



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA-SISTEMAS ELECTRÓNICOS

DESARROLLO DE UN MODULADOR ADAPTATIVO PARA UN SISTEMA DE TELECOMUNICACIÓN A BORDO DE UN
MICROSATÉLITE

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
GERALDO SALAZAR DIAZ

TUTOR (ES) PRINCIPAL(ES)
DR. SAÚL DE LA ROSA NIEVES FACULTAD DE INGENIERÍA

CIUDAD DE MÉXICO, SEPTIEMBRE 2017



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Jesús Savage Carmona
Secretario: Dr. Jorge Rodríguez Cuevas
Vocal: Dr. Saúl de la Rosa Nieves
1 er. Suplente: Dr. José Ismael Martínez López
2 do. Suplente: Dr. Juan Mario Peña Cabrera

Lugar o lugares donde se realizó la tesis:
Laboratorio de Instrumentación Electrónica de Sistemas Espaciales (LIESE)
Universidad nacional Autónoma de México (UNAM)
Ciudad Universitaria, Ciudad de México

TUTOR DE TESIS:

Dr. Saúl de la Rosa Nieves

FIRMA

Resumen

En este trabajo de tesis se presenta el desarrollo de un modulador adaptativo para un sistema de telecomunicaciones de a bordo de un microsatélite de órbita baja, con el uso de técnicas de Radio Definido por Software e implementado en una plataforma FPGA. Para definir los requerimientos de diseño se realiza una simulación del enlace de comunicaciones usando el software STK, y partiendo de estos requerimientos se realiza el diseño e implementación del sistema en la plataforma FPGA. Para este proyecto se utilizó la tarjeta de desarrollo Basys 3 de Xilinx, la cual soporta el FPGA xc7a35tcp236-1. Para la síntesis e implementación del diseño se utilizó el software Vivado HL Webpack Edition v2016.4 que provee Xilinx. Debido a los requerimientos especiales de los aparatos electrónicos que trabajan en el espacio, se investiga de manera documental los efectos de la radiación espacial en los módulos sensibles de los FPGA SRAM, identificando los mecanismos de falla en sus bloques internos debido a las partículas de alta energía que producen los Efectos de Eventos Individuales (SEE, por sus siglas en inglés). Se analizan las posibles técnicas de tolerancia a fallas que pueden permitir incrementar la confiabilidad en el sistema, y finalmente se implementa el modulador con un nivel de confiabilidad que posibilite su uso en aplicaciones espaciales.

Palabras clave: Modulación adaptativa, aplicaciones espaciales, microsatélite, FPGA

Abstract

This thesis presents the development of an adaptive modulator for the telecommunication system of a low orbit satellite, using Defined Software Radio techniques and implemented in an FPGA platform. To define the requirements of the design, a communication link simulation is performed using the STK software, and based on these requirements, the design and implementation of the system in the FPGA platform is performed. Due to the special requirements of electronic devices working in space, the effects of space radiation on the sensitive modules of the SRAM FPGA are investigated in a documentary manner, identifying the fault mechanisms in their internal blocks because of Single Events Effects (SEE) produced by high energy particles. We analyze possible fault tolerance techniques that can increase reliability in the system and finally implement the modulator with a level of reliability that allows its use in space applications.

Key words: Adaptive modulation, space applications, microsatellite, FPGA

Agradecimientos

A mis padres, gracias a su esfuerzo, sacrificio y guía hoy soy la persona que soy, y todos mis logros son posibles solamente porque ellos han estado ahí. Son mi ejemplo a seguir en todos los aspectos de la vida.

A mi hermana que me impulsó a iniciar la maestría, por todo su apoyo a lo largo de estos dos años de trabajo, gracias por guiarme en mis decisiones y por mostrarme el camino a seguir.

A Yazmin Dillarza Andrade, mi compañera durante casi la totalidad de este trabajo, por su comprensión y apoyo en los momentos difíciles, y por abrirme las puertas de su casa y de su corazón después de ver mis ángeles y mis demonios. Sin ella este trabajo no sería posible.

Al doctor Saúl de la Rosa Nieves, quien fue el tutor principal de este trabajo, y cuya asesoría y guía fueron pilar fundamental en la realización de la tesis con la calidad requerida. Gracias por aceptarme como estudiante y por el apoyo tanto en el aspecto profesional como personal.

A mis compañeros de maestría, Iván, Arturo, Antonio, Tonatiuh, Cesar, que de compañeros de aula se convirtieron en amigos, gracias a ellos trabajar no fue un trabajo.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca (CVU/Becario) 708878/585632, la cual me permitió completar satisfactoriamente los estudios de posgrado y la realización de este trabajo

Índice

Lista de acrónimos	vii
Lista de símbolos	ix
Lista de unidades	ix
Lista de figuras	x
Lista de tablas	xii
Introducción	1
Objetivo General	1
Objetivos Específicos.....	1
Justificación	2
Metodología.....	2
Estructura de la tesis.....	2
Capítulo 1. Marco teórico	4
1.1 Características del canal de transmisión de un satélite LEO.	4
1.2 Esquemas de modulación más usados en aplicaciones espaciales	5
1.3 Comparación entre las modulaciones más usadas en aplicaciones espaciales	6
1.3.1 Comparación entre QPSK y DQPSK ($M = 4$)	6
1.3.2 Comparación entre 8-PSK y 8-QAM ($M = 8$).....	7
1.3.3 Comparación entre 16-PSK y 16-QAM ($M=16$)	8
1.4 Esquemas de modulación adaptativos	9
1.5 Estado del arte de los sistemas de modulación adaptativa para aplicaciones espaciales.....	10
1.6 Base teórica de las modulaciones seleccionadas para la modelación adaptativa	11
1.6.1 Modulación QPSK.....	11
1.6.2 Modulación 8-PSK.....	12
1.6.3 Modulación 16-QAM.....	16
1.7 Conclusiones del Capítulo 1	18
Capítulo 2. Determinación de los requerimientos para el diseño del sistema de comunicaciones.....	19
2.1 Simulación del enlace de comunicaciones en STK.....	19

2.1.1 Elementos de mecánica Orbital.....	19
2.1.2 Bandas de Operación.....	21
2.1.3 Antena de estación terrena.....	21
2.1.4 Ancho de banda Permitido.....	21
2.1.5 Antena del Satélite.....	22
2.2 Resultados de la simulación en STK.....	24
2.3 Metodología de conmutación del sistema	26
2.4 Estimación de las demoras de transmisión.....	29
2.5 Estimación del BER en el canal de transmisión.....	31
2.6 Comparación de las velocidades obtenidas usando modulación adaptativa vs modulación fija.....	33
2.7 Resumen del Capítulo 2.....	33
Capítulo 3. Desarrollo del modulador adaptativo.....	35
3.1 Lenguaje VHDL, características principales.....	35
3.2 Selección de la tecnología FPGA a utilizar.....	35
3.3 FPGAs SRAM.....	36
3.4 Herramienta Vivado Design Suite HL WebPack.....	37
3.5 Metodología de diseño jerárquico en VHDL.....	37
3.6 Bloque Generador de Frecuencias.....	39
3.6.1 Componente sel_salida_pll.....	40
3.6.2 Componente PLL.....	41
3.7 Bloque QPSKModulator	42
3.7.1 Componente FDD.....	43
3.7.2 Componente divisor_frec_x2.....	44
3.7.3 Componente control_bits.....	44
3.7.4 Componente conversión_NRZ.....	45
3.8 Componente PSK_8_Modulator.....	45
3.8.1 Componente desplazador_3_bits.....	46
3.8.2 Componente convertidor 2 a 4.....	48
3.8.3 Componente negador.....	49
3.9 Componente QAM16modulator	50
3.9.1 Componente desplazador_4_bits.....	50
3.9.2 Componente divisor_frec_x4.....	51

