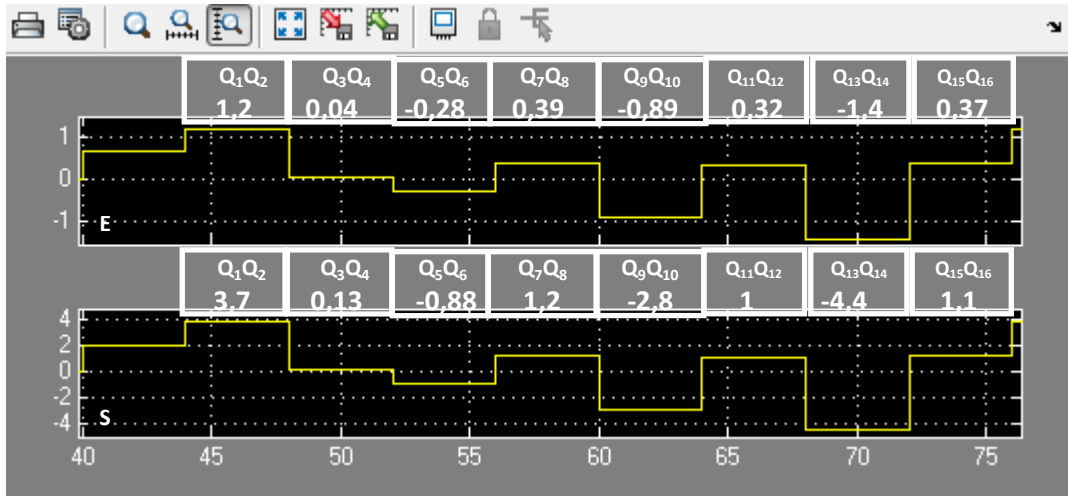


Figura 2.82. Señales de entrada y salida del desnormalizador de potencia de la rama I, modulación 64 QAM

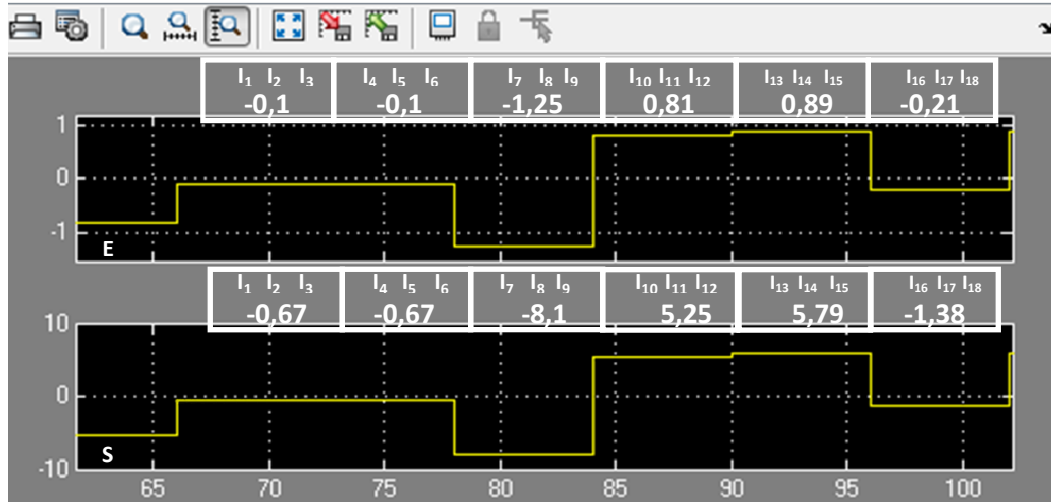
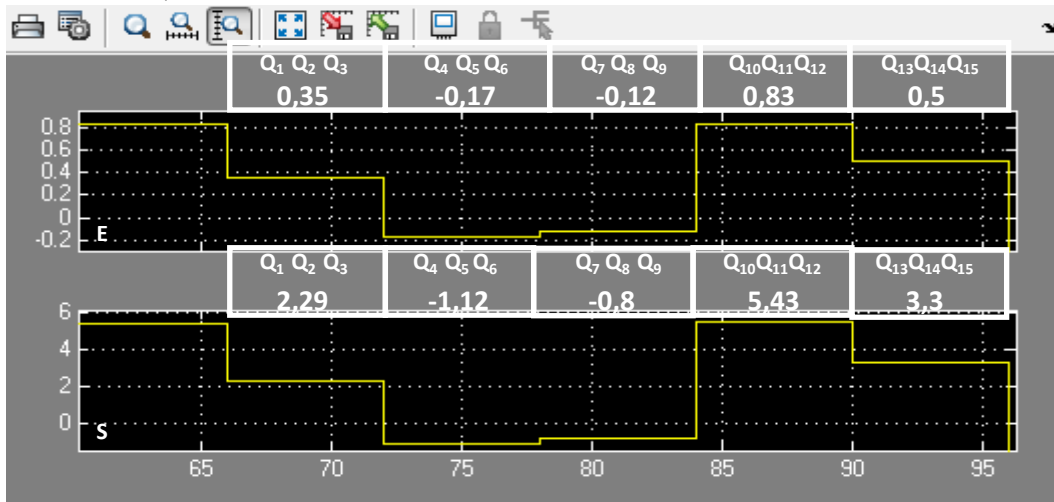


Figura 2.83. Señales de entrada y salida del desnormalizador de potencia de la rama Q, modulación 64 QAM



Bloque 7. Decisor. Conformado únicamente por un *Mcode* (ver figura 2.84) que establece rangos de decisión o valores umbrales de acuerdo al conjunto de todos los posibles valores que puede tomar un símbolo complejo (ver tablas 2.5 y 2.6) y con base en el criterio de decisión de distancia mínima²¹, decide a qué valores pertenecen los niveles de las señales bipolares multinivel I y Q proporcionadas por el desnormalizador de potencia (ver figuras 2.85 a 2.88).

²¹ De acuerdo a lo establecido en la sección 1.1.2.2, en el caso de símbolos equiprobables e independientes, el criterio de decisión óptimo para minimizar la probabilidad de error es el criterio de decisión de distancia mínima, motivo por el cual este criterio fue seleccionado para el decisor.

Para realizar la tarea del decisor, dentro del *Mcode* se desarrolla un código en lenguaje de programación Matlab®, basado en estructuras condicionales, que permite asignar el valor correspondiente del criterio de decisión.

Figura 2.84. Decisor

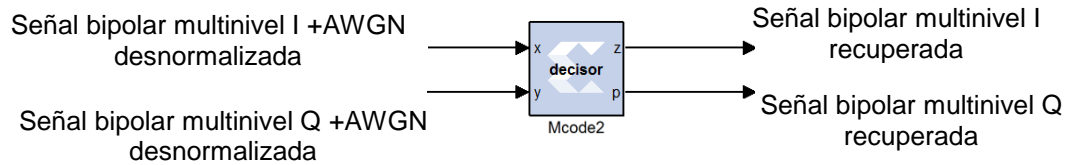


Tabla 2.5. Umbrales de decisión Mapeo Binario y Gray, modulación 16 QAM

Mapeo Binario		Mapeo de Gray	
Entrada I/Q	Decisión I/Q	Entrada I/Q	Decisión I/Q
$x \geq 2,5$	3	$x < -2$	-3
$1,5 \leq x < 2,5$	2	$-2 \leq x < 0$	-1
$0,5 \leq x < 1,5$	1	$0 \leq x < 2$	1
$x < 0,5$	0	$x \geq 2$	3

Tabla 2.6. Umbrales de decisión Mapeo Binario y Gray, modulación 64 QAM

Mapeo Binario		Mapeo Gray	
Entrada I/Q	Decisión I/Q	Entrada I/Q	Decisión I/Q
$x \geq 6,5$	7	$x < -6$	-7
$5,5 \leq x < 6,5$	6	$-6 \leq x < -4$	-5
$4,5 \leq x < 5,5$	5	$-4 \leq x < -2$	-3
$3,5 \leq x < 4,5$	4	$-2 \leq x < 0$	-1
$2,5 \leq x < 3,5$	3	$0 \leq x < 2$	1
$1,5 \leq x < 2,5$	2	$2 \leq x < 4$	3
$0,5 \leq x < 1,5$	1	$4 \leq x < 6$	5
$x < 0,5$	0	$x \geq 6$	7

Figura 2.85. Señales de entrada y salida del decisor de la rama I, modulación 16 QAM

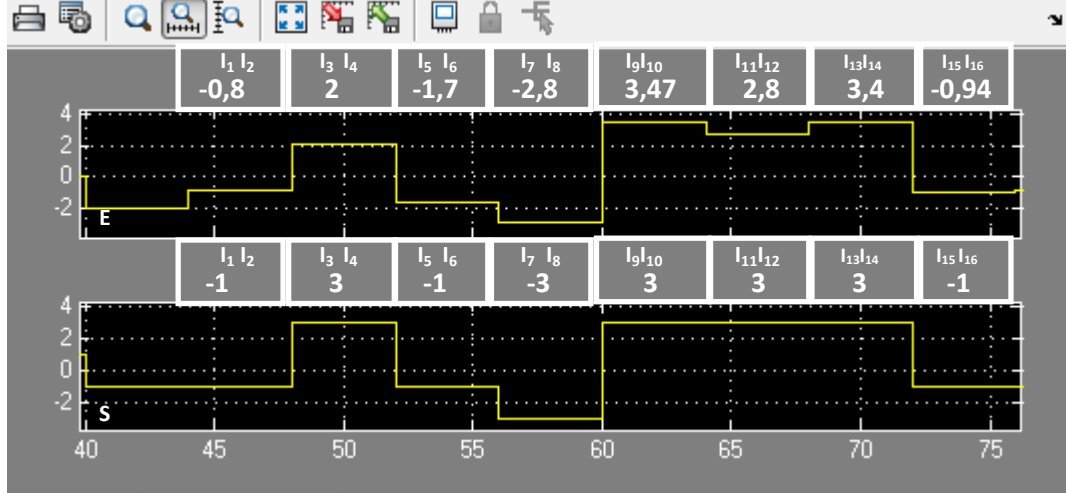


Figura 2.86. Señales de entrada y salida del decisor de la rama Q, modulación 16 QAM

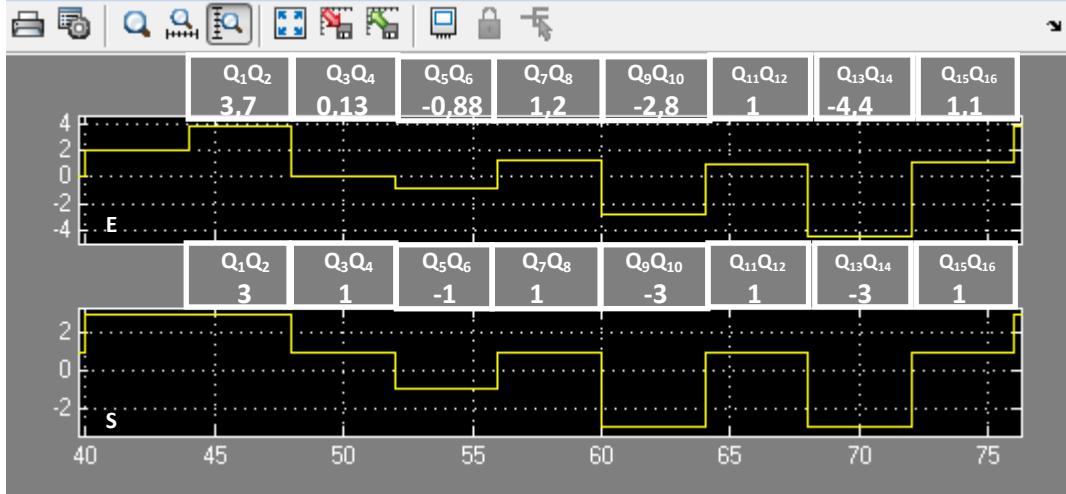


Figura 2.87. Señales de entrada y salida del decisor de la rama I, modulación 64 QAM

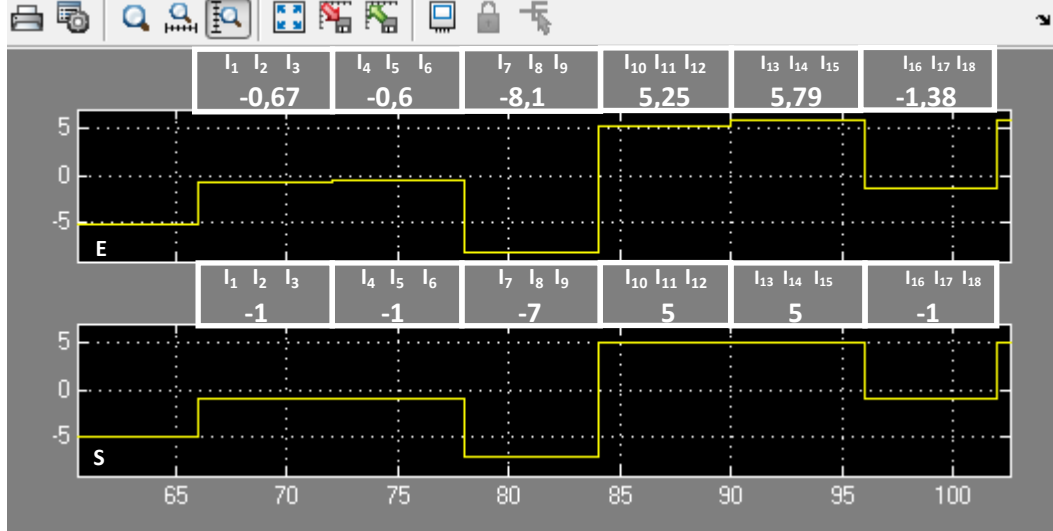
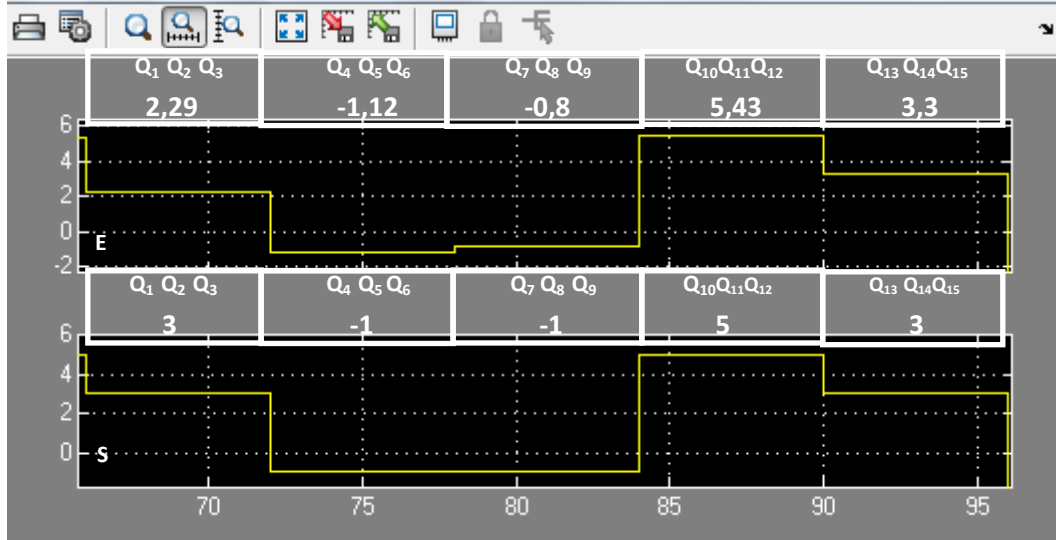
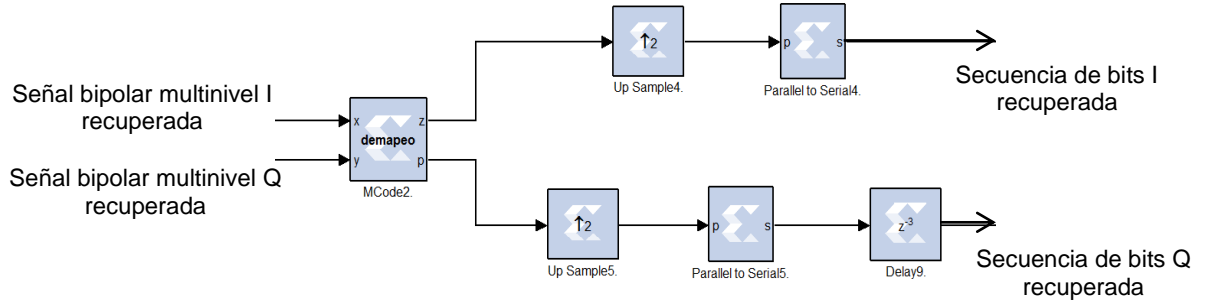


Figura 2.88. Señales de entrada y salida del decisor de la rama Q, modulación 64 QAM



Bloque 8. Conversor de M a 2 niveles. Este bloque consta de un *Mcode* que demapea las señales bipolares multinivel I y Q; un *Up Sample* y un *Parallel to Serial* en cada rama (I y Q) que introduce los ceros eliminados por el *Down Sample* del modulador y representa las señales como secuencias de bits respectivamente y un *Delay* que desincroniza las señales de las ramas I y Q (ver figura 2.89).