3.9.3 Componente convertidor_2_4_1.....	52
3.10 Componente MUX_sal.....	53
3.11 Componente NCO1	53
3.12 Componente multiplier.....	55
3.13 Componente adder	55
3.14 Resumen Capítulo 3.....	56
Capítulo 4. Aplicación de técnicas de tolerancia a fallas.....	57
4.3 Caracterización de los ambientes radioactivos. Daños que estos producen en los FPGAs SRAM.....	57
4.3.1 Efecto SEU.....	58
4.4 Métodos de mitigación de los efectos de la radiación en FPGAs	59
4.4.1 Técnicas orientadas al proceso de fabricación.....	59
4.4.2 Técnicas orientadas al diseño.....	59
4.4.3 Técnicas de recuperación	62
4.5 Herramientas que permiten la implementación de tolerancia a fallas de forma automática	63
4.5.1 Xilinx TMR Tool	63
4.5.2 Synopsis Synplify Premier	64
4.5.2 ByuEdifTool.....	64
4.6 Ejemplo de utilización de la herramienta ByuEdifTool.....	65
4.6.1 Descripción de la herramienta.....	66
4.7 Aplicación de TMR al modulador adaptativo usando la herramienta ByuEdifTool ..	70
4.8 Conclusiones del capítulo	72
Capítulo 5. Pruebas de validación y resultados.....	73
5.1 Validación del modulador QPSK.....	73
5.2 Validación del modulador 8PSK.....	76
5.3 Validación del modulador 16QAM	78
5.3 Conclusiones del capítulo	81
Capítulo 6 Conclusiones y trabajo a futuro	82
Conclusiones	82
Trabajo a futuro	82
Bibliografía	83

Lista de acrónimos

LEO	Low Earth Orbit
FPGA	Field Programmable Gate Array
COTS	Commercial To-the-Shelf
SDR	Software Defined Radio
ACM	Adaptive Coding and Modulation
SRAM	Static Random-Access Memory
SEE	Single Event Effects
RF	Radio Frequencies
TWTA	Traveling-Wave Tube Amplifier
SSPA	Solid State Power Amplifier
AM	Amplitude Modulation
PM	Phase Modulation
QAM	Quadrature Amplitude Modulation
PSK	Phase Shift Keying
HPA	High Power Amplifier
OBO	Output Back-Off
ASK	Amplitude Shift Keying
DQPSK	Differential Quadrature Phase Shift Keying
STK	System Toolkit
BER	Bit Error Rate
AWGN	Additive White Gaussian Noise
SNR	Signal to Noise Ratio
NRZ	No Return to Zero
UNAM	Universidad Nacional Autónoma de México
QPSK	Quadrature Phase Shift Keying
VHDL	Very High Speed Integrated Circuits Hardware Description Language
SEU	Single Event Upset

SEL	Single Event Latch up
SEFI	Single Event Functional Interrupt
SET	Single Event Transient
TID	Total Ionization Dosis
CMOS	Complementary Metal Oxide Semiconductor
TMR	Triple Modular Redundancy
EDAC	Error Detection and Correction Code
ECC	Error Correction Code
FEC	Forward Error Correction
DWC	Duplication With Compare

Lista de símbolos

ω_c	omega
t	tiempo
φ	varphi
λ	lambda

Lista de unidades

kg	kilogramo
dB	decibel
km	kilómetro
Hz	Hertz
K	Constante de Kelvin
W	Watts
dBi	decibeles con respecto a un transmisor isotrópico
s	segundo
bps	bits por segundo

Lista de figuras

Figura 1 Comparación del BER entre DPSK y QPSK	7
Figura 2 Comparación del BER entre 8-PSK y 8-QAM Adaptada de [32].....	8
Figura 3 Comparación entre 16-QAM y 16-8PSK a) Para un canal Rayleigh b) Para un canal AWGN Adaptada de [32].....	8
Figura 4 ACM Punto a Punto	9
Figura 5 Esquema del modulador QPSK.....	11
Figura 6 Constelación de la modulación QPSK	12
Figura 7 Esquema del modulador 8PSK	13
Figura 8 Determinación de los coeficientes de amplitud para un modulador 8 PSK	13
Figura 9 Posibles combinaciones de la modulación 8-PSK	14
Figura 10 Constelación 8PSK	15
Figura 11 Modulador 16-QAM	16
Figura 12 Triángulo obtenido de unir los puntos de la constelación 16-QAM	16
Figura 13 Inserción de la estación terrena en STK.....	20
Figura 14 Inserción del satélite en STK	20
Figura 15 Configuración de la antena de la estación terrena en STK	22
Figura 16 Configuración de la antena del satélite	23
Figura 17 Vista 2D de la órbita del satélite	23
Figura 18 Comportamiento del BER para un evento de línea de vista.....	25
Figura 19 Histograma de duración de los eventos de línea de vista.....	26
Figura 20 Comportamiento del BER para un evento de línea de vista.....	27
Figura 21 Histograma del tiempo que transcurre antes de que esté listo el enlace para la conmutación	28
Figura 22 Intervalo de aplicación de la conmutación entre modulaciones.....	28
Figura 23 Histograma de distancias durante un año.....	29
Figura 24 Histograma de distancias para eventos de duración superior a los 420 segundos	30
Figura 25 Modelo del sistema PSAM Adaptado de [17].....	31
Figura 26 Determinación del tiempo entre estimaciones Adaptado de [17]	32
Figura 27 Comportamiento de la velocidad de transmisión durante un evento de línea de vista.....	33
Figura 28 Top Level del sistema digital	38
Figura 29 Segundo nivel de abstracción.....	38
Figura 30 Top Level del bloque generador de frecuencias	40
Figura 31 Estructura del bloque sel_salida_pll	41
Figura 32 Top Level del componente PLL.....	42
Figura 33 Top Level del componente QPSKmodulator	42
Figura 55 Alteración aplicada a la lógica combinacional y secuencial Adaptado de [3]	58
Figura 56 Efecto SEU en una unidad de memoria Adaptado de [3]	59
Figura 57 a) Redundancia por tiempo b) Redundancia por hardware para lógica combinacional c) Redundancia por hardware para lógica Adaptado de [4]	60
Figura 58 TMR en tiempo Adaptado de [3]	61
Figura 59 TMR en hardware Adaptado de [4]	61
Figura 60 Votador para sincronización Adaptado de [20]	62
Figura 61 Gráfica de disponibilidad para un sistema simple y uno redundante.....	65
Figura 62 Flujograma de la herramienta JEdifTool.....	67

Figura 63 Exportar archivo .edf desde Vivado.....	68
Figura 64 Resultado de la síntesis del diseño simple	69
Figura 65 Resultado de la síntesis del diseño replicado	70
Figura 66 Reporte de utilización de Vivado por cantidad de elementos ocupados. Diseño sin redundancia.....	70
Figura 67 Reporte de utilización de Vivado dado en por ciento. Diseño sin redundancia.....	71
Figura 68 Reporte de utilización de Vivado por cantidad de elementos ocupados. Diseño con TMR	71
Figura 69 Reporte de utilización de Vivado dado en por ciento. Diseño con TMR.....	71
Figura 70 Esquema a bloques del demodulador QPSK.....	73
Figura 71 Circuito digital implementado en Vivado del demodulador QPSK.....	74
Figura 72 Validación del modulador QPSK.....	74
Figura 73 Validación del demodulador QPSK.....	75
Figura 74 Esquema a bloques del demodulador 8PSK.....	76
Figura 75 Circuito digital implementado en Vivado del demodulador 8PSK.....	77
Figura 76 Validación del modulador 8PSK.....	77
Figura 77 Validación del demodulador 8PSK.....	78
Figura 78 Esquema a bloques del demodulador 16QAM.....	79
Figura 79 Circuito digital implementado en Vivado del demodulador 16QAM.....	79
Figura 80 Validación del modulador 16QAM.....	80
Figura 81 Validación del demodulador 16QAM.....	80