Figura 2.89. Conversor de M a 2 niveles



El *Mcode* del conversor de M a 2 niveles recibe las señales bipolares multinivel (I y Q) recuperadas y las decodifica en las correspondientes señales unipolares multinivel (con periodo de muestreo igual a $T_b M$) que representan los símbolos complejos recuperados (ver figuras 2.90 a 2.93), de acuerdo al mapeo utilizado en el modulador (ver tablas 2.7 y 2.8). Para realizar esta tarea, dentro del *Mcode* se desarrolla un código en lenguaje de programación Matlab®, basado en estructuras condicionales, que permite realizar las respectivas acciones. Dado que tanto el decisor como el conversor de M a 2 niveles requieren un *Mcode*, se utiliza un único *Mcode* con los códigos de ambos bloques.

Tabla 2.7. Demapeo Binario y Gray, modulación 16 QAM

Entrada			
Mapeo Binario	Mapeo Gray		Salida
I/Q	I	Q	I/Q
3	1	-1	3
1	3	-3	2
-1	-1	1	1
-3	-3	3	0

Tabla 2.8. Demapeo Binario y Gray, modulación 64 QAM

Entrada			
Mapeo Binario	Mapeo Gray		Salida
I/Q	I	Q	I/Q
7	3	-3	7
5	1	-1	6
3	5	-5	5
1	7	-7	4
-1	-3	3	3
-3	-1	1	2
-5	-5	5	1
-7	-7	7	0

Figura 2.90. Señales de entrada y salida del *Mcode* de la rama I, modulación 16 QAM

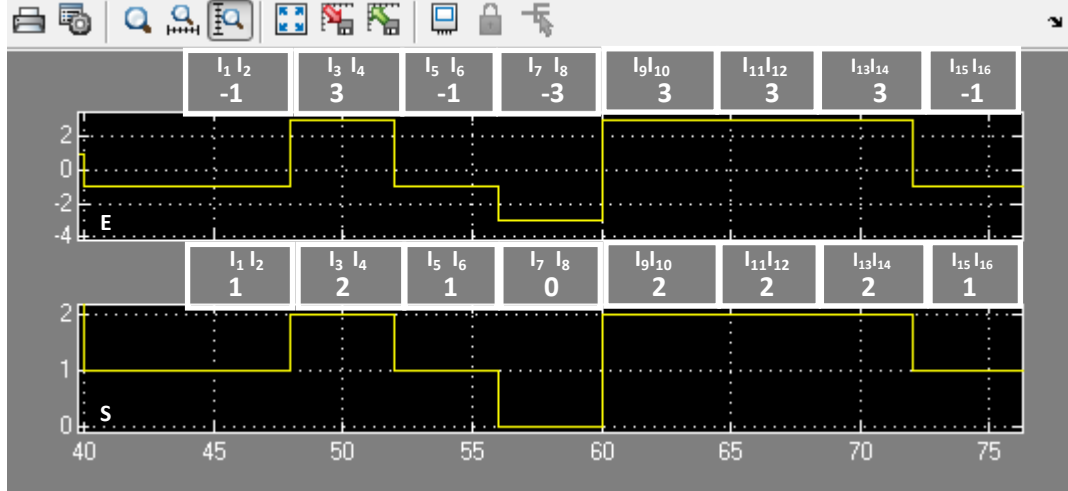


Figura 2.91. Señales de entrada y salida del *Mcode* de la rama Q, modulación 16 QAM

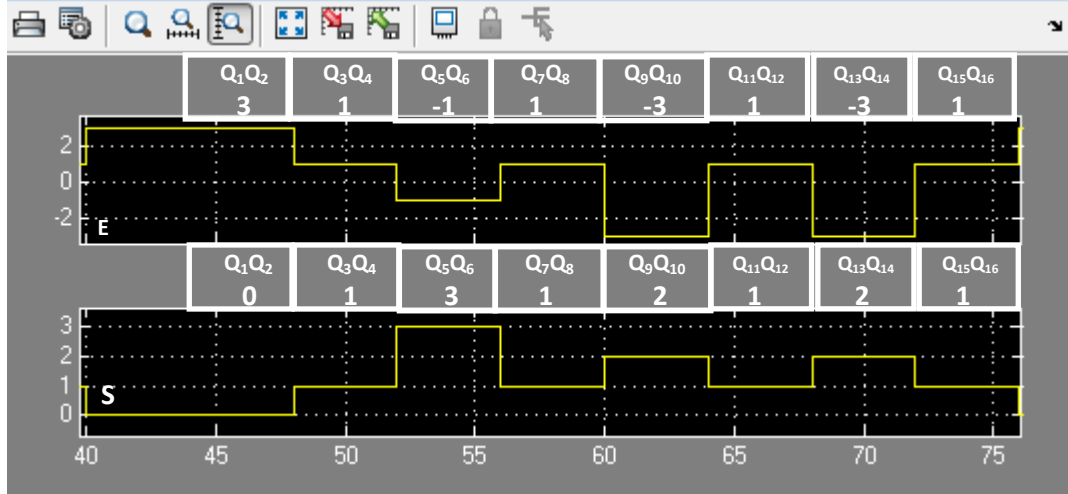


Figura 2.92. Señales de entrada y salida del *Mcode* de la rama I, modulación 64 QAM

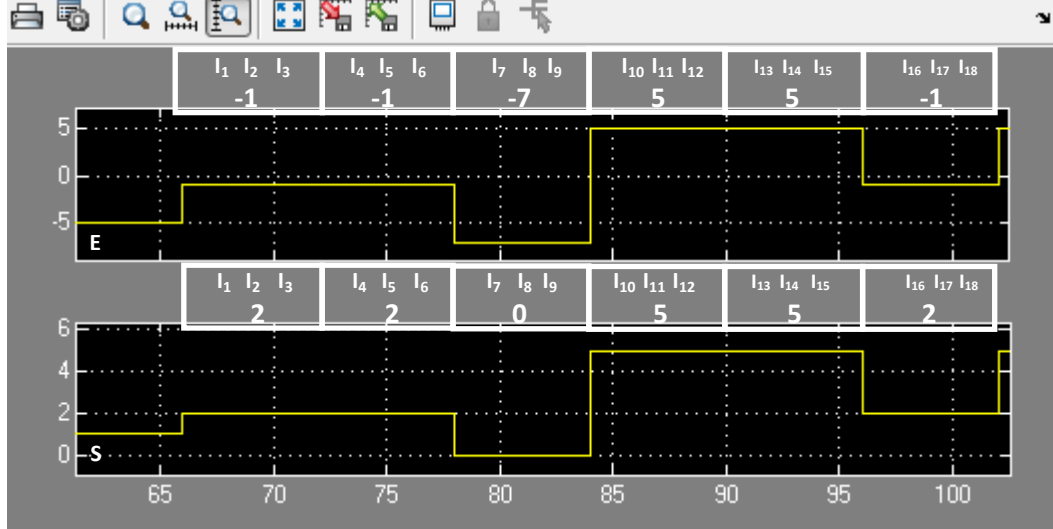
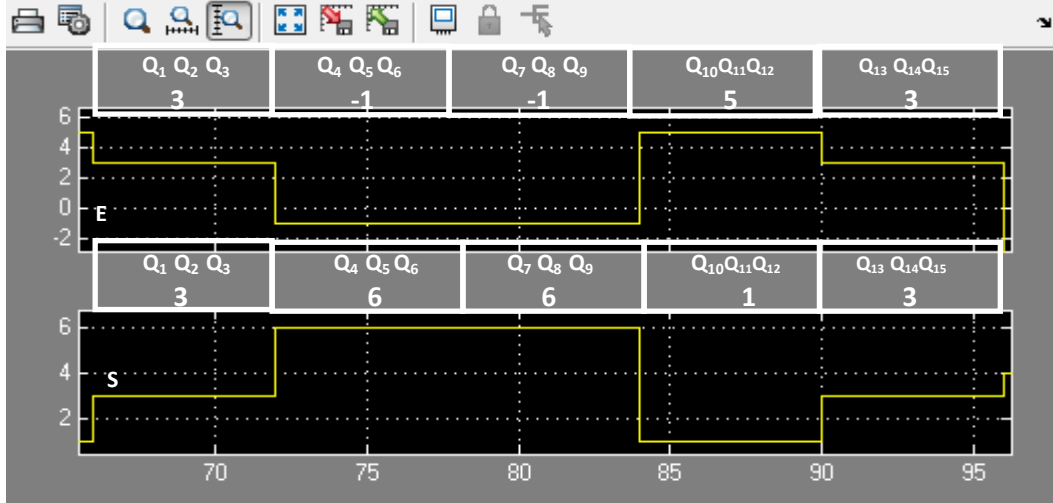


Figura 2.93. Señales de entrada y salida del *Mcode* de la rama Q, modulación 64 QAM



En cuanto se obtienen las señales recuperadas unipolares multinivel I y Q del *Mcode*, cada una de estas señales es procesada por un *Up Sample* (ver figura 2.89) que aumenta la frecuencia de muestreo al doble²² (obteniendo dos muestras por cada valor) y reemplaza la segunda muestra por un valor igual a cero. Así, las muestras obtenidas por el *Up Sample* generan una señal unipolar multinivel que toma valores iguales a cero de manera periódica (ver figuras 2.94 a 2.97).

²² Dado que el periodo de muestreo de las señales unipolares multinivel I y Q proporcionadas por el *Mcode* corresponde a $T_b M$, el *Up Sample* de cada rama (I y Q) muestrea la señal correspondiente con un periodo de muestreo igual a $T_b(M/2)$.

Figura 2.94. Señales de entrada y salida del *Up Sample* de la rama I, modulación 16 QAM

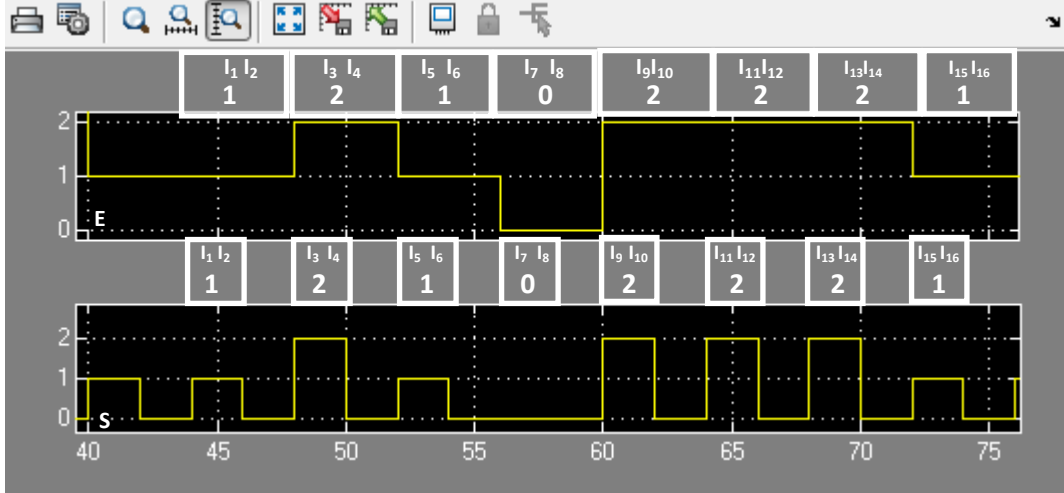


Figura 2.95. Señales de entrada y salida del *Up Sample* de la rama Q, modulación 16 QAM

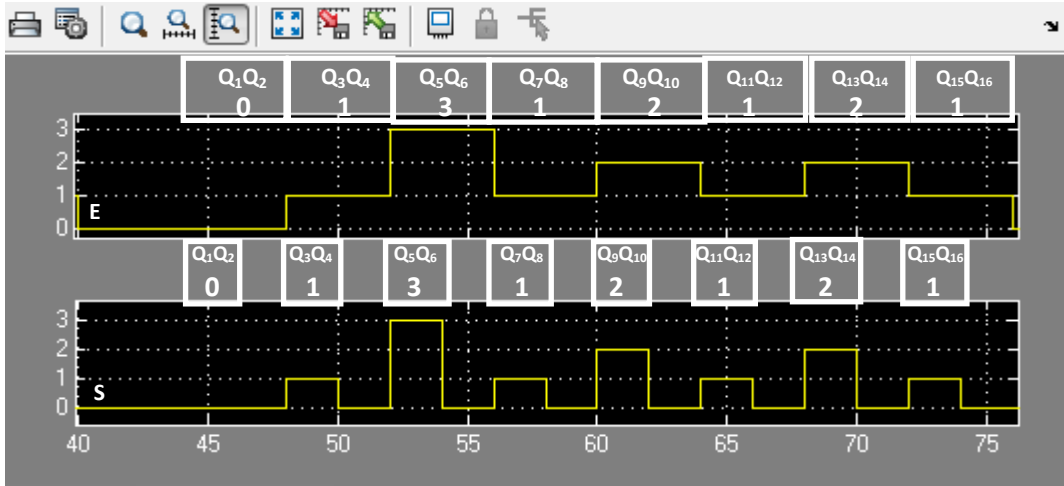


Figura 2.96. Señales de entrada y salida del *Up Sample* de la rama I, modulación 64 QAM

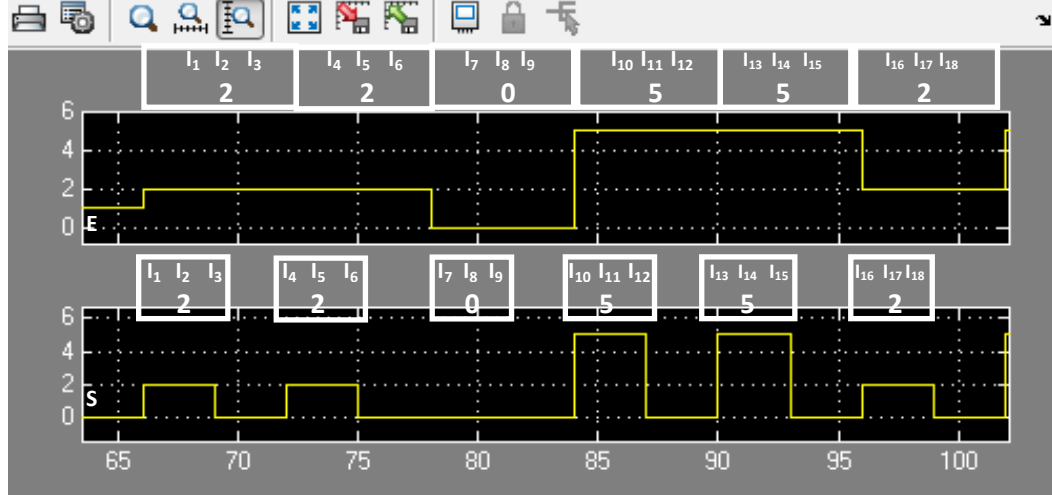
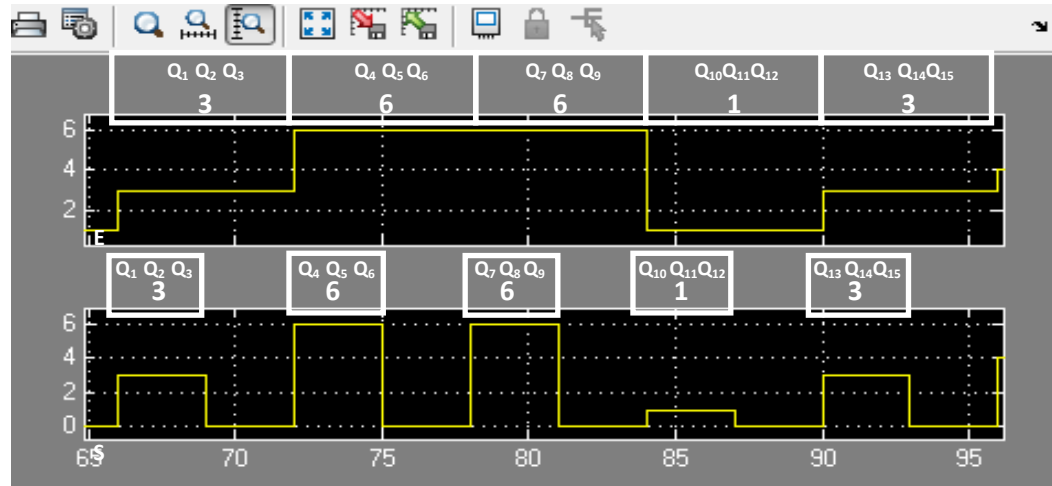


Figura 2.97. Señales de entrada y salida del *Up Sample* de la rama Q, modulación 64 QAM



Tan pronto como se obtienen las señales unipolares multinivel I y Q recuperadas de los correspondientes *Up Sample*, un *Parallel to Serial* en la rama I y otro en la rama Q (ver figura 2.89) convierten cada valor decimal de las señales unipolares multinivel en un número binario de $M/2$ bits, donde el bit del extremo izquierdo corresponde al MSB, obteniendo como resultado las secuencias de bits I y Q recuperadas, cuyos periodos de bit son iguales a T_b (ver figuras 2.98 a 2.101).