Lista de tablas

Tabla 1 Resumen de la revisión bibliográfica	11
Tabla 2 Tabla de la verdad del bloque Convertidor de 2 a 4 niveles.....	14
Tabla 3 Salida del modulador 8-PSK dependiendo de los valores de I, Q y C	15
Tabla 4 Tabla de la verdad del conversor de 2 a 4 niveles para el modulador 16QAM.....	17
Tabla 5 Salida del sumador dependiendo de los bits I, I', Q y Q'	17
Tabla 6 Bandas de Operación.....	21
Tabla 7 Antena de la estación terrena.....	21
Tabla 8 Antena del satélite	22
Tabla 9 . Demoras de transmisión	30
Tabla 10 Comparación entre tecnologías de fabricación de FPGAs	36
Tabla 11 Tabla de la verdad del bloque sel_sal_pll.....	40
Tabla 12 Lógica de funcionamiento del bloque conversión_NRZ.....	45
Tabla 13 Lógica de funcionamiento del bloque conversión 2 a 4	48
Tabla 14 Correspondencia entre las salidas de la red de flip-flops y el bloque desplazador_4_bits.....	51
Tabla 15 Lógica de funcionamiento del bloque conversión_2_4_1	52
Tabla 16 Lógica de funcionamiento del bloque MUX_sal.....	53

Introducción

Los microsátélites (satélites de 100 a 10 kg) de Órbita Baja (LEO, por sus siglas en inglés) han adquirido gran popularidad en el campo de las comunicaciones debido a sus ventajas sobre los satélites geoestacionarios, entre las que podemos mencionar que son generalmente de menor tamaño, el costo de desarrollo y puesta en órbita es una fracción de los costos de un satélite geoestacionario y se puede alcanzar cobertura global con varios satélites. Tienen varias áreas de aplicación como observación terrestre, monitoreo de desastres y comunicaciones inalámbricas. [1]

Con el fin de obtener un mejor aprovechamiento de recursos críticos como el ancho de banda y la potencia, y debido a la dinámica que presentan los micros satélites en órbita baja, es muy atractivo el diseño de sistemas de telecomunicaciones para microsátélites basados en esquemas de modulación adaptativa, debido a las ventajas que ofrecen estos esquemas con respecto al uso de una modulación fija. [2]

En la actualidad el uso de los Campos de Arreglos de Compuertas Programables, (FPGAs, por sus siglas en inglés) de tipo comercial (COTS, por sus siglas en inglés) para el desarrollo de sistemas electrónicos espaciales ha cobrado un gran auge debido al enorme potencial que estos dispositivos presentan como son: gran velocidad de procesamiento; capacidad de reconfiguración; flexibilidad y un costo mucho más bajo que sus análogos de calidad militar o espacial.

Una de las aplicaciones de los FPGA que ofrecen una gran perspectiva en el desarrollo de sistemas espaciales es la aplicación del concepto de Radio Definido por Software (SDR, por sus siglas en inglés) para implementar los sistemas que realizan la comunicación entre un microsátélite y la estación terrena. Un radio definido por software se puede definir como “un sistema de radio en el cual alguna o todas sus funciones de capa físicas están definidas por software”. Al definir el comportamiento del sistema de radio en software, se elimina la necesidad de alteraciones en el hardware durante una actualización de la tecnología. [3]

El principal problema para el diseño de sistemas espaciales en base a los FPGA COTS, es que su tolerancia a la radiación es mucho menor con respecto a los dispositivos de calidad espacial o militar. Por lo anterior para poder realizar diseños aceptables para misiones espaciales en base a FPGA COTS, es necesaria la aplicación de técnicas de tolerancia a fallas para obtener un producto de mayor confiabilidad. [4]

Objetivo General

Desarrollo de un modulador adaptativo para un sistema de telecomunicaciones de a bordo de un microsátélite de órbita baja, sistema de radio con modulación adaptativa con el uso de técnicas de SDR implementado en FPGAs,

Objetivos Específicos

- a) Diseño de una metodología de conmutación de modulaciones para el modulador adaptativo
- b) Diseño de la arquitectura del modulador adaptativo para su implementación en un FPGA.
- c) Selección de una herramienta para implementar técnicas de tolerancia a fallas para el sistema desarrollado para que en trabajo a futuro se pueda asegurar que el sistema cuente niveles de confiabilidad aceptables para su aplicación en aparatos espaciales.

Justificación

Las características de los canales de transmisión usados por los microsátélites, de conjunto con el creciente aumento de la demanda en la cantidad y calidad de servicios que prestan estos, hace necesaria la aplicación de técnicas que permitan aprovechar al máximo el ancho de banda disponible, ajustándose a las limitaciones de potencia presentes en estos dispositivos. Una vía con gran perspectiva para aumentar la capacidad y flexibilidad de un sistema de radio de a bordo de un microsátélite es el uso de métodos de codificación y modulación adaptativos, para aprovechar al máximo las capacidades del canal de transmisión.

El desarrollo de SDR basados en FPGA COTS es una opción muy atractiva gracias a su bajo costo, flexibilidad y capacidad de procesamiento, pero se requiere la aplicación de técnicas de tolerancia a fallas para alcanzar niveles de confiabilidad aceptables para su aplicación en aparatos espaciales.

Metodología

- a) Estudio de los requerimientos de ancho de banda y potencia para el canal de comunicaciones de los micro satélites LEO.
- b) Revisión del marco teórico para el desarrollo de sistemas de modulación adaptativos (ACM, por sus siglas en inglés).
- c) Investigación del estado del arte de sistemas ACM a bordo de satélites.
- d) Desarrollo del modulador adaptativo
- e) Análisis de los efectos de la radiación espacial en los módulos sensibles de los FPGA SRAM, identificando los mecanismos de falla en sus bloques internos debido a las partículas de alta energía que producen los Efectos de Eventos Individuales (SEE, por sus siglas en inglés).
- f) Análisis de las posibles técnicas de tolerancia a fallas que pueden permitir incrementar la confiabilidad de un diseño de un sistema de radio SDR de a bordo de un aparato espacial;
- g) Selección de una herramienta para implementar técnicas de tolerancia a fallas.
- h) Comprobación del desempeño del sistema mediante simulaciones.

Estructura de la tesis

Esta tesis consta de 5 capítulos

En el Capítulo 1 – Marco teórico, se presentan los fundamentos teóricos para el diseño del sistema de comunicaciones, así como un estudio de bibliografía actualizada acerca de la implementación de sistemas de modulación adaptativos para aplicaciones espaciales.

Se presentan los fundamentos teóricos que se usarán para la implementación del sistema. Se realiza un estudio donde se exponen los esquemas de modulación más usados en aplicaciones espaciales, partiendo del cual se realiza una comparación entre los esquemas de modulación identificados para seleccionar los que mejor se ajusten a los requerimientos de nuestro sistema Posteriormente se realiza el estudio del estado del arte de la modulación adaptativa en aplicaciones espaciales, haciendo especial énfasis en aquellas que se basan en FPGAs. Finalmente se presenta la base teórica para la construcción de los moduladores seleccionados para ser usados en nuestro sistema.

En el Capítulo 2 – Determinación de los requerimientos para el diseño del sistema de comunicaciones, se realiza un modelo del canal de comunicaciones mediante simulación que nos permite determinar las principales afectaciones que sufrirá nuestro sistema, las

cuáles definirán las técnicas y parámetros de diseño que serán utilizados posteriormente en el proceso de diseño e implementación. Se obtiene una aproximación del comportamiento de los parámetros de transmisión, específicamente el comportamiento de la tasa de bit erróneo, (BER, por sus siglas en inglés). Para esto se usarán los requerimientos que se tienen del diseño de la misión.