Figura 2.98. Señales de entrada y salida del *Parallel to Serial* de la rama I, modulación 16 QAM

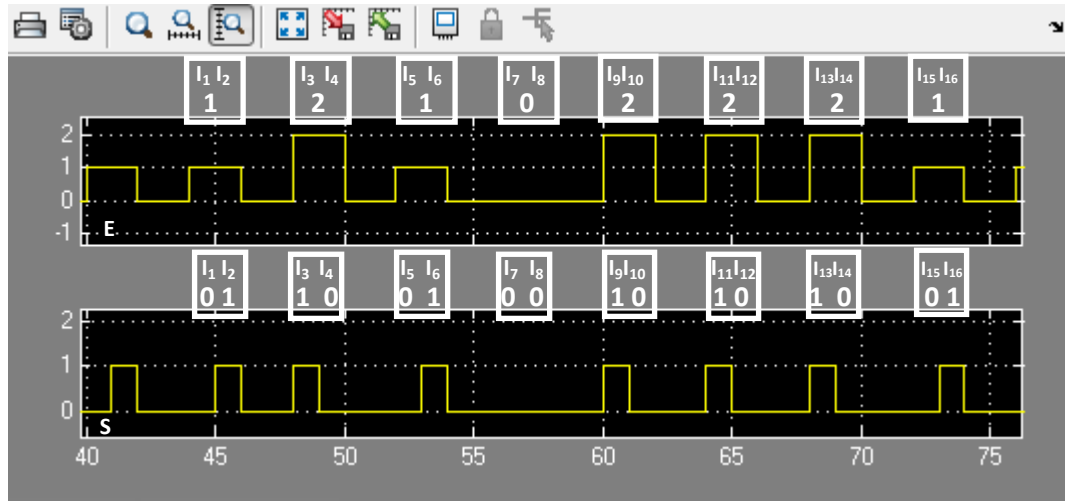


Figura 2.99. Señales de entrada y salida del *Parallel to Serial* de la rama Q, modulación 16 QAM

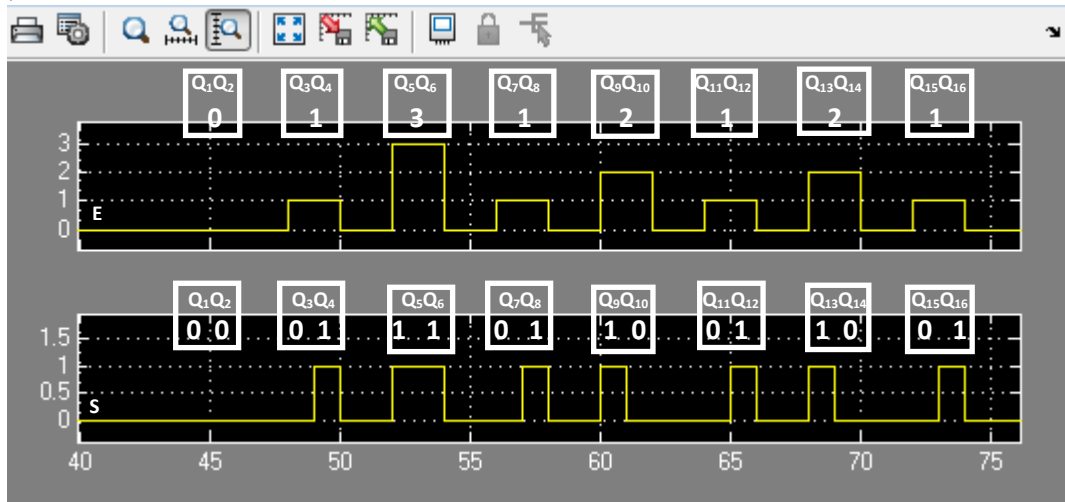


Figura 2.100. Señales de entrada y salida del *Parallel to Serial* de la rama I, modulación 64 QAM

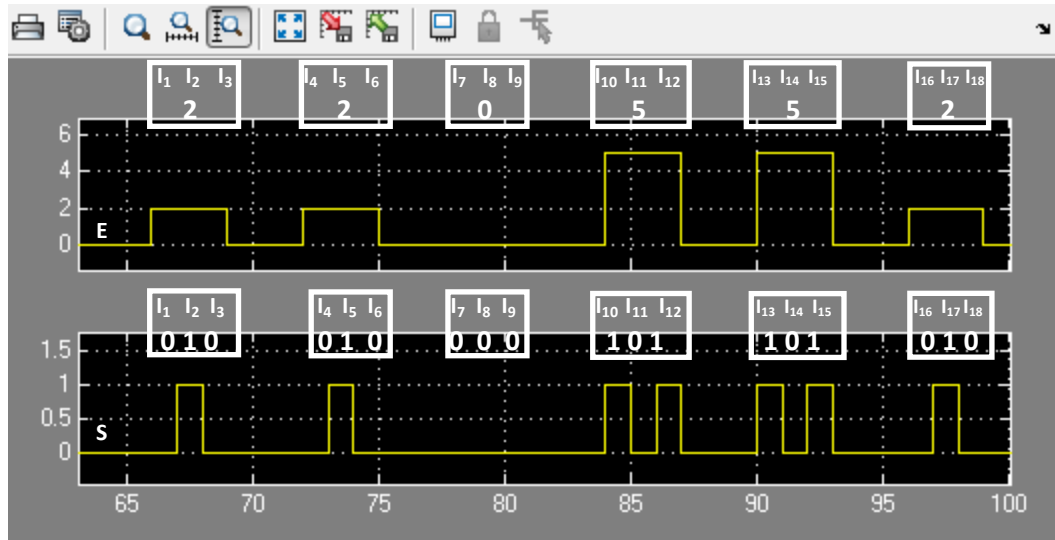
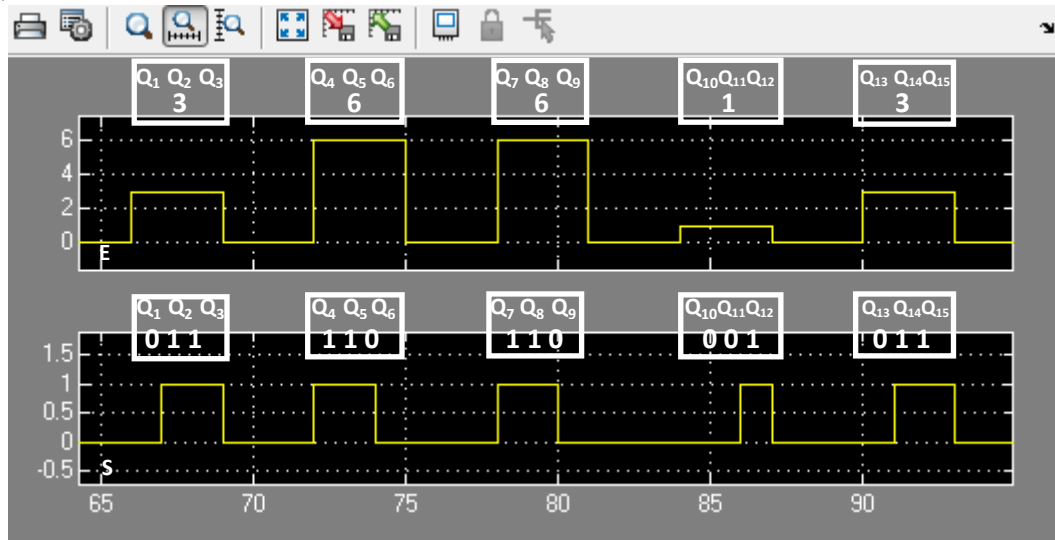


Figura 2.101. Señales de entrada y salida del *Parallel to Serial* de la rama Q, modulación 64 QAM



Finalmente un *Delay* con valor igual a $T_b(M/2)$ en la rama Q (ver figura 2.89) desincroniza la secuencia de bits Q recuperada respecto a la secuencia de bits I recuperada (ver figuras 2.102 y 2.103).

Figura 2.102. Señales de las ramas I y Q desincronizadas, modulación 16 QAM

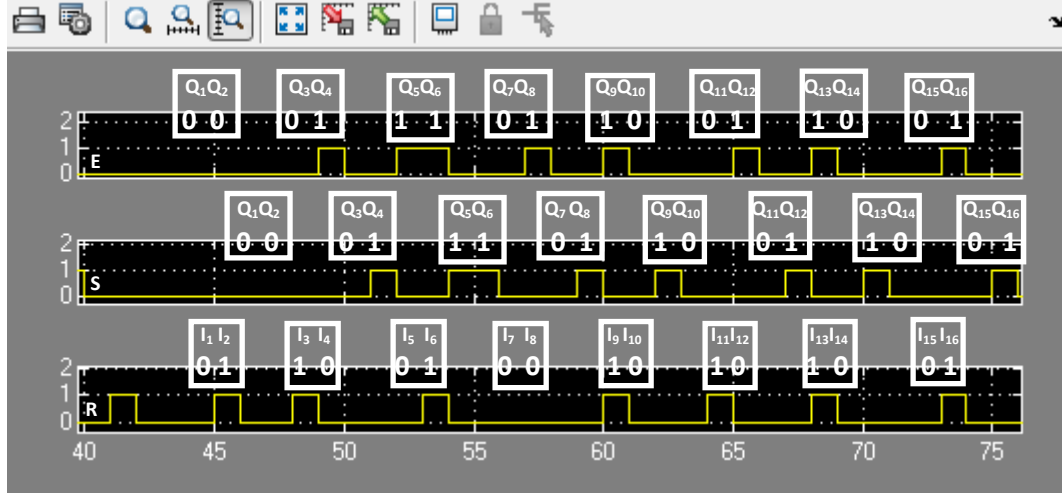
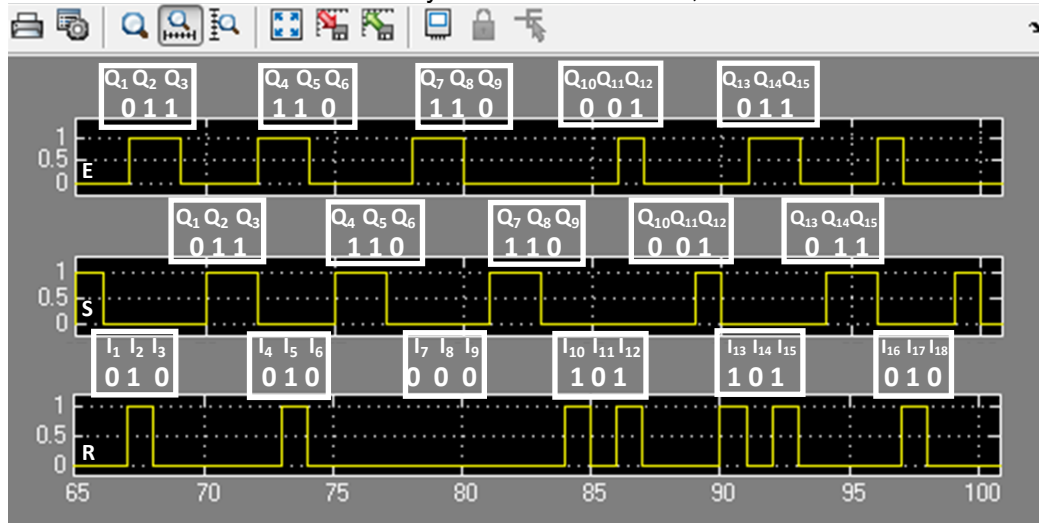
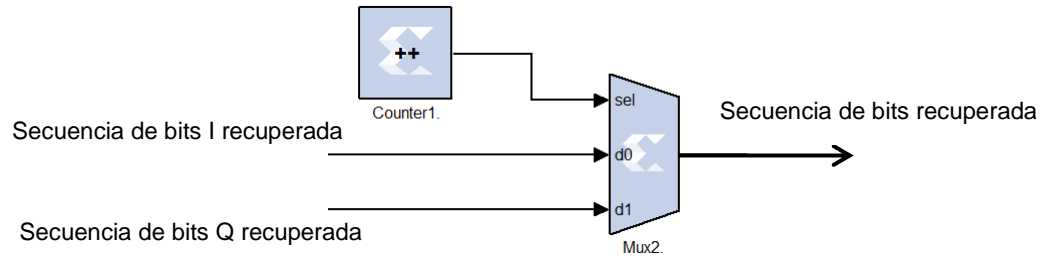


Figura 2.103. Señales de las ramas I y Q desincronizadas, modulación 64 QAM



Bloque 9. Multiplexor de datos. Hace uso de un *Mux* controlado por un *Counter* para combinar las secuencias de bits I y Q recuperadas en una sola secuencia (ver figura 2.104). La salida obtenida por el Multiplexor de datos corresponde a la secuencia de bits recuperada o también denominada señal demodulada (ver figuras 2.105 y 2.106).

Figura 2.104. Multiplexor de datos



Las figuras 2.107 y 2.108 muestran la señal transmitida por la fuente de información y la secuencia de bits recuperada, donde se aprecia un retardo total igual a $T_b(11M)$ resultado del procesamiento de la señal transmitida en cada uno de los bloques del sistema básico de comunicaciones.

Dado que los primeros bits de la secuencia recuperada corresponden a bits de la rama I, estos bits indican el momento en que la secuencia inicia y por lo tanto el retardo de dicha secuencia; la ecuación 2.19 presenta cada uno de los retardos de la rama I y la suma de ellos.

Retardo del *Delay* del convertor de 2 a M niveles = $T_b(M/2)$

Retardo del *Serial to Parallel* del convertor de 2 a M niveles = $T_b(M/2)$

Retardo de los *WNGGs* del canal AWGN complejo = $T_b(10M)$

$$\text{Retardo total} = T_b\left(\frac{M}{2}\right) + T_b\left(\frac{M}{2}\right) + T_b(10M) = T_b(11M) \quad (2.19)$$

Figura 2.105. Secuencia de bits recuperada, modulación 16 QAM

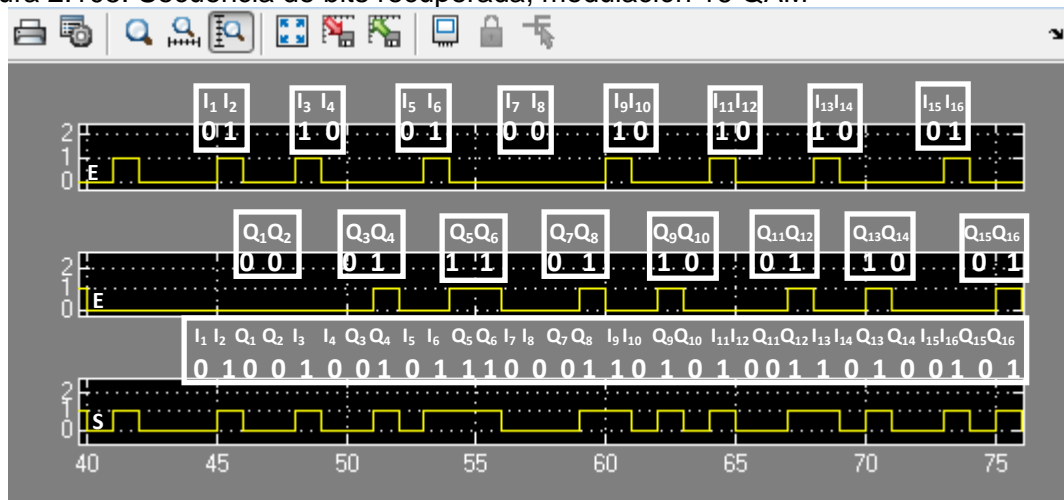


Figura 2.106. Secuencia de bits recuperada, modulación 64 QAM

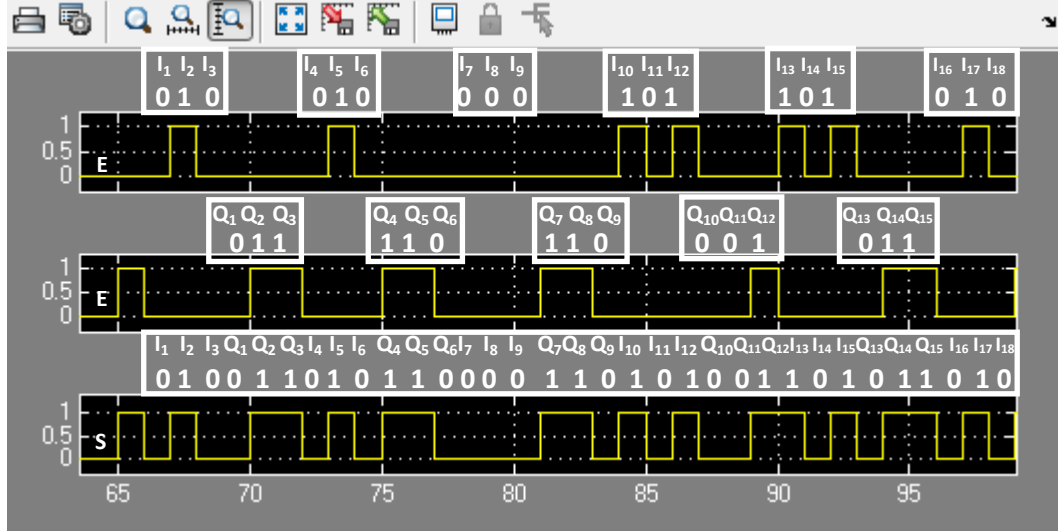


Figura 2.107. Secuencia de bits predefinida y secuencia de bits recuperada, modulación 16 QAM

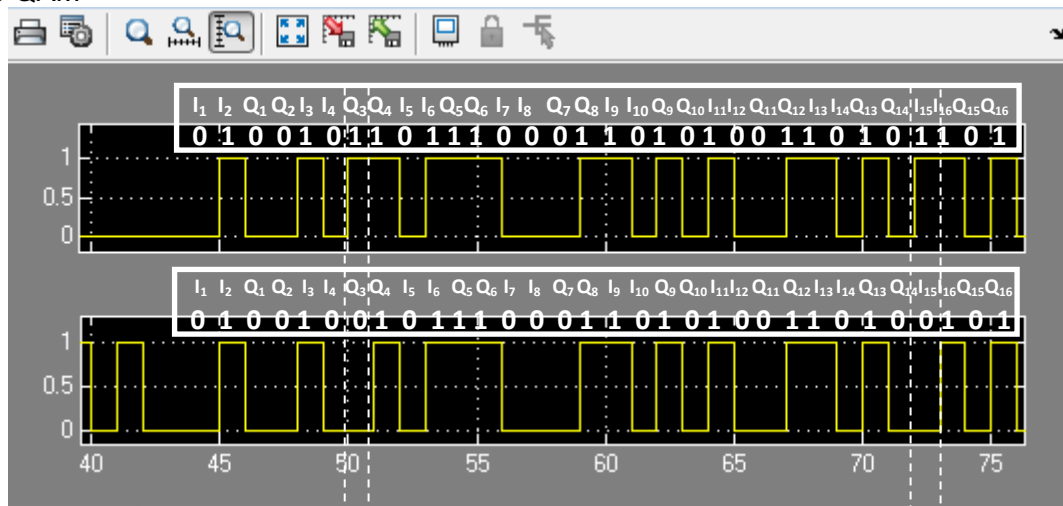
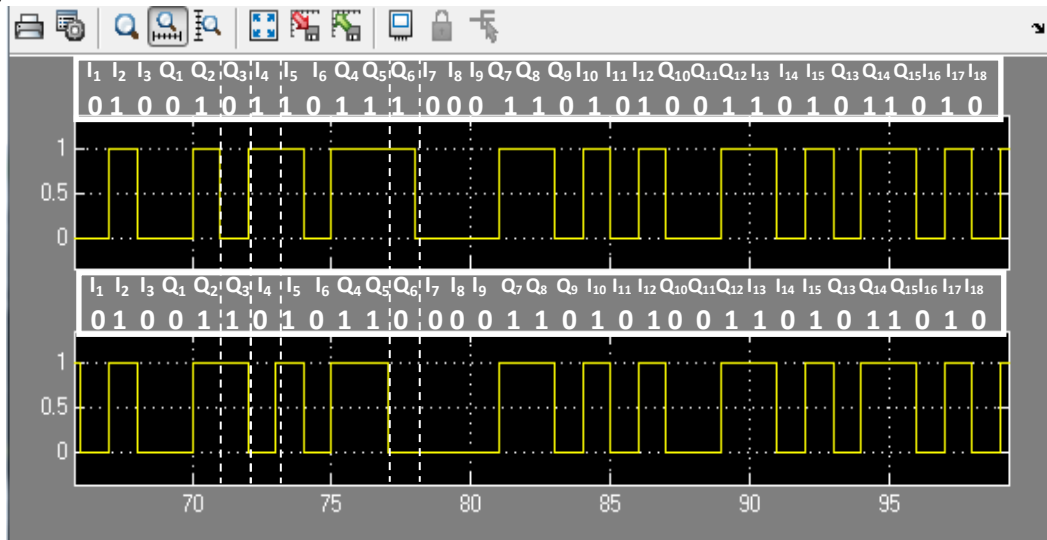


Figura 2.108. Secuencia de bits predefinida y secuencia de bits recuperada, modulación 64 QAM



Finalmente, las figuras 2.109 y 2.110 muestran la señal transmitida por la fuente de información y la secuencia de bits recuperada, tanto para la modulación 16 QAM como para la modulación 64 QAM, para un ruido de canal igual a cero, donde ambas señales son idénticas.

Figura 2.109. Secuencia de bits predefinida y secuencia de bits recuperada, ruido igual a cero, modulación 16 QAM

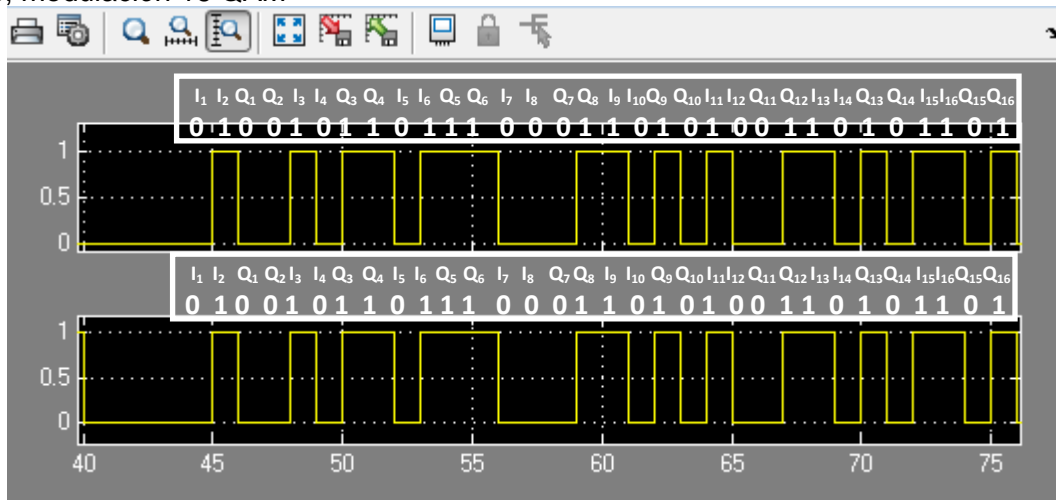
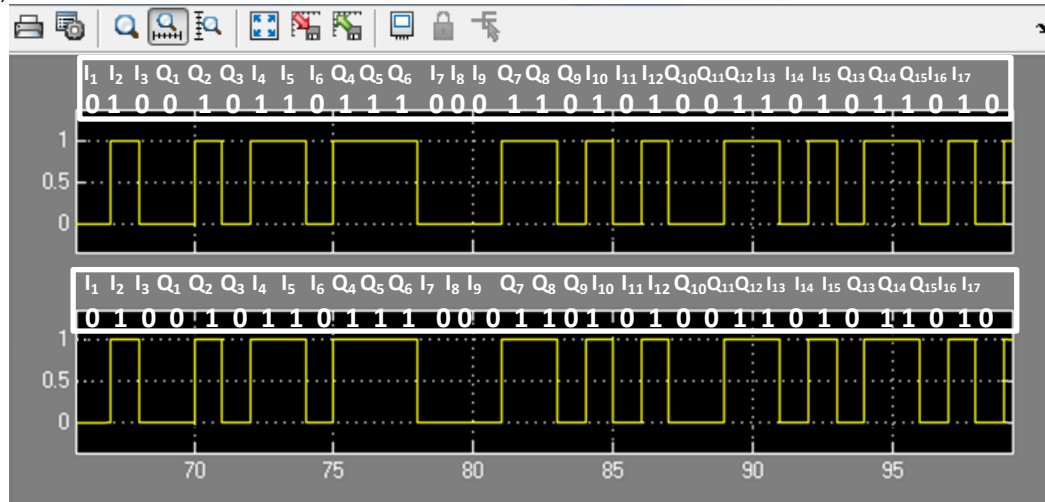


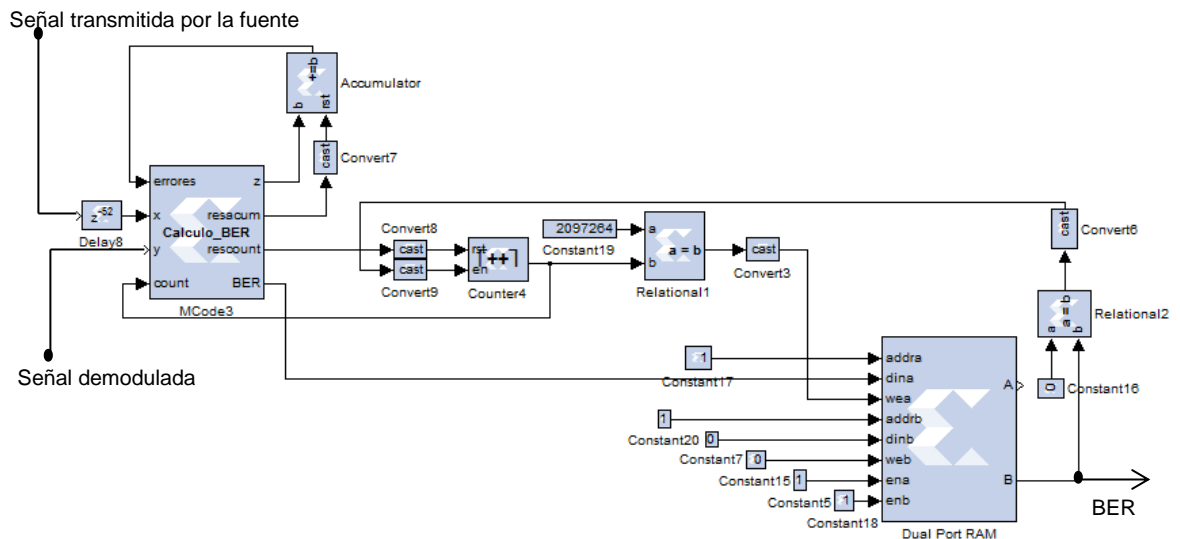
Figura 2.110. Secuencia de bits predefinida y secuencia de bits recuperada, ruido igual a cero, modulación 64 QAM



Bloque 10. Bloque Calculador de BER. La construcción de este bloque requiere el uso de un *Delay* que sincroniza la señal transmitida con la señal demodulada; un *Counter* que controla el número total de bits transmitidos; un *Accumulator* que almacena el número de bits errados de la señal demodulada; un *Mcode* que calcula el valor de la BER y un *Dual Port Ram* que almacena este valor.

Los anteriores elementos corresponden a los más importantes del diseño de la BER, sin embargo también se utilizan *Converts* que cambian el tipo de dato entre dos elementos, *Constants* que habilitan e indican los puertos de lectura y escritura del *Dual Port Ram* y *Relationals* que habilitan o deshabilitan los elementos utilizados (ver figura 2.111).

Figura 2.111. Calculador de BER



El bloque calculador de BER requiere la sincronización entre la señal transmitida y la señal demodulada con el fin de comparar correctamente una señal con la otra; dicha sincronización es llevada a cabo por un *Delay* con valor igual a $T_b(11M)$ en la señal transmitida.

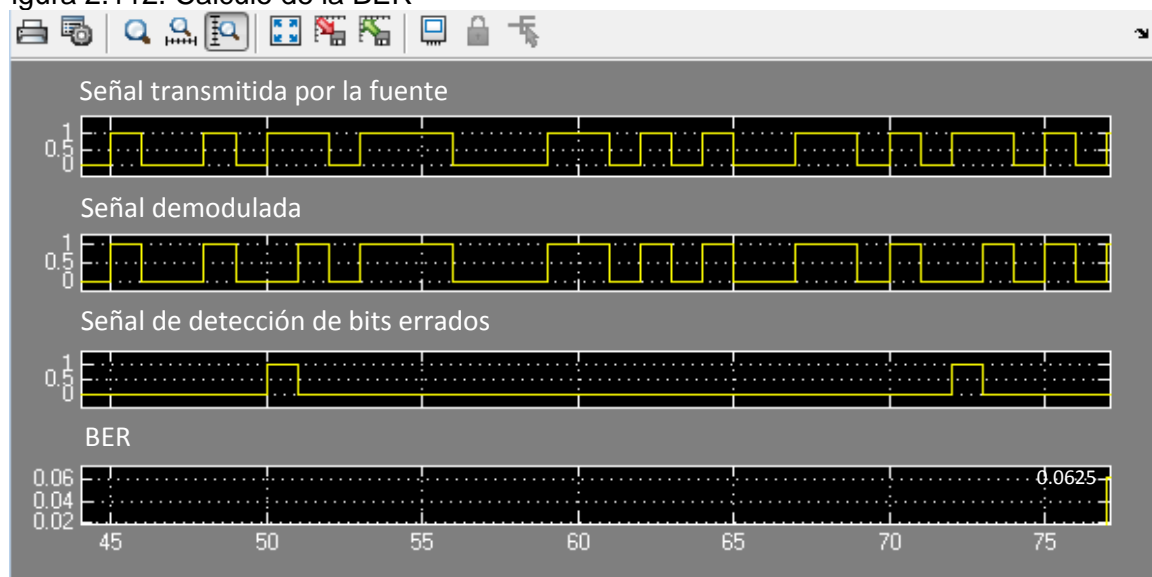
Tan pronto como se obtiene la sincronización de las señales, un *Mcode* compara estas señales y detecta diferencias entre ellas a nivel de bit, donde cada diferencia corresponde a un bit errado que se acumula en un *Accumulator*. Así, con base en la cantidad total de bits transmitidos y el número de bits errados almacenado en el *Accumulator*, este mismo *Mcode* calcula el valor de la BER correspondiente.

El valor de la BER calculado por el *Mcode* dura un instante corto de tiempo en el puerto de salida, motivo por el cual este valor es almacenado en un *Dual Port Ram* y posteriormente leído de forma constante. El almacenamiento del valor de la BER implica el uso de un *Relational* que habilita el *Dual Port Ram* en el momento preciso que se calcula el valor.

Debido a que el *Mcode* únicamente permite realizar divisiones entre potencias de dos, la cantidad total de bits transmitidos en el sistema básico de comunicaciones banda base con modulación 16/64 QAM debe ser también potencia de dos; esta cantidad es controlada por un *Counter* que indica constantemente este valor al *Mcode* y una vez se alcanza el número total de bits a transmitir, este elemento es deshabilitado por otro *Relational*. Para que el *Mcode* realice el procedimiento correspondiente, dentro del *Mcode* se desarrolla un código en lenguaje de programación Matlab® que realiza las acciones pertinentes.

El bloque calculador de BER permite realizar un seguimiento paso a paso del proceso: la figura 2.112 muestra la señal transmitida por la fuente de información, la señal demodulada, una señal de detección de bits errados y el valor de BER correspondiente.

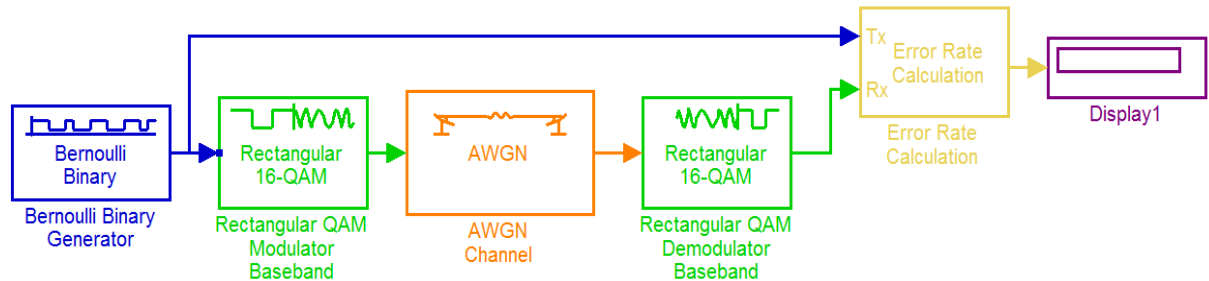
Figura 2.112. Cálculo de la BER



2.1.3.4 Validación del modelo en System Generator. La validación del correcto funcionamiento del modelo en System Generator se realizó por medio de la comparación de las curvas de desempeño obtenidas a partir del sistema básico de comunicaciones banda base con modulación 16/64 QAM simulado en System Generator y las curvas de desempeño obtenidas a partir de un modelo en Simulink® (ver figura 2.113).

En Simulink®, la fuente de información pseudoaleatoria se obtuvo a través del bloque *Bernoulli Binary Generator*, el bloque *Rectangular QAM Modulator Baseband* permitió la configuración tanto del orden de modulación como del tipo de mapeo Gray y la normalización en potencia; en el canal AWGN se asignó el valor de varianza²³ para un periodo de símbolo igual a $T_b M$, donde T_b corresponde a un segundo y en el *Rectangular QAM Demodulator Baseband* se configuró el algoritmo de decisión *Hard*²⁴.