En el Capítulo 3 – Desarrollo del modulador adaptativo, se describe el proceso de diseño, descripción e implementación del modulador adaptativo, auxiliándose de la base teórica y los requerimientos de diseño establecidos en el Capítulo 2.

En el Capítulo 4 – Aplicación de técnicas de tolerancia a fallas, se realiza un análisis del efecto que tienen las partículas de alta energía presentes en el ambiente espacial sobre los bloques sensibles del FPGA SRAM, y se estudia de forma documental las técnicas de tolerancia a fallas más utilizadas en sistemas espaciales, con el objetivo de brindar protección contra efectos indeseables como los SEU. Se realiza un análisis comparativo entre varias herramientas que permiten la implementación de sistemas tolerantes a fallas de forma automática, y se selecciona una de ellas para realizar un ejemplo demostrativo de su uso.

En el Capítulo 5 – Pruebas de validación y resultados, se muestran las pruebas realizadas para validar el correcto funcionamiento del sistema

En el Capítulo 6 – Conclusiones y Trabajo a futuro, se presentan las conclusiones y las propuestas de trabajo a futuro.

Capítulo 1. Marco teórico

En este capítulo se presentan los fundamentos teóricos que se usarán para la implementación del sistema. Se realiza un estudio donde se exponen los esquemas de modulación más usados en aplicaciones espaciales, partiendo del cual se realiza una comparación entre los esquemas de modulación identificados para seleccionar los que mejor se ajusten a los requerimientos de nuestro sistema. Posteriormente se realiza el estudio del estado del arte de la modulación adaptativa en aplicaciones espaciales, haciendo especial énfasis en aquellas que se basan en FPGAs. Finalmente se presenta la base teórica para la construcción de los moduladores seleccionados para ser usados en nuestro sistema.

1.1 Características del canal de transmisión de un satélite LEO.

El medio de propagación que conecta el transmisor con el receptor se denomina canal de transmisión. [5] En general este puede ser pares telefónicos, cable coaxial, fibra óptica, y en el caso de Radio Frecuencia (RF), guías de onda, la atmósfera o el espacio libre.

Las principales fuentes de ruido y pérdida de la potencia de la señal transmitida en un canal de transmisión para un enlace satelital son:

1. Pérdidas de potencia de la señal transmitida [5]

- a. **Pérdida por limitación en ancho de banda:** Provocada por el filtrado que se realiza en el transmisor para garantizar que la señal transmitida no excede un ancho de banda especificado.
- b. **Pérdida por eficiencia de la antena:** Las antenas son transductores que convierten señales electrónicas en campos electromagnéticos, y viceversa. También tienen como objetivo concentrar la energía de la señal electromagnética en una dirección deseada. La eficiencia de la antena se describe como la relación entre el área efectiva de la antena y su área física, con valores típicos entre 50 y 80 %. Una reducción en la eficiencia de la antena se traduce en una disminución de la potencia de la señal recibida.
- c. **Pérdida por modulación:** En el receptor, solo la potencia relacionada con la señal de información es relevante, ya que el desempeño del sistema ante errores está en función de la energía transmitida por símbolo. Cualquier potencia usada para transmitir la portadora, en vez de la señal modulada, se conoce como pérdida por modulación.
- d. **Pérdida por propagación en el espacio libre:** La densidad de potencia de una onda electromagnética en un punto dado es inversamente proporcional a la distancia entre el transmisor y dicho punto, mientras mayor es la distancia menor es la densidad de potencia. En el caso de los sistemas de comunicaciones espaciales, esta componente de pérdida es la mayor de todas.

2. Fuentes de ruido o interferencia [5]

- a. **Interferencia Inter Símbolos:** Este fenómeno tiene lugar en los procesos de filtrado en el canal de transmisión, y consiste en la superposición de los pulsos transmitidos, ya que la energía del símbolo no se encuentra concentrada en el intervalo correspondiente al mismo, influyendo en el proceso de decisión del símbolo adyacente.

- b. **Interferencia de canal adyacente:** Se produce cuando frecuencias provenientes de otros canales añaden energía adicional al canal de interés, y tiene lugar cuando la separación entre canales no es la adecuada.
- c. **Ruido del receptor:** Este es el ruido térmico presente en el receptor.

3. Fuentes combinadas de ruido o interferencia y pérdida de potencia [5]

- a. **Ruido de fase en el oscilador local:** Cuando se usa un oscilador local para el mezclado de las señales, fluctuaciones de fase del mismo añaden ruido de fase a la señal transmitida. Cuando se usa como señal de referencia en el receptor, estos cambios de fase producen degradación de la detección y por tanto pérdida de la señal transmitida.
- b. **Ruido y pérdidas provocadas por la atmósfera:** Las partículas presentes en la atmósfera son responsables de la absorción de energía que produce una pérdida de potencia de la señal transmitida. Esta absorción varía dependiendo de la frecuencia de la señal transmitida, ya que ciertas partículas tienen un mayor índice de absorción a ciertas frecuencias. Al mismo tiempo, estas partículas que absorben energía también emiten energía hacia el exterior, y la porción de energía emitida que se encuentra en la frecuencia de la señal transmitida afecta a la misma en forma de ruido. La causa primaria de absorción de energía por parte de la atmósfera y la consiguiente emisión de ruido es la lluvia.

1.2 Esquemas de modulación más usados en aplicaciones espaciales

Los sistemas de comunicación digital son aquellos que codifican la información a transmitir usando valores discretos [5]. Los mismos operan en presencia de pérdidas por trayectoria y desvanecimiento inducido por la atmósfera. Para mantener suficiente potencia recibida en el destino, se requiere el uso de un dispositivo para generar suficiente potencia de transmisión, ejemplos de los cuales son los amplificadores de tubo de ondas de desplazamiento (TWTA, por sus siglas en inglés) y los amplificadores de potencia de estado sólido (SSPA, por sus siglas en inglés) operados en modo no lineal para maximizar la eficiencia de la conversión.

Lamentablemente, este requisito introduce distorsión en la señal a transmitir, esta distorsión puede ser en la amplitud de la señal, y se conoce como modulación de amplitud-amplitud (AM-AM, por sus siglas en inglés) y puede ser también en la fase de la señal, conocida como modulación de amplitud-modulación de fase (AM-PM, por sus siglas en inglés). Esta fue la causa por la cual las modulaciones que transmiten información a través de su amplitud, requiriendo una característica de amplificación lineal (por ejemplo, modulación de amplitud en cuadratura (QAM, por sus siglas en inglés)), no fueron muy populares en el inicio de la era espacial, siendo más usadas las modulaciones en fase M-PSK [6].

El crecimiento en la demanda de cantidad y calidad de los servicios que brindan los micro y nano satélites, ha traído consigo numerosas investigaciones encaminadas a la utilización de técnicas de modulación que alcancen mejores prestaciones en términos de utilización del ancho de banda y mayores velocidades de transmisión. [7]

Ejemplos de estos trabajos los podemos encontrar en [7], [8], [9], donde el objetivo principal es proponer técnicas que permitan disminuir la severa distorsión no lineal que introduce el amplificador de alta potencia (HPA, por sus siglas en inglés) de a bordo. Entre las propuestas más relevantes podemos encontrar:

- Utilizar el HPA en un punto de operación cercano a la saturación, sin llegar a la misma. La relación logarítmica entre la potencia de saturación y la potencia promedio de salida del amplificador se conoce como “output back-off” (OBO, por sus siglas en inglés) : $OBO(dB) = 10\log(P_{sat}/P_{out})$. La idea es usar un punto de operación con un OBO muy alto, sin llegar a saturación. Aunque el uso de este punto de operación no alcanza la máxima eficiencia del amplificador, se puede obtener un equilibrio entre la no-linealidad y la eficiencia.
- Pre-distorsión digital, la cual consiste en aplicar un proceso no-lineal inverso a la característica de transferencia del HPA, obteniendo así un proceso lineal. [9], [10]

Estos trabajos han abierto las puertas a la utilización de esquemas de modulación como M-QAM y M-ASK en aplicaciones espaciales, de manera que se puedan mantener niveles de eficiencia muy altos y aun así aprovechar las ventajas que ofrecen los esquemas de modulación de amplitud variable como son altas velocidades de transmisión y aprovechamiento del ancho de banda disponible.