Figura 2.113. Sistemas básicos de comunicaciones banda base con modulación 16/64 QAM en Simulink®



De este modo, las figuras 2.114 y 2.115 muestran las curvas de desempeño del sistema básico de comunicaciones banda base con modulación 16/64 QAM y mapeo Gray, simulado en Simulink® y System Generator, para una cantidad de bits transmitida igual a 32768 (2^{15}). Las superposiciones de las curvas de desempeño permiten asegurar que el modelo propuesto es consistente y por ende apto para el estudio de desempeño a nivel de implementación sobre hardware reconfigurable (FPGA). Las tablas de datos a partir de las cuales se graficaron las curvas de desempeño se muestran en los Apéndices B.1 y B.2.

²³ Se utilizó el valor de varianza (σ^2) en vez de la relación energía de bit a densidad espectral de potencia de ruido (E_b/N_o) directamente, en razón al diseño del canal AWGN complejo de la fase de simulación en System Generator. Aun así, al comparar los resultados obtenidos en Simulink® con estos dos parámetros, no hubo diferencia.

²⁴ Basado en el criterio de distancia mínima.

Figura 2.114. Curvas de desempeño a partir de Simulink® y System Generator, modulación 16 QAM

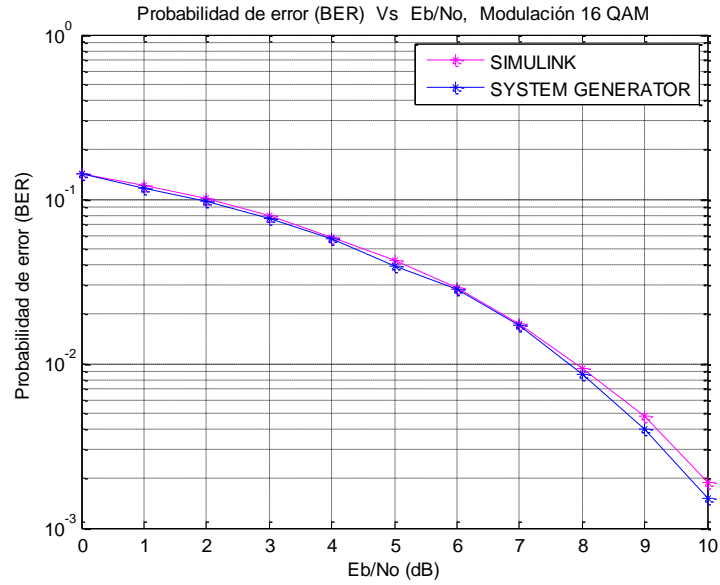
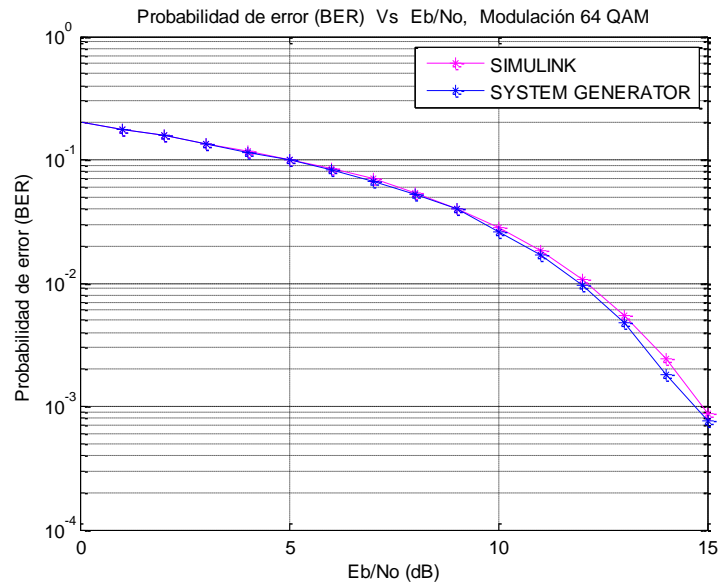


Figura 2.115. Curvas de desempeño a partir de Simulink® y System Generator, modulación 64 QAM

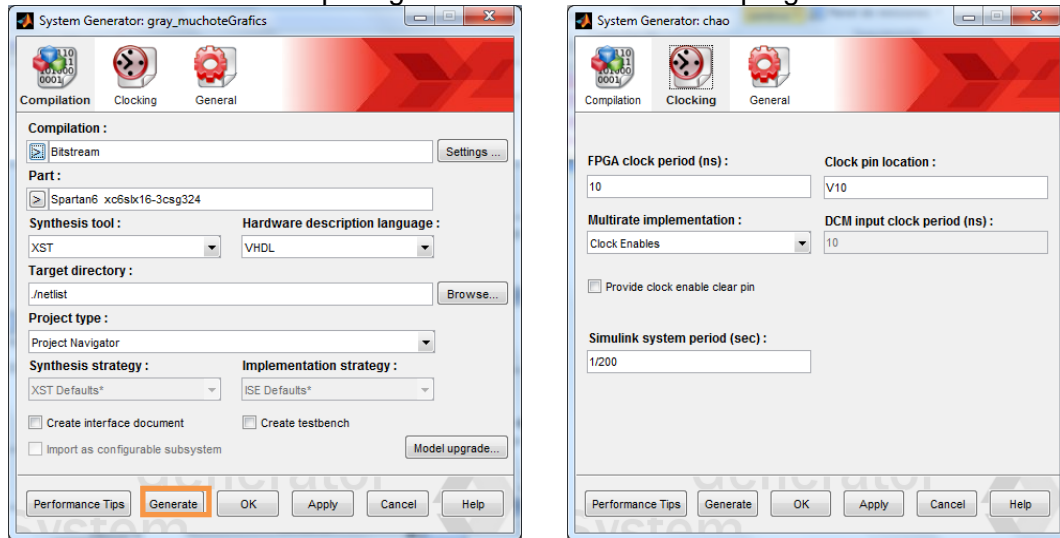


2.1.4 Implementación. Una vez validado el correcto funcionamiento del diseño en System Generator, a través de la simulación en Simulink®, el paso por seguir consistió en la generación del archivo de programación del hardware reconfigurable (FPGA), para lo cual se recurrió al elemento System Generator Token. La descarga del archivo sobre el FPGA se llevó a cabo por medio de la herramienta de programación iMPACT. De este

modo, la implementación del sistema básico de comunicaciones propuesto y simulado en la sección 2.1.3 fue llevada a cabo de manera secuencial para cada una de las modulaciones (16 QAM y 64 QAM), sobre un FPGA Spartan 6 de Xilinx haciendo uso de System Generator y la herramienta iMPACT del paquete de diseño ISE.

2.1.4.1 Generación del Archivo de Programación Bitstream (.bit). A través del *System Generator Token*, se configuraron los parámetros principales de implementación según el hardware reconfigurable (FPGA) utilizado. La figura 2.116 muestra la ventana de configuración para la generación del archivo de programación (.bit) del sistema básico de comunicaciones banda base.

Figura 2.116. Uso del Token para generación de archivos de programación



La generación del archivo de programación se llevó a cabo a través el botón *Generate*, para cada una de las modulaciones (16 QAM y 64 QAM) del sistema básico de comunicaciones banda base, configurando el parámetro *Compilation* como *Bitstream* con el fin de obtener directamente el archivo de programación sin la necesidad de generar archivos adicionales de evaluación del proceso de implementación.

El parámetro *Part* se configuró de acuerdo a la referencia del hardware reconfigurable (FPGA); por otro lado, los parámetros *Synthesis Tool* y *Hardware Description Language*, se configuraron con la Tecnología de Síntesis de Xilinx (XST, *Xilinx Synthesis Technology*) y VHDL respectivamente, ya que XST es la herramienta de síntesis ofrecida por Xilinx exclusivamente para trabajar de la mano con VHDL. El lenguaje de programación VHDL fue seleccionado dada la familiaridad que representa, aunque como tal, el código HDL (VHDL o Verilog) generado no requiere ser tratado. Las otras herramientas de síntesis (*Synplify* y *Synplify Pro*) se descartaron por pertenecer a otro proveedor (*Synplicity*) y no estar enfocada exclusivamente a la síntesis de VHDL. El *FPGA Clock Period (ns)* se configuró en 10 y el *Clock pin location* se configuró con V10 ya que el FPGA Spartan 6 cuenta con un oscilador de 100 MHz ubicado en el pin V10, y el *Simulink System Period (sec)* se configuró con un valor tal que al multiplicarse por la frecuencia del FPGA, se obtuviera la velocidad de transmisión de datos del sistema básico de comunicaciones banda base. De este modo se completó la configuración de las

herramientas que intervinieron en la representación del modelo en bajo nivel. Las tablas 2.9 y 2.10 muestran los recursos requeridos para la implementación del sistema básico de comunicaciones banda base con modulación 16/64 QAM

Tabla 2.9. Recursos requeridos por el sistema básico de comunicaciones banda base con modulación 16QAM

Resumen de Utilización del Dispositivo			
Parte Lógica Utilizada	Usado	Disponible	Utilización
<i>Number of Slice Registers</i>	1720	18224	9%
<i>Number of Slice LUTs</i>	1368	9112	15%
<i>Number used as logic</i>	789	9112	8%
<i>Number used as Memory</i>	528	2176	24%
<i>Number of occupied Slices</i>	590	2278	25%
<i>Number of MUXCYs used</i>	184	4556	4%
<i>Number with an unused Flip Flop</i>	455	1613	28%
<i>Number with an unused LUT</i>	245	1613	15%
<i>Number of fully used LUT-FF pairs</i>	913	1613	56%
<i>Number of slice register sites lost to control set restrictions</i>	45	18224	1%
<i>Number of bonded IOBs</i>	22	232	9%
<i>Number of LOCed IOBs</i>	22	22	100%
<i>Number of RAMB16BWERs</i>	1	32	3%
<i>Number of RAMB8BWERs</i>	16	64	25%
<i>Number of DSP48A1s</i>	24	32	75%
<i>Average Fanout of Non-Clock Nets</i>	2,12		

Tabla 2.10. Recursos requeridos por el sistema básico de comunicaciones banda base con modulación 64QAM

Resumen de Utilización del Dispositivo			
Parte Lógica Utilizada	Usado	Disponible	Utilización
<i>Number of Slice Registers</i>	1773	18224	9%
<i>Number of Slice LUTs</i>	1379	9112	15%
<i>Number used as logic</i>	796	9112	8%
<i>Number used as Memory</i>	528	2176	24%
<i>Number of occupied Slices</i>	640	2278	28%
<i>Number of MUXCYs used</i>	184	4556	4%
<i>Number with an unused Flip Flop</i>	464	1664	27%
<i>Number with an unused LUT</i>	285	1664	17%
<i>Number of fully used LUT-FF pairs</i>	915	1664	54%
<i>Number of slice register sites lost to control set restrictions</i>	63	18224	1%
<i>Number of bonded IOBs</i>	22	232	9%
<i>Number of LOCed IOBs</i>	22	22	100%
<i>Number of RAMB16BWERs</i>	1	32	3%
<i>Number of RAMB8BWERs</i>	16	64	25%
<i>Number of BUFG/BUFGMUXs</i>	1	16	6%
<i>Number of DSP48A1s</i>	24	32	75%
<i>Average Fanout of Non-Clock Nets</i>	2,12		

A lo largo de la implementación se evidenció la similitud en requerimientos de recursos para cada una de las modulaciones (16 QAM y 64 QAM) del sistema básico de comunicaciones banda base, donde el porcentaje de utilización en las tablas 2.9 y 2.10 varió ligeramente.

2.1.4.2 Programación del hardware reconfigurable (FPGA). La programación del hardware reconfigurable (FPGA) se realizó a través del iMPACT, que permitió la descarga del archivo de programación (.bit) sobre el FPGA, la figura 2.117 muestra el mensaje de programación exitosa.

La figura 2.118 tomada de un osciloscopio, muestra la señal de entrada y la señal de salida (obtenidas a partir de implementación) del sistema básico de comunicaciones banda base, con un ruido de canal igual a cero, donde la señal de color amarillo corresponde a la señal de entrada y la señal de color azul corresponde a la señal de salida o demodulada. Estas dos señales fueron las mismas tanto para la modulación 16 QAM como para la modulación 64QAM del sistema de comunicaciones.

Figura 2.117. Programación del FPGA

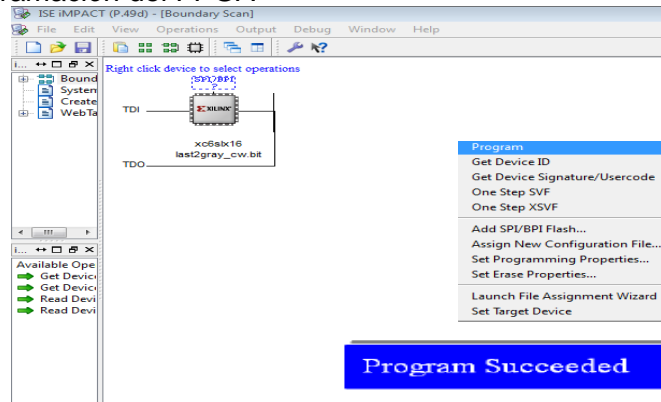


Figura 2.118. Señales de entrada y salida obtenidos a partir de implementación, sistema básico de comunicaciones banda base con modulación 16/64 QAM



3. PLAN DE PRUEBAS, RESULTADOS Y ANÁLISIS

Una vez validado el correcto funcionamiento en simulación del sistema básico de comunicaciones banda base con modulación 16/64 QAM y realizada la implementación sobre hardware reconfigurable (FPGA), el plan de pruebas inició con el estudio del comportamiento del sistema básico de comunicaciones propuesto ante la variación de tres parámetros concretos: velocidad de transmisión de datos, tipo de mapeo y cantidad de bits transmitidos. Posteriormente se compararon los resultados obtenidos en simulación, implementación y valores teóricos ofrecidos por la herramienta Bertool²⁵ de Matlab®, para su respectivo análisis junto con las apreciaciones realizadas durante el capítulo 2.

3.1 PLAN DE PRUEBAS

El plan de pruebas consistió en la variación de tres parámetros propios del sistema básico de comunicaciones banda base con modulación 16/64 QAM propuesto, a efectos de observar su desempeño tanto en simulación como en implementación. Los parámetros considerados fueron: Tipo de mapeo, velocidad de transmisión de datos y cantidad de bits transmitidos. La tabla 3.1 presenta el plan de pruebas establecido.

Tabla 3.1. Plan de pruebas

Prueba	Descripción	Parámetros					
		Velocidad de transmisión de datos (Kbps)		Tipo de mapeo		Cantidad de bits transmitidos	
		500	10000	Gray	Binario	32768	1048576
1	Variación de la velocidad de transmisión de datos	✓		✓		✓	
			✓	✓		✓	
2	Variación del tipo de mapeo	✓		✓		✓	
		✓			✓	✓	
3	Variación de la cantidad de bits transmitidos	✓		✓		✓	
		✓		✓			✓

²⁵ Interfaz Gráfica de Usuario (GUI, *Graphical User Interface*) ofrecida por Matlab® para el análisis del desempeño de sistemas de comunicaciones a partir de la BER. Esta herramienta considera mapeo tipo Gray, el algoritmo de decisión Hard y la función de probabilidad de error (P_e) para el cálculo de la BER descrita y analizada en [16], para el trazado de las curvas de desempeño teóricas.

3.2 RESULTADOS

A partir del plan de pruebas planteado en la tabla 3.1, se obtuvieron los correspondientes resultados para cada una de las pruebas estimadas, tanto en simulación a partir de *System Generator* como en implementación sobre hardware reconfigurable (FPGA), siguiendo el método de Monte Carlo.

Dado que las variaciones obtenidas en cada una de las pruebas fueron mínimas (evidenciándose una pequeña varianza muestral) se estableció que, en este caso, 60 ejecuciones fueron suficientes para obtener resultados que presentaran una tendencia estadística definida.

3.2.1 Prueba 1. Variación de la velocidad de transmisión de datos. Para una cantidad de 32768 bits transmitidos y mapeo Gray, las figuras 3.1 y 3.2, muestran los resultados de simulación en *System Generator* e implementación sobre hardware reconfigurable (FPGA), respectivamente, del desempeño del sistema básico de comunicaciones banda base con modulación 16 QAM, con cada una de las velocidades de transmisión de datos: 500 Kbps y 10 Mbps. Aunque las gráficas mostradas corresponden únicamente a la modulación 16 QAM, los resultados tuvieron el mismo comportamiento para la modulación 64 QAM.

La tabla de datos a partir de la cual se graficaron las curvas de desempeño se muestra en el Apéndice B.1.