A manera de conclusión, los esquemas de modulación más populares en aplicaciones espaciales son las modulaciones M-PSK y M-QAM, por tanto, basaremos nuestro análisis en este tipo de modulaciones.

1.3 Comparación entre las modulaciones más usadas en aplicaciones espaciales

Para seleccionar los esquemas de modulación a usar se realizó un estudio bibliográfico para encontrar elementos comparativos entre los distintos esquemas de modulación disponibles, que cumplieran con las especificaciones de diseño del sistema.

Se determinó realizar las comparaciones para modulaciones de igual orden, específicamente se compararon modulaciones usadas en aplicaciones espaciales donde $M = 4$, $M = 8$ y $M = 16$, escogiendo una en cada caso para la implementación del sistema.

1.3.1 Comparación entre QPSK y DQPSK ($M = 4$)

La modulación QPSK, cuyo acrónimo proviene del inglés “Quadrature Phase Shift Keying” que significa Modulación por Cambio de Fase en Cuadratura, transmite la información modificando la fase de la señal portadora, existiendo 4 posibles valores de fase a tomar, cada uno separado 90 grados del otro.

En el caso de la modulación DQPSK, su acrónimo proviene del inglés “Differential Quadrature Phase Shift Keying”, que significa Modulación por Cambio de Fase en Cuadratura Diferencial, la información se transmite mediante la diferencia de fase entre símbolos adyacentes, por lo que se considera un tipo de modulación no coherente, donde no es necesaria información sobre la portadora en el receptor.

Para realizar la comparación del desempeño obtenido con estas modulaciones se realizó una simulación comparativa entre ambas modulaciones usando el software STK (System Toolkit), en el punto 2.1 se describen las capacidades de este software. En la simulación se mantuvieron fijos los parámetros de transmisión como tipo de antena, potencia de transmisión y características del canal, los cuáles se analizan con más profundidad en el punto 2.1. Usando estos parámetros se realizó la comparación entre ambas modulaciones, obteniéndose los resultados mostrados en la Figura 1:

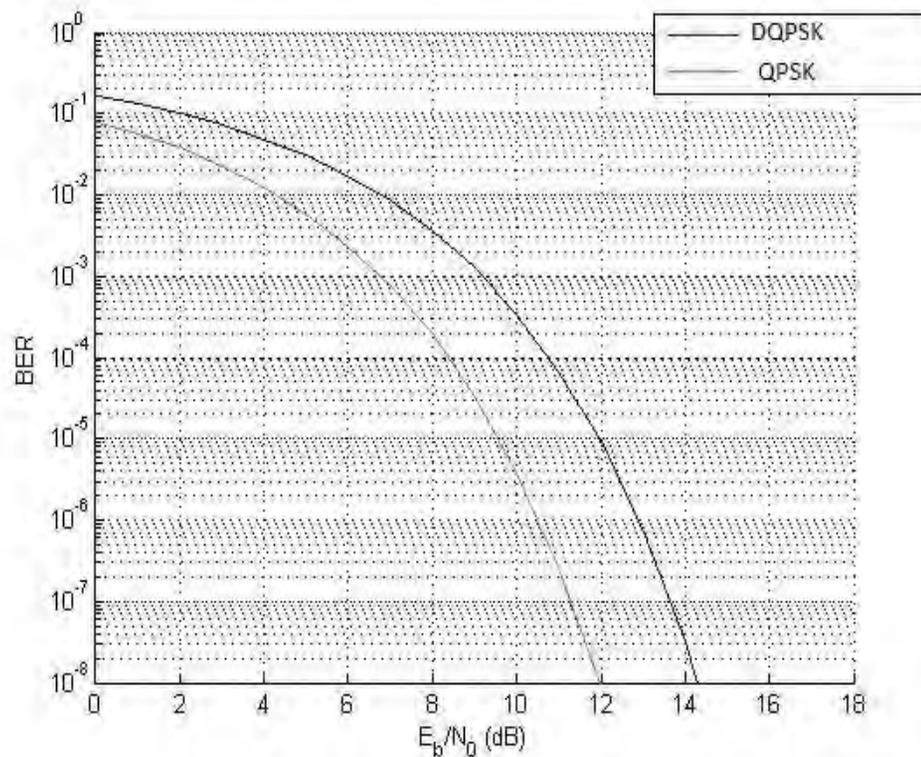


Figura 1 Comparación del BER entre DPSK y QPSK

Se puede apreciar que QPSK tiene un mejor desempeño en cuanto al BER en comparación con DQPSK, razón por la cual seleccionamos QPSK para ser usada en nuestro diseño.

En [32] se realiza un análisis comparativo entre las modulaciones M-PSK y M-QAM, para $M = 8$ y 16 , el cual tomamos como referencia para la selección entre este tipo de modulaciones.

1.3.2 Comparación entre 8-PSK y 8-QAM ($M = 8$)

La modulación 8-PSK, cuyo acrónimo proviene del inglés Phase Shift Keying, transmite información mediante la variación de fase de la portadora, manteniendo la amplitud constante; el valor 8 proviene de las ocho fases posibles a obtener, ya que los bits de información se agrupan en bloques de tres bits, generando ocho combinaciones posibles, a cada una de las cuales se le asigna una fase específica.

En el caso de 8 QAM, cuyo acrónimo proviene del inglés Quadrature Amplitude Modulation, los bits de entrada se agrupan en bloques de tres bits, pero a diferencia de 8-PSK la amplitud no se mantiene constante. En este caso se tienen dos posibles amplitudes y cuatro posibles fases.

En la Figura 2 se muestra una comparación de desempeño en cuanto al comportamiento del BER para las modulaciones 8-PSK y 8-QAM, bajo los efectos de un canal con ruido AWGN y RAYLEIGH. El análisis de la gráfica muestra que, para un mismo valor de E_b/N_0 , la modulación 8-PSK tiene un mejor valor de BER.

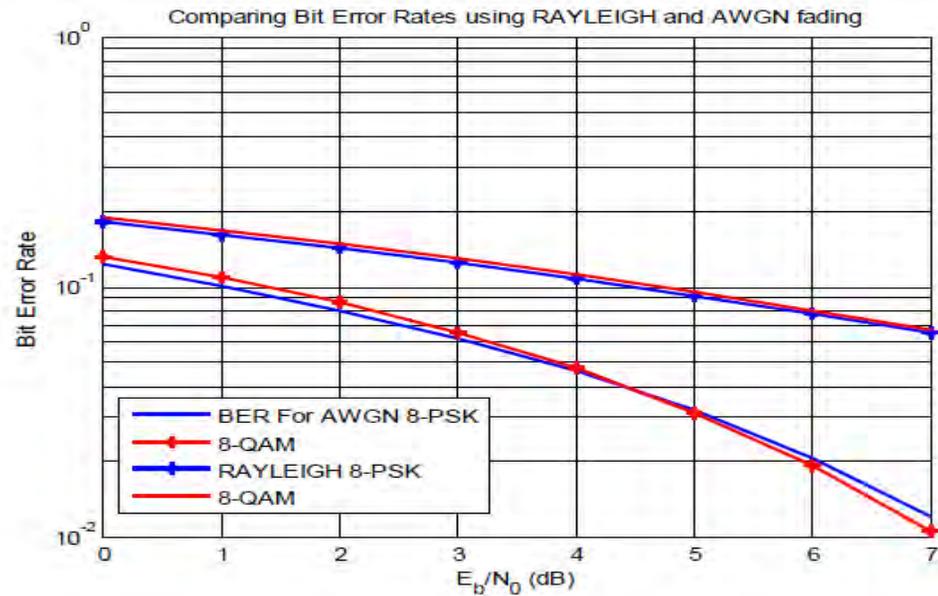


Figura 2 Comparación del BER entre 8-PSK y 8-QAM Adaptada de [32]

1.3.3 Comparación entre 16-PSK y 16-QAM (M=16)

La modulación 16-PSK, al igual que la modulación 8-PSK, mantiene la amplitud de la señal constante, pero en este caso se tienen 16 posibles fases, ya que los bits de entrada se agrupan en bloques de cuatro bits.