Figura 3.1. Curvas de desempeño para velocidades de transmisión de datos 500 Kbps y 10 Mbps, simulación en *System Generator* con modulación 16 QAM

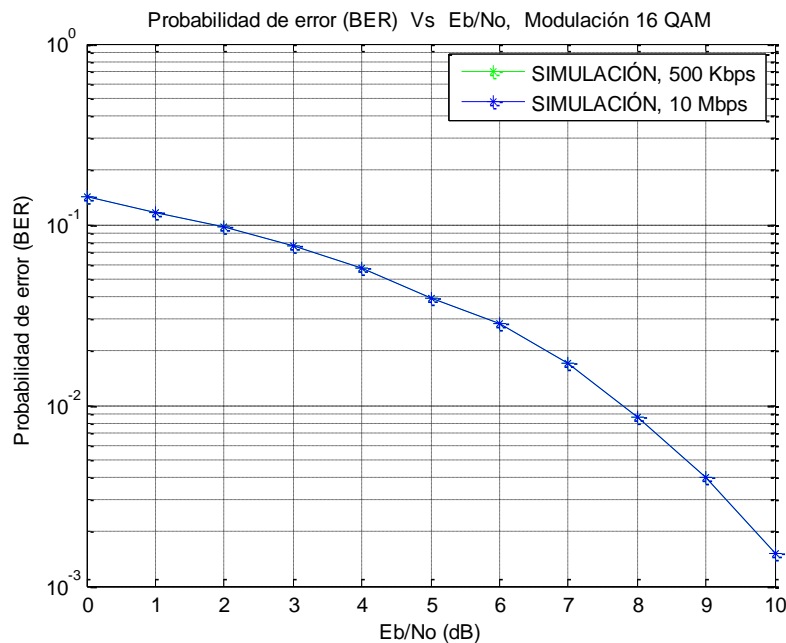
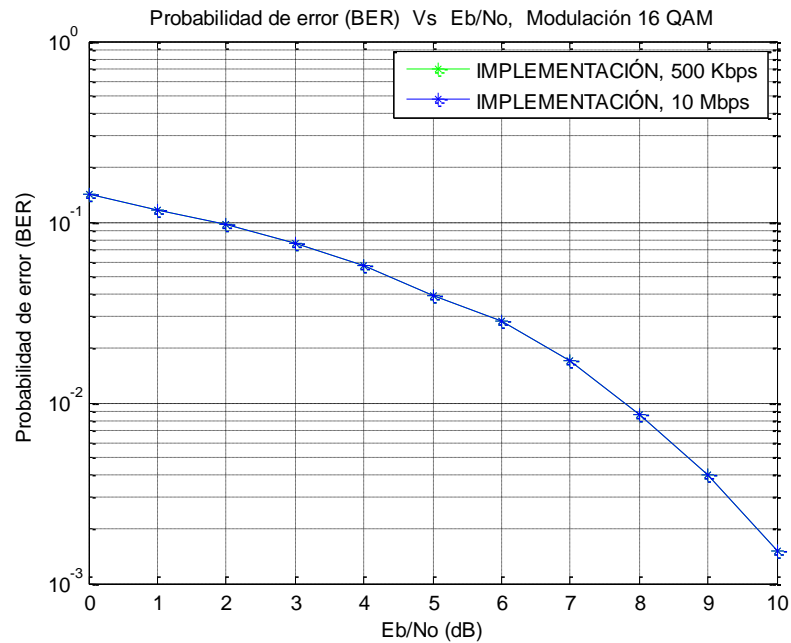


Figura 3.2. Curvas de desempeño para velocidades de transmisión de datos 500 Kbps y 10 Mbps, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM



3.2.2 Prueba 2. Variación del tipo de mapeo. Para una cantidad total de 32768 bits transmitidos y una velocidad de transmisión de datos de 500 Kbps, las figuras 3.3 y 3.4 muestran los resultados de simulación en System Generator del desempeño del sistema básico de comunicaciones banda base con modulación 16/64 QAM, para cada uno de los dos tipos de mapeo: Gray y Binario; Asimismo las figuras 3.5 y 3.6 muestran los resultados de implementación sobre hardware reconfigurable (FPGA).

Las tablas de datos a partir de las cuales se graficaron las curvas de desempeño se muestran en los Apéndices B.1, B.2, B.3, B.4.

Figura 3.3. Curvas de desempeño mapeos Gray y Binario, simulación en System Generator con modulación 16 QAM

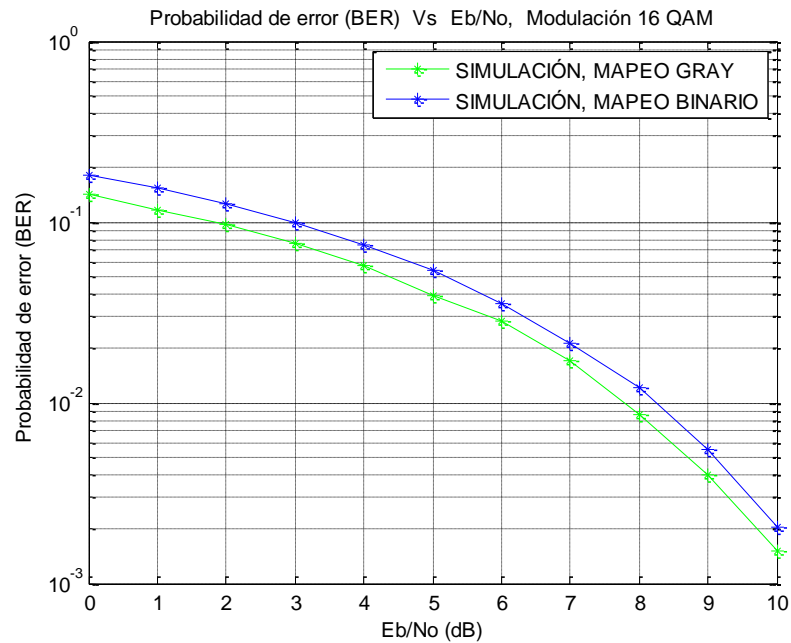


Figura 3.4. Curvas de desempeño mapeos Gray y Binario, simulación en System Generator con modulación 64 QAM

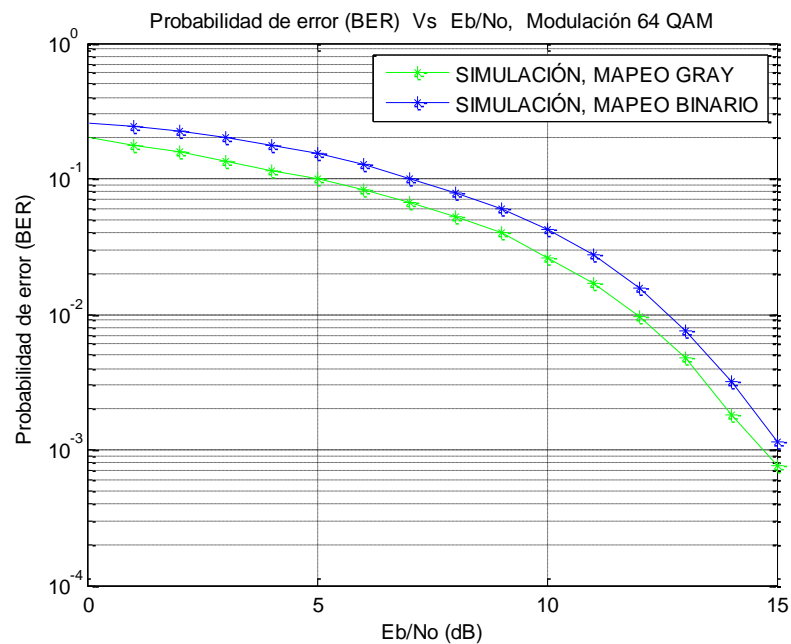


Figura 3.5. Curvas de desempeño mapeos Gray y Binario, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM

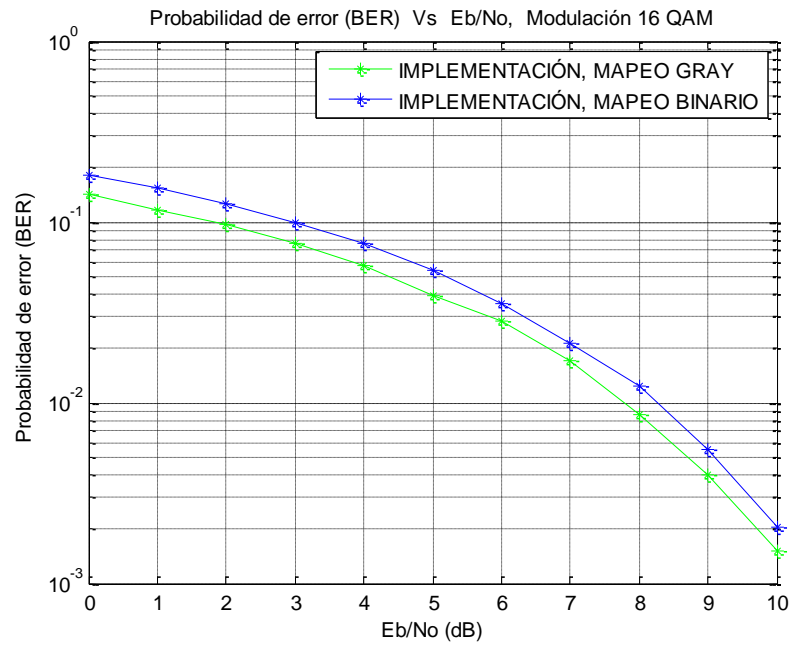
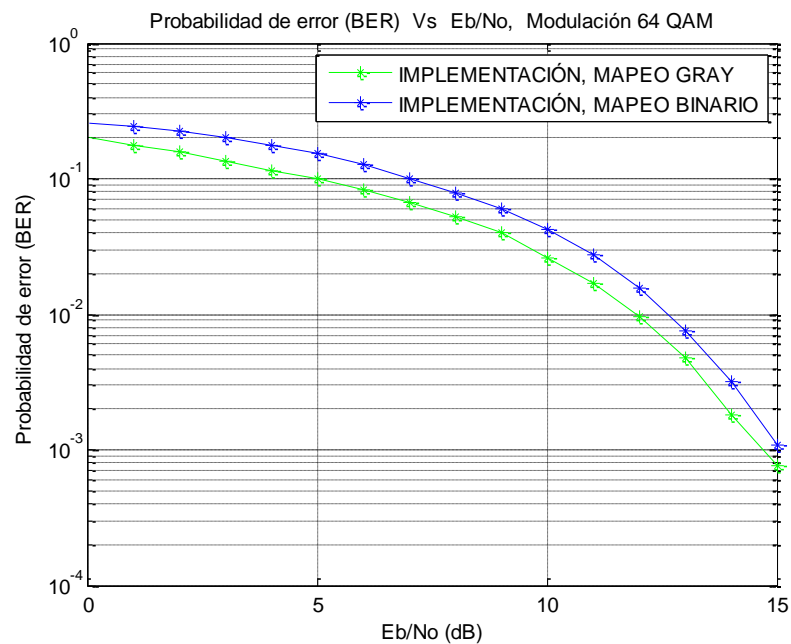


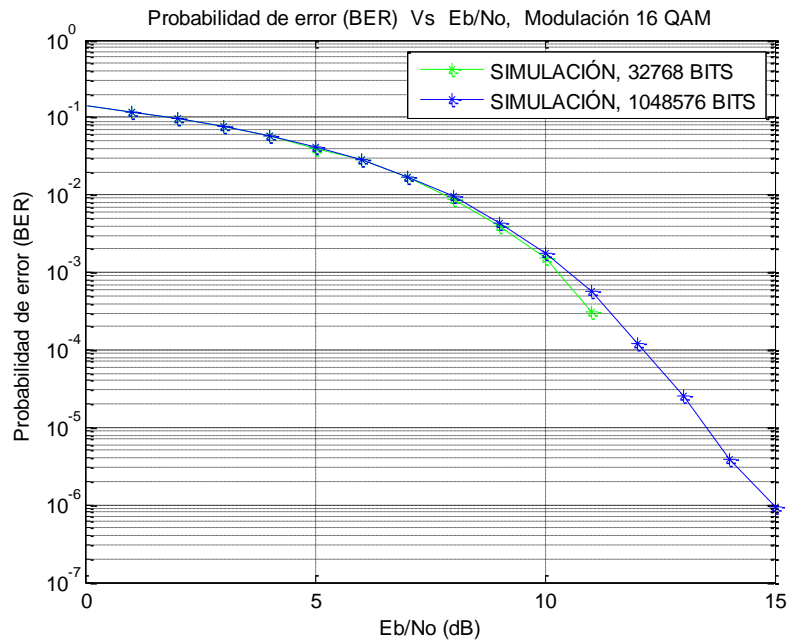
Figura 3.6. Curvas de desempeño mapeos Gray y Binario, implementación sobre hardware reconfigurable (FPGA) con modulación 64 QAM



3.2.3 Prueba 3. Variación de la cantidad de bits transmitidos. Para una velocidad de transmisión de datos de 500 Kbps y mapeo Gray, las figuras 3.7 y 3.8 muestran los resultados de simulación en System Generator e implementación sobre hardware reconfigurable (FPGA), respectivamente, del desempeño del sistema básico de comunicaciones banda base con modulación 16 QAM, para cada una de las cantidades de bits transmitidos²⁶: 32768 y 1048576²⁷. Aunque las gráficas mostradas corresponden únicamente a la modulación 16 QAM, los resultados tuvieron la misma tendencia para la modulación 64 QAM

Las tablas de datos a partir de las cuales se graficaron las curvas de desempeño se muestran en los Apéndices B.1, B.2, B.5, B6.

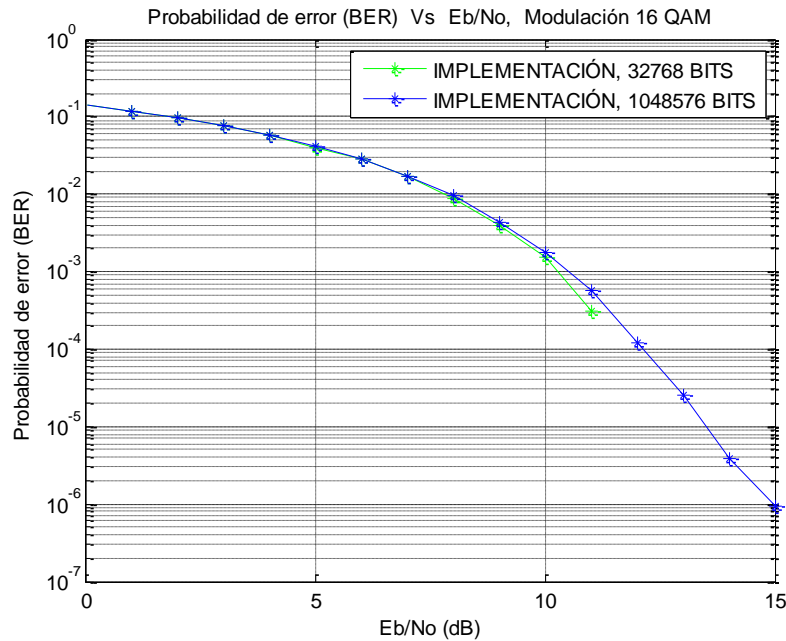
Figura 3.7. Curvas de desempeño para transmisión de 32768 y 1048576 bits, simulación en System Generator con modulación 16 QAM



²⁶ Considerando una transmisión de bits mucho mayor a 32768 (cantidad inicial), con el fin de observar su efecto sobre el trazado de la curva de desempeño, se selecciona la transmisión de 1048576 bits, lo correspondiente a 32 veces la cantidad de bits inicial (32768 bits). Tanto 32768 como 1048576 corresponden a valores que son a potencias de 2, dada la restricción del calculador de BER para procesar cantidades de bits con esta característica.

²⁷ Los valores de mínima BER alcanzable diferente de cero para 32768 y 1048576 bits son 10^{-5} y 10^{-7} respectivamente. Para el caso de 32768 bits a nivel de simulación e implementación los últimos valores obtenidos de BER fueron cero y por esta razón las curvas alcanzaron valores mínimos diferentes de cero aproximados a 10^{-4} .

Figura 3.8. Curvas de desempeño para transmisión de 32768 y 1048576 bits, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM



3.3 ANÁLISIS

La presente sección se centra en el análisis de todos los resultados obtenidos y apreciaciones realizadas a lo largo del desarrollo del presente trabajo de grado.

3.3.1 Señales de entrada y salida del sistema básico de comunicaciones banda base con modulación 16/64 QAM. De la figura 2.118 de la sección 2.1.4.2, donde se muestra la señal de salida del sistema básico de comunicaciones banda base con modulación 16/64 QAM a partir de la implementación sobre hardware reconfigurable (FPGA), se identificaron dos aspectos importantes: Sincronización y correspondencia en forma con la señal proveniente de la fuente de información, aspectos que siendo completamente diferentes se debieron a la capacidad del hardware reconfigurable (FPGA) para reproducir exactamente cada una de las configuraciones establecidas en simulación, especialmente en materia de retardos, donde el valor exacto de retardo de los *Delays* se reprodujo en hardware. El caso más diciente lo representa el bloque calculador de BER, donde el retardo entre las señales de entrada y salida del sistema básico de comunicaciones banda base fue idéntico en simulación e implementación.

3.3.2 Prueba 1. Variación de la velocidad de transmisión de datos. De los resultados obtenidos en simulación en System Generator e implementación sobre hardware reconfigurable (FPGA), con respecto a la variación en la velocidad de transmisión de datos, dada la superposición de las curvas de desempeño con velocidades de transmisión de datos de 500 Kbps y 10Mbps de las figuras 3.1 y 3.2, se pudo verificar la independencia del sistema básico de comunicaciones banda base con modulación 16/64 QAM a las variaciones de este parámetro, ello en virtud a que el sistema básico de

comunicaciones banda base no considera efecto de multitrayectoria o corrimiento en frecuencia y no considera limitación en ancho de banda que hagan que el canal se comporte como un sistema de comunicaciones selectivo en frecuencia, con lo cual pudo obviarse la adición de una etapa de ecualización y la velocidad de transmisión de datos no dio lugar a ISI, un factor determinante en la demodulación.

3.3.3 Prueba 2. Tipo de Mapeo. De los dos tipos de mapeo utilizados en el proceso de modulación, la figura 3.5 permitió observar que en el caso de la modulación 16 QAM del sistema básico de comunicaciones banda base implementado sobre hardware reconfigurable (FPGA), para una BER objetivo de 10^{-2} , la ganancia de codificación del mapeo Gray frente al Binario fue de 0.47 dB, lo correspondiente a un ahorro de energía aproximado al 6.05%; de la misma manera, en el caso de la modulación 64QAM del sistema básico de comunicaciones banda base implementado sobre hardware reconfigurable (FPGA), para una BER objetivo de 10^{-2} , la ganancia de codificación del mapeo Gray frente al Binario fue de 0.67 dB, lo correspondiente a un ahorro de energía aproximado al 5.65%.

Lo anterior en razón a que la diferencia entre símbolos vecinos corresponde a un bit para el mapeo Gray, caso contrario al mapeo Binario, donde puede encontrarse variaciones de más de un bit (entre símbolos vecinos) dando lugar a la posibilidad de mayor cantidad de bits errados por símbolo errado que se demodule.

3.3.4 Prueba 3. Cantidad de bits transmitidos. Adicional al comportamiento de las curvas de desempeño de las figuras 3.7 y 3.8, donde consistentemente las gráficas presentaron variaciones en los últimos valores de BER para un valor de E_b/N_o dado, respecto a la curva teórica de Bertool®; las figuras 3.9 y 3.10 muestran que la curva de desempeño resultante de la transmisión de 1048576 bits presenta una estimación más confiable de la probabilidad de error de bit, ello debido a que el número de transmisiones independientes es mayor que para el caso de 32768 bits, lo cual conlleva a una mayor precisión en el cálculo de la frecuencia relativa del evento “error”.

De este modo, de acuerdo a lo planteado, una mayor cantidad de bits transmitidos implica una mayor precisión en la BER calculada²⁸, y aunque los sistemas de comunicaciones propuestos soportan la transmisión de una mayor cantidad de bits, la carga computacional es alta y los resultados obtenidos con 1048576 bits fueron suficientes para el cumplimiento de los objetivos del presente trabajo de grado.

Las tablas de datos a partir de las cuales se graficaron las curvas de desempeño se muestran en los Apéndices B.1, B.5.

²⁸ Teniendo en cuenta el método de Montecarlo, el cual se fundamenta en el teorema del límite central, se considera que a mayor cantidad de bits transmitidos, los resultados presentan mayor validez estadística.

Figura 3.9. Curvas de desempeño para transmisión de 32768 y 1048576 bits, simulación en System Generator con modulación 16 QAM

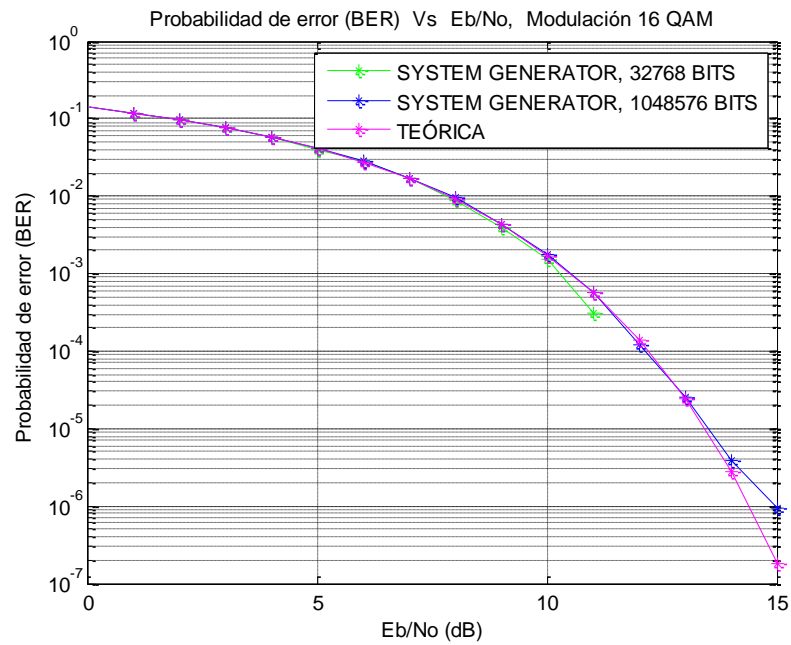
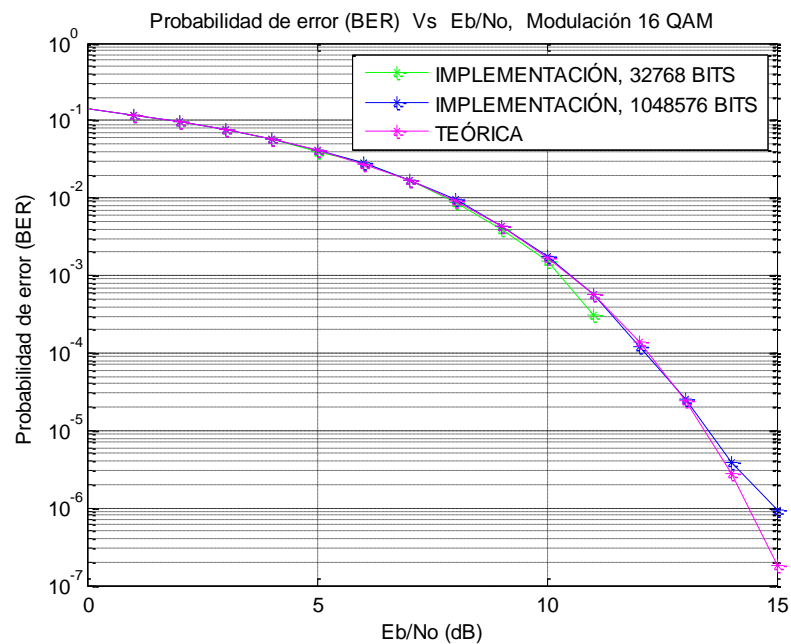


Figura 3.10. Curvas de desempeño para transmisión de 32768 y 1048576 bits, implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM



3.3.5 Curvas de desempeño. Teniendo en cuenta lo plateado en los análisis anteriores, a efectos de obtener curvas de desempeño con mejores condiciones y por ende lo más similares posible a la curva de desempeño teórica, para este análisis se considera el uso del mapeo Gray y la transmisión de 1048576 bits así como una velocidad de transmisión de datos igual a 10Mbps; que bien pudo haber sido 500Mbps, dado que su valor no afecta el desempeño del sistema básico de comunicaciones banda base.

Las figuras 3.11 y 3.12 muestran la comparación de las curvas desempeño del sistema básico de comunicaciones banda base con modulación 16/64 QAM obtenidas a partir de simulación en System Generator y a partir de implementación sobre hardware reconfigurable (FPGA), donde se apreció una clara correspondencia y estrecha relación entre los resultados obtenidos, en cada una de las figuras, dada la superposición presentada entre las curvas de desempeño; de este modo, se verificó que la implementación generó resultados coherentes de acuerdo a los modelos planteados. Asimismo, las figuras 3.13 y 3.14 muestran los resultados obtenidos a partir de implementación comparados con las curvas de desempeño teóricas ofrecidas por Bertool®. Para este par de imágenes se pudo apreciar también una clara correspondencia y estrecha relación entre cada par de curvas de desempeño (implementación y teórica), por lo cual se confirmó que tanto a nivel de simulación e implementación, el sistema básico de comunicacinoes bada base con modulación 16/64 QAM presentó un comportamiento óptimo al replicar de manera muy similar las curvas de desempeño teróricas ofrecidas por Bertool®.

Finalmente, la figura 3.15 muestra simultáneamente las curvas de desempeño del sistema básico de comunicaciones banda base con modulación 16/64 QAM obtenidas a partir de implementación sobre hardware reconfigurable (FPGA), permitiendo corroborar que para menor cantidad de estados (16 QAM) se presenta mayor robustez frente al ruido, lo cual se traduce en menor requerimiento de E_b/N_o ²⁹ para una misma BER. Consecuentemente, se pudo apreciar que pese a la utilización del mismo tipo de mapeo, la curva de desempeño correspondiente a la modulación 16QAM del sistema básico de comunicaciones banda base obtuvo un ahorro energía (aproximadamente 50.87% menos energía para una BER objetivo de 10^{-2}) con respecto a la curva de desempeño correspondiente a la modulación 64QAM del sistema básico de comunicaciones banda base. Por otro lado, la separación entre las curvas de desempeño para los últimos valores de E_b/N_o (valores más grandes), mayor que para los primeros valores, se debió a la exigencia de mayor cantidad de energía por símbolo, lo cual se hace mucho más estricto en el caso de la modulación 64 QAM en razón a la cercanía que presentan los símbolos de la constelación.

Las tablas de datos a partir de las cuales se graficaron las curvas de desempeño se muestran en los Apéndices B.5 y B.6.

²⁹ Es decir, menor cantidad de energía.

Figura 3.11. Comparación curvas de desempeño, simulación en System Generator e implementación sobre hardware reconfigurable (FPGA) con modulación 16 QAM y mapeo Gray

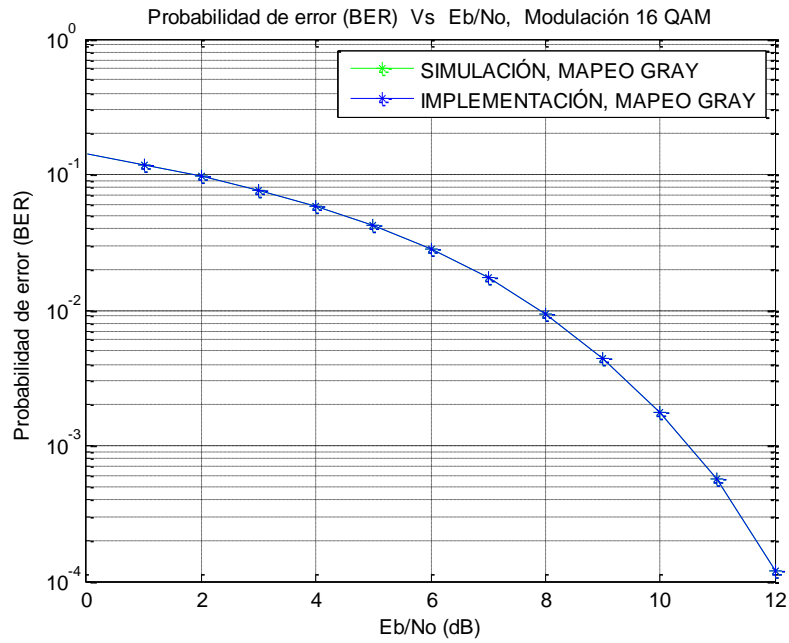


Figura 3.12. Comparación curvas de desempeño, simulación en System Generator e implementación sobre hardware reconfigurable (FPGA) con modulación 64 QAM y mapeo Gray

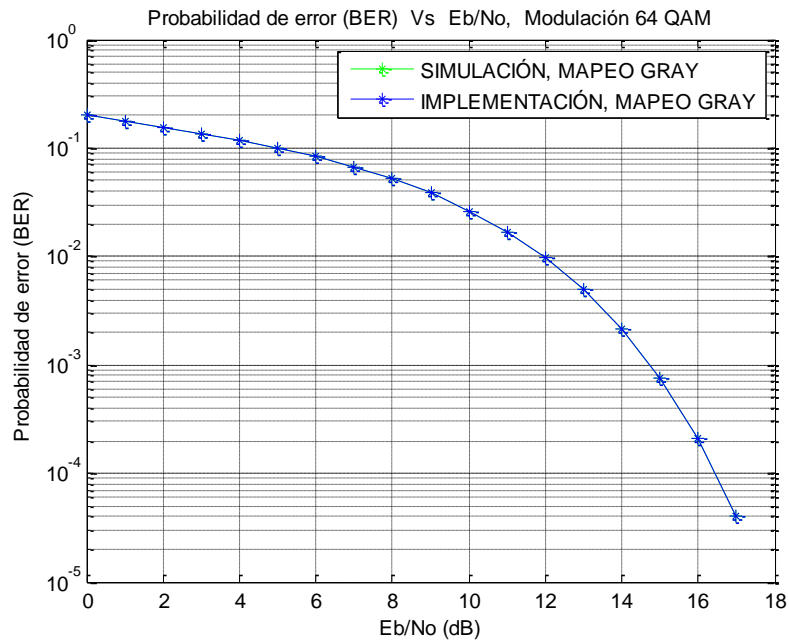


Figura 3.13. Comparación curvas de desempeño, implementación sobre hardware reconfigurable (FPGA) y teórica con modulación 16 QAM y mapeo Gray

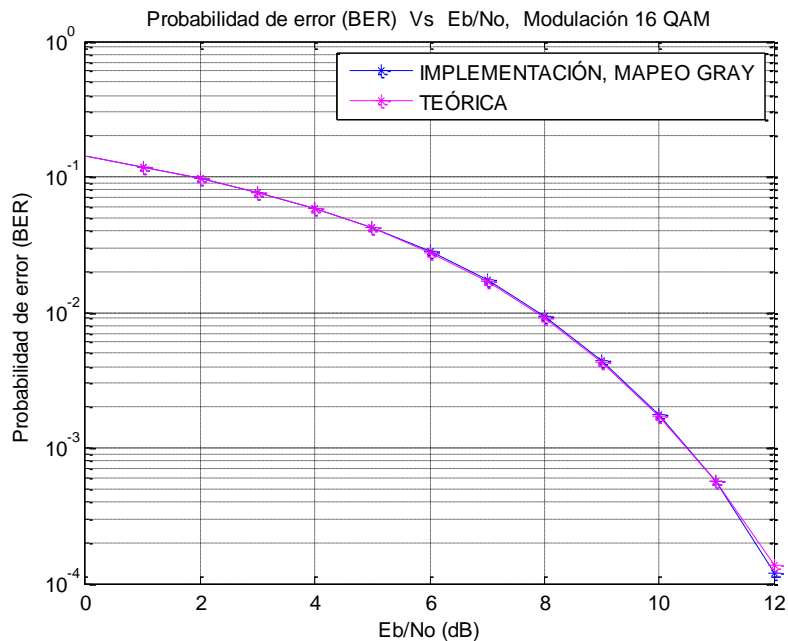


Figura 3.14. Comparación curvas de desempeño, implementación sobre hardware reconfigurable (FPGA) y teórica con modulación 64 QAM y mapeo Gray

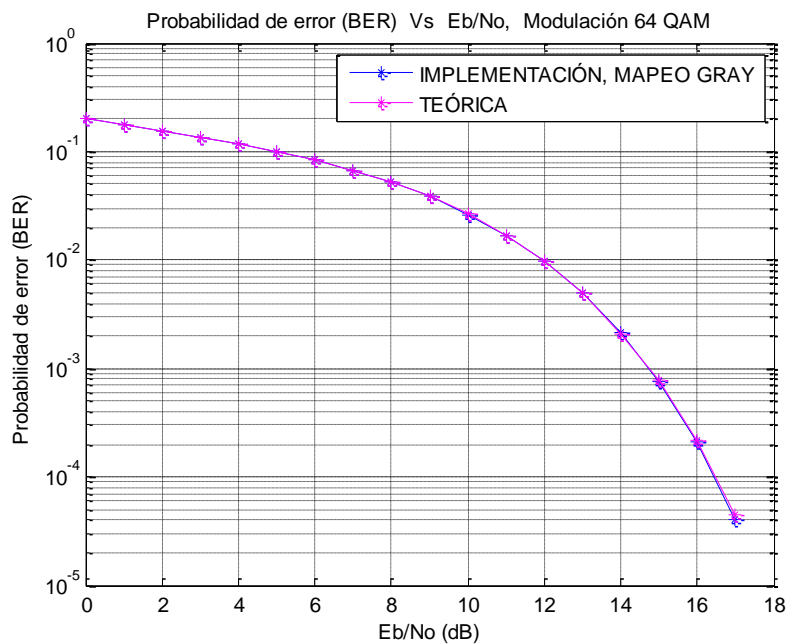
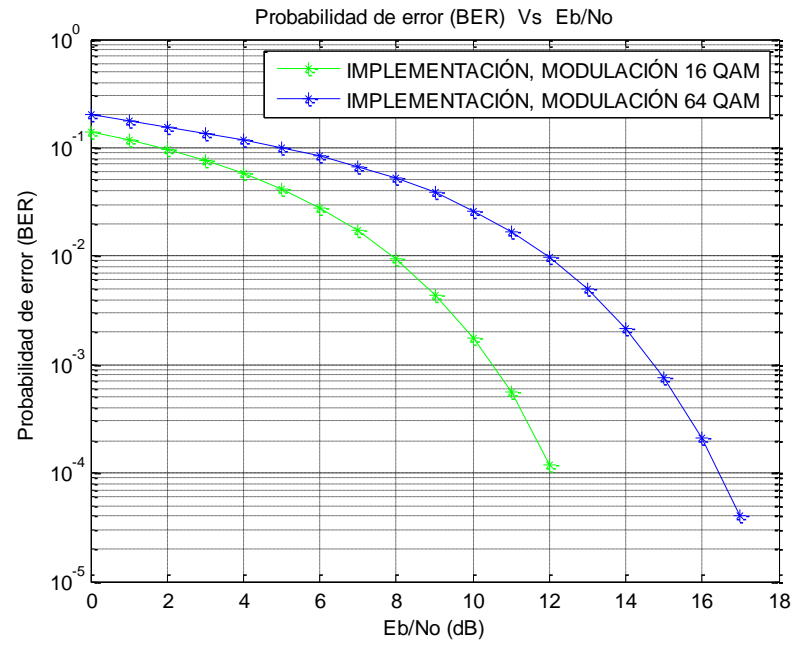