Asimismo, la modulación 16-QAM sólo se diferencia de la modulación 8-QAM en la cantidad de niveles y fases, en este caso se agrupan los bits en bloques de cuatro, y se pueden generar cuatro amplitudes y cuatro fases diferentes, generando las 16 combinaciones.

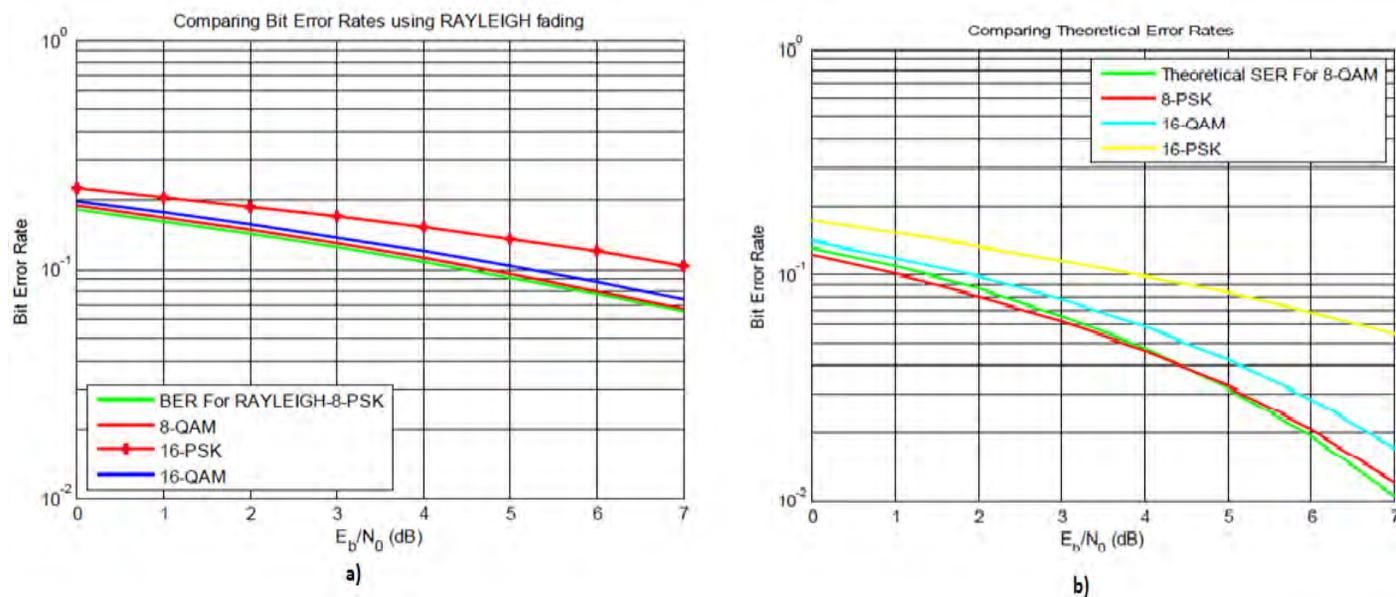


Figura 3 Comparación entre 16-QAM y 16-8PSK a) Para un canal Rayleigh b) Para un canal AWGN Adaptada de [32]

En las Figura 3 se realiza el mismo análisis, en este caso para una constelación de $M = 16$, donde tenemos los casos de 16-PSK y 16-QAM. En 1.3 a) se realiza el estudio comparativo para el caso de un canal de tipo Rayleigh, y se puede apreciar que el desempeño de 16-QAM es superior en cuanto al BER, para un mismo valor de E_b/N_0 . La misma situación se repite en el caso del análisis del BER teórico de ambas modulaciones, considerando un canal AWGN, el cual se muestra en 1.3 b).

Luego del análisis realizado, proponemos el uso de la modulación QPSK para $M = 4$ y 8-PSK para $M = 8$, mientras que para $M = 16$ seleccionamos 16-QAM. Estos serán los posibles esquemas de modulación a los que tendrá acceso el sistema de comunicaciones.

1.4 Esquemas de modulación adaptativos

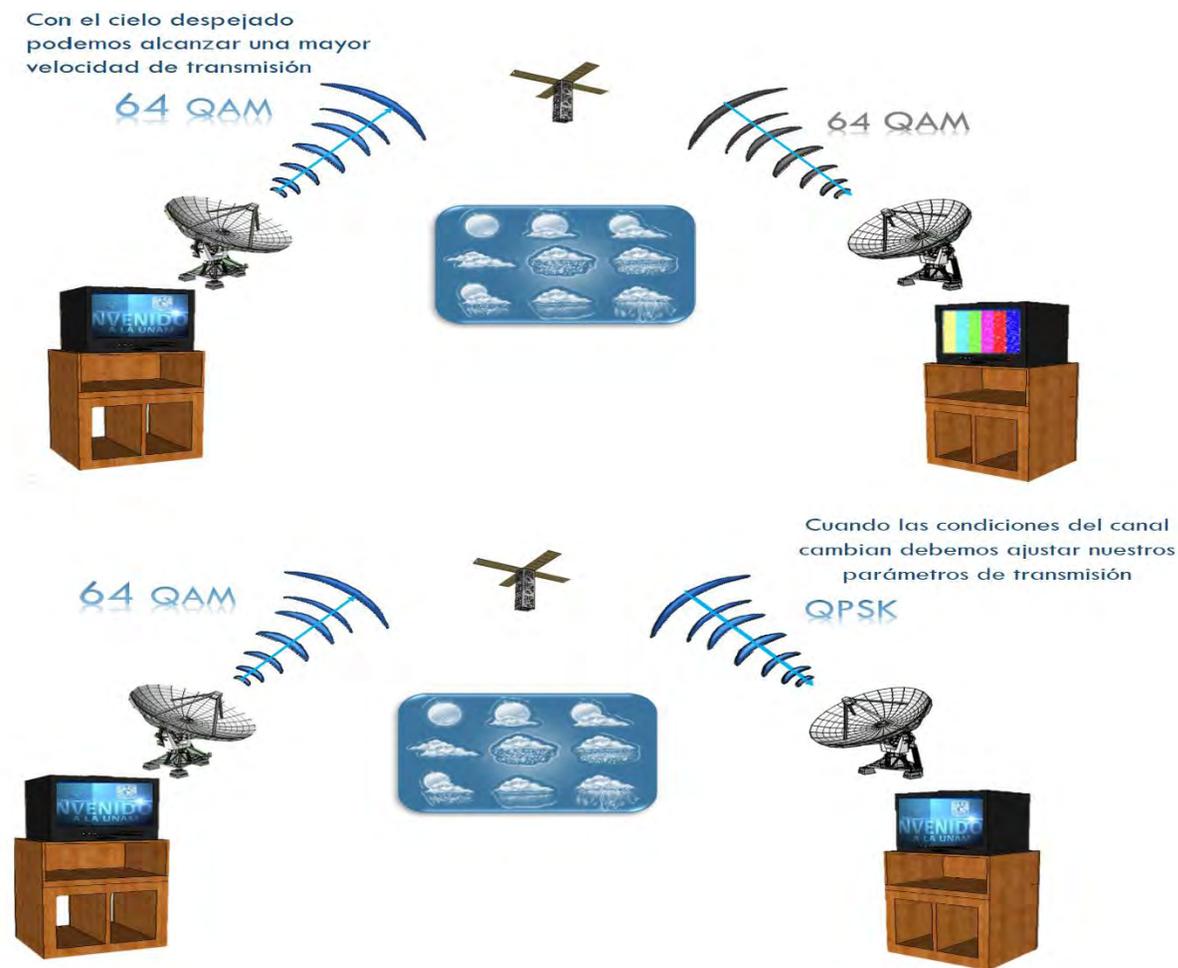


Figura 4 ACM Punto a Punto

Los esquemas de modulación fijos son diseñados para operar en las peores condiciones de relación señal a ruido (SNR, por sus siglas en inglés) y no pueden aprovechar cuando dicha relación se eleva. Un sistema de modulación adaptativo (ACM, por sus siglas en

inglés) que usa diferentes patrones de codificación y modulación puede ofrecer mejores prestaciones en medios de transmisión como en el que se encuentran los satélites LEO. [1]

Este concepto ha sido usado en las comunicaciones inalámbricas como comunicaciones terrestres por microondas y más recientemente en enlaces satelitales [11]. Su función principal es optimizar el rendimiento en un enlace de datos adaptando la modulación y la codificación con respecto al ruido presente en un momento dado. La modificación de estos parámetros influye directamente en la eficiencia espectral, expresada en bits por segundo por Hertz. Un ejemplo genérico de ACM se muestra en la Figura 4.

1.5 Estado del arte de los sistemas de modulación adaptativa para aplicaciones espaciales

Con el objetivo de conocer los antecedentes de la aplicación de esquemas de modulación adaptativos en sistemas de comunicaciones diseñados para operar en satélites pequeños, se realizó una revisión bibliográfica donde se estudiaron proyectos en los cuales se usaran o se propusieran esquemas de modulación adaptativos. Este estudio nos permitió además conocer los principales tipos de modulación usados en sistemas de comunicaciones para aplicaciones espaciales, y algunos parámetros de diseño de los mismos.

En [1], se propone un ACM usando modulación QAM con diferentes patrones de constelación (ej. 16 QAM 32 QAM) para obtener diferentes tasas de transmisión. Esto se une a un codificador Trellis, con el objetivo de obtener ganancias de codificación. El sistema es simulado usando un modelo Rician para el canal de transmisión, asumiendo varios valores de calidad de SNR. Las simulaciones mostraron un buen desempeño del sistema a la hora de usar el SNR en ambientes de ruido blanco gaussiano aditivo (AWGN por sus siglas en inglés) y con desvanecimiento moderado. Sin embargo, bajo efectos severos de desvanecimiento Rician/Rayleigh, el sistema mostró no ser adecuado, y se propone el uso de corrección de error hacia adelante (FEC, por sus siglas en inglés).

En [12], se propone el uso del código convolucional Turbo Code, de conjunto con varios niveles de modulación QAM, desde 4-QAM hasta 512-QAM. La estrategia propuesta, es dividir el rango de SNR en $N + 1$ regiones de desvanecimiento, y a cada una de estas regiones se le asigna una constelación. Así, cuando el SNR estimado está en una región específica, la constelación correspondiente es usada. La simulación del sistema muestra que este es adaptable a los cambios en un ambiente con desvanecimiento, asumiendo una buena estimación del canal y poco tiempo de demora. Usando 10 regiones de desvanecimiento, el BER promedio se aproxima a 10^{-6} cuando el SNR es más de 10 dB, y se obtienen ganancias de hasta 20 dB con respecto a la modulación no adaptativa.

En [13] se propone el uso de SDR con ACM para su uso en un satélite CubeSat. Se usan como modulaciones BPSK, QPSK, 8-PSK, 16, 32 y 64-QAM. El procesamiento digital de señal es realizado en un FPGA Zynq 7200, y posee un mecanismo de refrescamiento para asegurar el funcionamiento del sistema ante fallas.

En la Tabla 1 se muestra un resumen de los elementos principales detectados en la revisión bibliográfica realizada. El primer elemento a señalar es el uso predominante de modulaciones de fase o de amplitud y fase, las cuales van desde modulaciones con una tasa de transmisión binaria como BPSK la cual es muy robusta, pero permite velocidades de transmisión bajas, hasta 32QAM, la cual permite velocidades de transmisión superiores a costa de una necesidad de potencia de transmisión superior en aras de mantener un nivel de cantidad de errores.

Otro elemento a tener en cuenta es el uso de codificadores. Se puede apreciar que en algunos proyectos se usa mientras que en otros no. Esto depende fundamentalmente de la cantidad de errores que se pueda permitir el sistema. El uso de codificadores minimiza

la cantidad de errores, pero a costa de agregar redundancia al sistema, lo cual implica un desperdicio de potencia y de recursos de forma general.

Tabla 1 Resumen de la revisión bibliográfica

Artículo	Esquemas de modulación utilizados	Codificación	Criterio de selección del tipo de modulación	Tecnología
1	16QAM, 32QAM	Trellis, FEC	SNR	No mencionado
2	Desde QAM hasta 512 QAM	TurboCode	SNR	No mencionado
3	BPSK, QPSK, 8 PSK, 16 QAM, 32 QAM	Sin Codificación	No mencionado	FPGA Zync 7200

El siguiente elemento a tener en cuenta es el criterio de selección de la modulación a usar, donde predomina el uso de la relación señal a ruido o SNR, el cual es una medida de desempeño muy popular en cualquier sistema de comunicaciones.

Por último, vale destacar que en los proyectos que se revisaron, se pudo detectar el uso de FPGAs SRAM para diseño electrónico en sistemas espaciales, el cual es el tipo de dispositivo electrónico propuesto para la implementación de nuestro sistema.

1.6 Base teórica de las modulaciones seleccionadas para la modelación adaptativa

1.6.1 Modulación QPSK

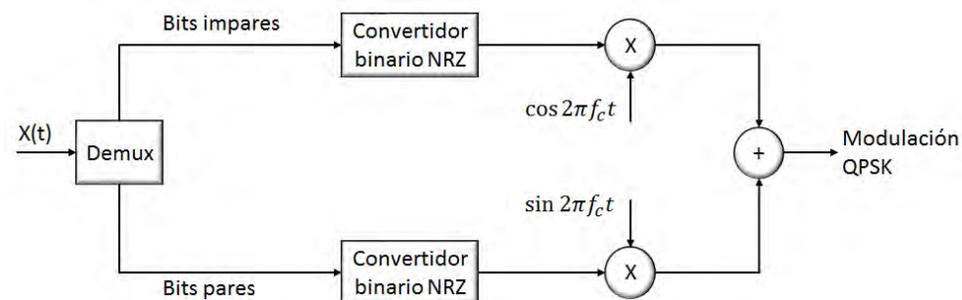


Figura 5 Esquema del modulador QPSK

El modulador QPSK (Figura 5), para generar la señal modulada, toma el flujo de datos binarios de entrada y lo divide mediante un demultiplexor en bits pares e impares, cada uno de los flujos de salida tiene el doble del período del flujo de datos original. Los bits resultantes son convertidos a la forma polar usando un codificador no-retorno a cero (NRZ por sus siglas en inglés), donde los símbolos

1 y 0 son representados ahora por $\sqrt{E_b}$ y $-\sqrt{E_b}$, respectivamente. Estos flujos de datos son multiplicados por las señales en fase y cuadratura respectivamente, para luego ser combinados y obtener la señal QPSK.