Figura 3.15. Comparación curvas de desempeño, implementación sobre hardware reconfigurable (FPGA) con modulación 16/64 QAM y mapeo Gray



4. CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

4.1 CONCLUSIONES

- La simulación e implementación sobre hardware reconfigurable (FPGA) conforman la base para herramientas de apoyo a procesos de enseñanza/aprendizaje en el área de los sistemas de comunicaciones con esquemas de modulación digital.
- A nivel de diseño, tanto 16 QAM como 64 QAM representan la misma complejidad, pues los bloques ofrecidos por System Generator permiten modificar las configuraciones para cada una de las modulaciones sin la necesidad de alterar el modelo base, funcional para los esquemas de modulación 16 QAM y 64 QAM.
- El diseño en System Generator de dos canales AWGN paralelos (real e imaginario) con diferente valor de semilla en cada uno de los WGNs, permitió cumplir los requerimientos que debe tener todo proceso de ruido AWGN complejo, donde las componentes de señal real e imaginaria deben ser no correlacionadas.
- Para cada una de las modulaciones, 16 QAM y 64 QAM, el mapeo Gray introdujo ganancia de codificación respecto al mapeo Binario, generando consigo un ahorro de energía mayor en 64 QAM que en 16 QAM.
- Para una BER objetivo de 10^{-2} , la modulación 16QAM con mapeo Gray obtuvo un ahorro de energía aproximado al 50.87% respecto a la modulación 64QAM con mapeo Gray.
- Dada la obtención de curvas de desempeño muy similares a las curvas de desempeño en System Generator, Simulink® y Bertool®, el sistema básico de comunicaciones banda base con modulación 16/64 QAM alcanzó resultados óptimos de implementación sobre hardware reconfigurable (FPGA).

4.2 RECOMENDACIONES

- El bloque Mcode de System Generator no permite la implementación de algoritmos complejos, únicamente operaciones básicas. Sin embargo es una buena opción para realizar procesos lógicos con pequeños algoritmos.
- Cada implementación genera archivos temporales que agilizan este proceso en caso de futuras pruebas, de este modo, pueden presentarse conflictos durante la implementación de un nuevo modelo. Por esta razón se recomienda eliminar constantemente la carpeta de archivos temporales del sistema operativo.
- Partiendo de la instalación del paquete de diseño ISE sobre el directorio raíz del sistema operativo, la ubicación del archivo de simulación (.mdl) debe tener una ruta de almacenamiento corta, de lo contrario se presentan problemas al momento de la simulación.

4.3 TRABAJOS FUTUROS

- Uso de la figura de mérito Magnitud del Vector de Error (EVM, *Error Vector Magnitude*) para evaluación del desempeño del sistema básico de comunicaciones banda base con modulación 16/64 QAM sobre hardware reconfigurable (FPGA).
- Separación de los bloques transmisor y receptor del sistema básico de comunicaciones banda base con modulación 16/64 QAM e implementación sobre hardware reconfigurable (FPGA) recurriendo a un protocolo de comunicación.
- Adición de los bloques de codificación de fuente y canal, al sistema básico de comunicaciones banda base con modulación 16/64 QAM implementado sobre hardware reconfigurable (FPGA).
- Adición de técnicas de transmisión: Acceso Múltiple Por División De Código (CDMA, *Code Division Multiple Access*) y Multiplexación por División de Frecuencia Ortogonal (OFDM, *Orthogonal Frequency Division Multiplexing*) al sistema básico de comunicaciones banda base con modulación 16/64 QAM.
- Construcción del sistema de comunicaciones con modulación 16/64 QAM completo, sobre hardware reconfigurable (FPGA), con sus respectivas etapas banda base y pasa banda.
- Análisis del desempeño de un sistema de comunicaciones con modulación 16/64 QAM implementado sobre hardware reconfigurable (FPGA), con un canal multitrayecto y corrimiento en frecuencia para comunicación móvil.

BIBLIOGRAFÍA

- [1] V. Miramá, "El Sistema de Comunicaciones". En: "Teoría De Telecomunicaciones I, Sistemas Analógicos", Departamento de Telecomunicaciones, Universidad del Cauca, 2010.
- [2] J. Ramirez, J. Moreno, "Análisis del Desempeño de la Modulación Wavelet", Popayán, 2009, Trabajo de grado (Ingeniería en Electrónica y Telecomunicaciones), Universidad del Cauca, Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Telecomunicaciones
- [3] R. Gallager, "Channels Modulations and Demodulations". En: "Course Materials for 6.450 Principles of Digital Communications I", MIT OpenCourseWare (<http://ocw.mit.edu/>), Massachusetts Institute of Technology, 2006. Descargado el [24 de Enero de 2014].
- [4] X. López, C. Mora "Análisis de Técnicas de Modulación Adaptiva en Redes Inalámbricas de Banda Ancha (ieee 802.16, wimax)", Ciudad de México D. F., 2006. Trabajo de grado (Ingeniería en Telecomunicaciones). Universidad Nacional Autónoma de México, Facultad de Ingeniería.
- [5] M. Faúndez, "Probabilidad de error de bit y probabilidad de error de símbolo". En: _____, "Sistemas de comunicaciones", [en línea]. Marcobo, 2001. 171 p. Disponible en:
http://books.google.com.co/books?id=_arH8J1d1FYC&pg=PA171&lpg=PA171&dq=probabilidad+de+error+simbolos+adyacentes&source=bl&ots=ZJOhGvh5ke&sig=T1a0qebp_aCH21bbgM3B6LgL8Cg&hl=es-419&sa=X&ei=qGbUUs_-AuahsQTwwYHwDQ&ved=0CCsQ6AEwAA#v=onepage&q&f=false. [Accedido: 24 de Enero de 2014].
- [6] P. Prandoni, M. Vetterli, "Design of a Digital Communication System". En: _____, "Signal Processing for Communications", [en línea]. EFPL Press, 2008. p 327-365. Disponible en:
<http://books.google.com.co/books?id=aA6y2rdDKXsC&printsec=frontcover&dq=Signal+Processing+for+Communications&hl=es-419&sa=X&ei=T47iUsTaF4-rkQfpyoCYBg&ved=0CCoQ6AEwAA>. [Accedido: 24 de Enero de 2014].
- [7] R. Gallager, "Random Processes and Noise" Op. cit., Capítulo 7.
- [8] R. Seebacher, "Union Bound on the Probability of Error" En: "Communication Systems", Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology, 2011.
- [9] E. Ferlin, V. Junior, "The Learning of Reconfigurable Computing in the Computer Engineering Program," Frontiers in Education Conference, 36th Annual , vol., no., pp.16,20, 27-31 Oct. 2006

- [10] M. Iqbal, U. Awan, S. Khan, "Reconfigurable computing technology used for modern scientific applications", Education Technology and Computer (ICETC), 2010 2nd International Conference on , vol.5, no., pp.V5-36,V5-41, 22-24 June 2010
- [11] K. Leijten, J. Meerbergen, "Embedded Reconfigurable Logic Core for DSP Applications". En: M. Glesner, P. Zipf, M. Renovell, "Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream", Springer Berlin Heidelberg, 2002.
- [12] S. Voros, K. Masselos, "System Level Design of Reconfigurable Systems-on-Chip", Springer US, 2005, ISBN: 978-0-387-26103-4 (Print) 978-0-387-26104-1 (Online)
- [13] M. Safadi, D. Ndzi, "Digital Hardware Choices For Software Radio (SDR) Baseband Implementation," Information and Communication Technologies, 2006. ICTTA '06. 2nd, vol.2, no., pp.2623,2628.
- [14] E. Boemo, "Estado del Arte de la Tecnología FPGA". Internet:
http://www.inti.gob.ar/electronicaeinformatica/instrumentacion/utic/publicaciones/cuadernilloUE/CT_Microelectronica17_FPGA.pdf
- [15] R. Pressman, "Ingeniería del Software, un Enfoque Práctico", Quinta edición, McGraw Hill, 2002.
- [16] K. Cho; D. Yoon, "On the general BER expression of one- and two-dimensional amplitude modulations," Communications, IEEE Transactions on , vol.50, no.7, pp.1074,1080, Jul 2002.

APÉNDICES

APÉNDICE A. VALORES DE DESVIACIÓN ESTÁNDAR (σ) CORRESPONDIENTES A DETERMINADOS VALORES DE RELACIÓN DE ENERGÍA DE BIT A DENSIDAD ESPECTRAL DE POTENCIA DE RUIDO (E_b/N_o)

E_b/N_o (dB)	16 QAM	64 QAM
	σ	σ
0	0.35439289	0.28867555
1	0.315852997	0.257282355
2	0.28150428	0.22930314
3	0.250890954	0.204366639
4	0.223606798	0.182141959
5	0.199289768	0.162334191
6	0.177617193	0.1446805
7	0.15830149	0.128946632
8	0.141086351	0.114923806
9	0.125743343	0.10242595
10	0.112068872	0.091287224
11	0.099881488	0.081359824
12	0.08901947	0.07251202
13	0.079338686	0.064626406
14	0.070710678	0.057598345
15	0.063020958	0.051334579
16	0.056167488	0.045751991
17	0.050059326	0.040776505
18	0.044615422	0.036342099
19	0.039763536	0.032389929
20	0.035439289	0.028867555

APÉNDICE B. VALORES DE BER CORRESPONDIENTES A DETERMINADOS VALORES DE RELACIÓN DE ENERGÍA DE BIT A DENSIDAD ESPECTRAL DE POTENCIA DE RUIDO (E_b/N_o)

B.1 Modulación 16 QAM con mapeo Gray y 32768 bits transmitidos

E_b/N_o	System Generator	Implementación	Simulink	Bertool
0	0.1412	0.141265869140625	0.1431	0.1409
1	0.11730	0.117309570312500	0.1202	0.1189
2	0.09774	0.097747802734375	0.1006	0.0977
3	0.07574	0.075744628906250	0.0793	0.0774
4	0.05725	0.057250976562500	0.059	0.0586
5	0.03930	0.039306640625000	0.0428	0.0418
6	0.02822	0.028228759765625	0.029	0.0278
7	0.017211	0.017211914062500	0.0175	0.0169
8	0.008605	0.008605957031250	0.0093	0.0092
9	0.0040283	0.004028320312500	0.0048	0.0043
10	0.0015258	0.001525878906250	0.0019	0.0017
11	0.000305175	0.000305175781250	0.00076	0.000564
12	0	0	0.0003	0.000138
13	0	0	0	0.0000242

B.2 Modulación 64 QAM con mapeo Gray y 32768 bits transmitidos

E_b/N_o	System Generator	Implementación	Simulink	Bertool
0	0.19952	0.1995239257812500	0.2001	0.1998
1	0.174896	0.1748962402343750	0.1775	0.1779
2	0.158081	0.1580810546875000	0.1565	0.1569
3	0.135070	0.1350708007812500	0.1361	0.1371
4	0.115509	0.1155090332031250	0.1183	0.1185
5	0.099761	0.0997619628906250	0.1006	0.1007
6	0.0830688	0.0830688476562500	0.0845	0.0838
7	0.0660705	0.0660705566406250	0.0696	0.0675
8	0.0529785	0.0529785156250000	0.0538	0.0523
9	0.0399780	0.0399780273437500	0.0398	0.0384
10	0.0256958	0.0256958007812500	0.0281	0.0265
11	0.016784	0.0167846679687500	0.0183	0.0168
12	0.009674	0.0096740722656250	0.0108	0.0097
13	0.004730	0.0047302246093750	0.0055	0.0049
14	0.0018005	0.0018005371093750	0.0024	0.0021
15	0.0007629	0.0007629394531250	0.00088	0.000772

B.3 Modulación 16 QAM y mapeo Binario con 32768 bits transmitidos

E_b/N_o	System Generator	Implementación	Simulink
0	0.1811	0.181121826	0.1844
1	0.1542	0.154296875	0.1576
2	0.1267	0.126708984	0.132
3	0.0993	0.099334717	0.1048
4	0.0753	0.075378418	0.0786
5	0.0537	0.053741455	0.0576
6	0.0354	0.035461426	0.0389
7	0.0214	0.021484375	0.0237
8	0.0122	0.012298584	0.0126
9	0.005462	0.005462646	0.0065
10	0.002044	0.002044678	0.0026

B.4 Modulación 64 QAM con mapeo Binario y 32768 bits transmitidos

E_b/N_o	System Generator	Implementación	Simulink
0	0.2594	0.259490967	0.2646
1	0.243	0.24307251	0.2452
2	0.2219	0.221954346	0.225
3	0.201	0.201049805	0.2035
4	0.1764	0.176422119	0.1811
5	0.1535	0.153533936	0.1575
6	0.1266	0.126678467	0.1323
7	0.0993	0.099334717	0.11
8	0.0778	0.077819824	0.0862
9	0.0597	0.059783936	0.0639
10	0.0417	0.041748047	0.0449
11	0.0272	0.027282715	0.0301
12	0.0156	0.015594482	0.0179
13	0.007568	0.007568359	0.0091
14	0.003173	0.003173828	0.0041
15	0.001098	0.001098633	0.0016

B.5 Modulación 16 QAM con mapeo Gray y 1048576 bits transmitidos

Eb/No	System Generator	Implementación	Simulink	Bertool
0	0.14064	0.1406431198120120000	0.1407	0.1409
1	0.11849	0.1184930801391600000	0.1188	0.1189
2	0.09793	0.0979356765747070000	0.0976	0.0977
3	0.07722	0.0772285461425781000	0.0772	0.0774
4	0.0582427	0.0582427978515625000	0.0586	0.0586
5	0.041803	0.0418033599853516000	0.0418	0.0418
6	0.028071	0.0280714035034180000	0.0277	0.0278
7	0.017255	0.0172557830810547000	0.0168	0.0169
8	0.0094547	0.0094547271728515600	0.0092	0.0092
9	0.0043821	0.0043821334838867200	0.0043	0.0043
10	0.0017461	0.0017461776733398400	0.0018	0.0017
11	0.00056552	0.0005655288696289060	0.000499	0.000564
12	0.00011825	0.0001182556152343750	0.000129	0.000138
13	0.0000257492	0.0000257492065429687	0.0000257	0.0000242
14	0.0000038146	0.0000038146972656250	0.000000953	0.00000276
15	0.000000953674	0.0000009536743164062	0	0.000000184

B.6 Modulación 64 QAM con mapeo Gray y 1048576 bits transmitidos

E _b /N _o	System Generator	Implementación	Simulink	Bertool
0	0.199398	0.1993980407714840000	0.1999	0.1998
1	0.177877	0.1778774261474610000	0.1776	0.1779
2	0.155647	0.1556472778320310000	0.1566	0.1569
3	0.13693	0.1369390487670900000	0.137	0.1371
4	0.11794	0.1179418563842770000	0.1187	0.1185
5	0.10051	0.1005191802978520000	0.1009	0.1007
6	0.08333	0.0833349227905273000	0.084	0.0838
7	0.067380	0.0673809051513672000	0.0676	0.0675
8	0.052328	0.0523281097412109000	0.0523	0.0523
9	0.0384187	0.0384187698364258000	0.0384	0.0384
10	0.0263414	0.0263414382934570000	0.0265	0.0265
11	0.0165519	0.0165519714355469000	0.0167	0.0168
12	0.00962352	0.0096235275268554700	0.0096	0.0097
13	0.0049419	0.0049419403076171900	0.005	0.0049
14	0.0021438	0.0021438598632812500	0.0022	0.0021
15	0.00074577	0.0007457733154296870	0.000755	0.000772
16	0.000207901	0.0002079010009765620	0.000198	0.000217
17	0.0000400543	0.0000400543212890625	0.0000524	0.0000449