De esta forma se tendrá a la salida del modulador 4 posibles corrimientos de fase con respecto a la señal portadora generada en nuestro oscilador local $\sqrt{\frac{2}{T}} \sin(2\pi f_c t)$, los cuales son:

$\varphi_{1,2,3,4} = \frac{\pi}{4}; 3\frac{\pi}{4}; 5\frac{\pi}{4}; 7\frac{\pi}{4}$ respectivamente. La constelación de la modulación QPSK se muestra en la Figura 6.

Podemos observar que la constelación está basada en código Grey, de manera que entre símbolos adyacentes solo cambie un bit, con el objetivo de minimizar errores en la transmisión.

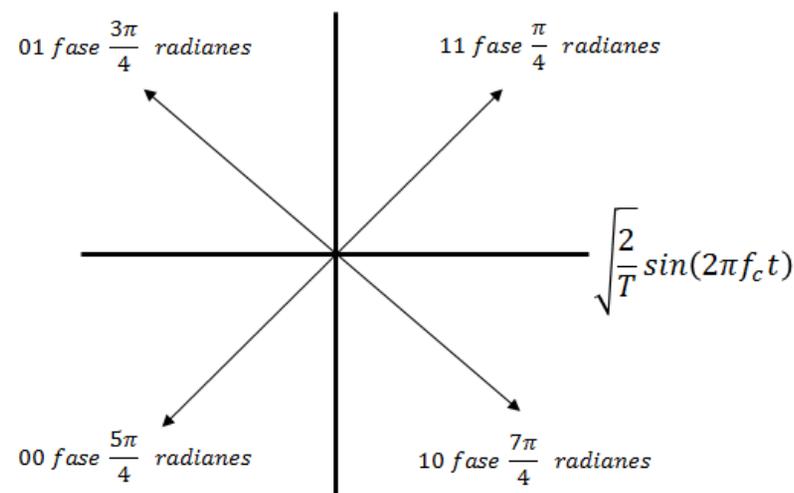


Figura 6 Constelación de la modulación QPSK

1.6.2 Modulación 8-PSK

En la Figura 7 se muestra el esquema de un modulador PSK de 8 fases. Los bits de entrada serial se agrupan en bloques de 3 bits o tribits, y dependiendo del orden de llegada estos se denominan C, I y Q, siendo C el primero en llegar. En el caso del bit I, controla un convertidor de 2 a 4 niveles, el cual toma como entrada dos bits y produce una de las cuatro salidas posibles. El otro bit usado para controlar el bloque es el bit C. De esta forma, el bit I controla la polaridad a la salida del convertidor (1 ó -1) mientras que el bit C controla el nivel “analógico” de la salida. En el caso del bit Q, sigue la misma lógica que el bit I, la diferencia es que en este caso el bit C se toma negado, de esta forma se garantiza que las salidas de los convertidores nunca coincidan en amplitud, factor necesario para garantizar el requerimiento de amplitud constante de la modulación 8PSK. Con este arreglo se garantiza que la combinación de amplitudes siempre genere el mismo valor final de amplitud.

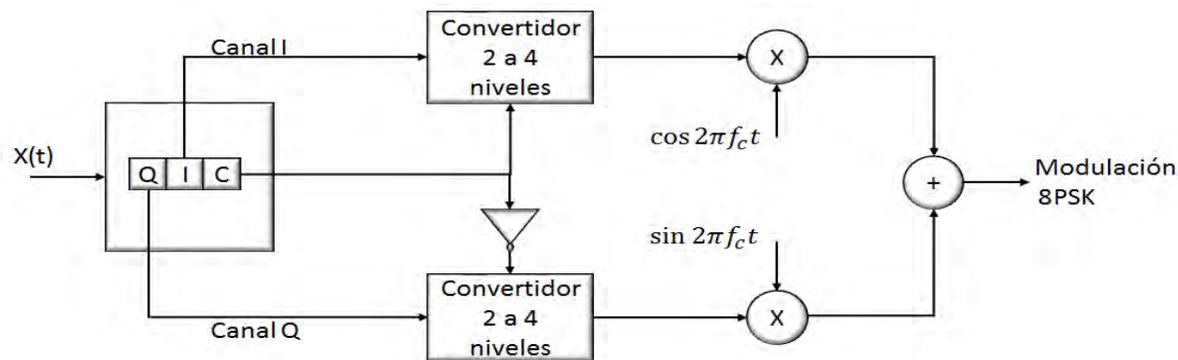


Figura 7 Esquema del modulador 8PSK

Para implementar el bloque convertidor de 2 a 4 niveles, es necesario determinar los valores de amplitud necesarios para obtener una amplitud constante. Debemos tener en cuenta que la separación angular entre cada vector de la constelación 8 PSK es de 45° , que es el resultado de dividir 360° entre 8, como se muestra en la Figura 8.

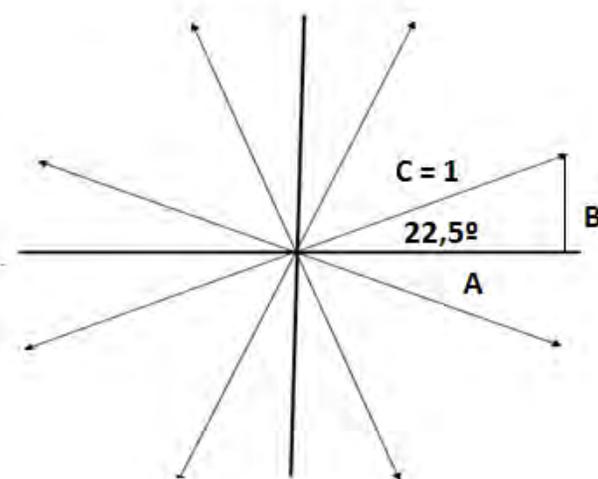


Figura 8 Determinación de los coeficientes de amplitud para un modulador 8 PSK

De la Figura 8 podemos deducir los valores de amplitud que deben combinarse para garantizar la amplitud deseada, en este caso se seleccionó como amplitud un vector de amplitud 1, pero se podría haber tomado cualquier otro valor arbitrario. Usando propiedades de los triángulos obtenemos las ecuaciones 1.1 y 1.2, las cuáles conforman un sistema de ecuaciones que debemos resolver.

$$\frac{B}{A} = \tan 22.5^\circ = 0.414 \quad (1.1)$$

$$C^2 = A^2 + B^2 \quad (1.2)$$

Resolviendo el sistema de ecuaciones planteado obtenemos los siguientes valores:

$$A = 0.91; B = 0.38$$

En la tabla 1.2 se muestra la tabla de la verdad para los valores de A, B y C:

Tabla 2 Tabla de la verdad del bloque Convertidor de 2 a 4 niveles

A ó B	C	Salida
0	0	-0.91
0	1	0.91
1	0	-0.38
1	1	0.38

La salida del sumador se muestra en forma vectorial, donde se puede apreciar cómo queda conformada la constelación. Vemos que, con dos posibles amplitudes y dos posibles signos, se realizan las combinaciones de las señales seno y coseno, obteniéndose solamente 8 combinaciones ya que la presencia del bloque negador en el canal C provoca que no existan nunca combinaciones de seno y coseno con igual amplitud. Con las combinaciones mostradas en la Figura 9, garantizamos la separación deseada entre cada fase de 45° , como se puede apreciar en la Tabla 3. Si se realiza el desarrollo matemático de cada una de las expresiones mostradas en la Figura 9, el resultado es el enlistado en la Tabla 3, donde también se muestra la combinación de valores necesarios en los canales I, Q y C para producir dicha salida.

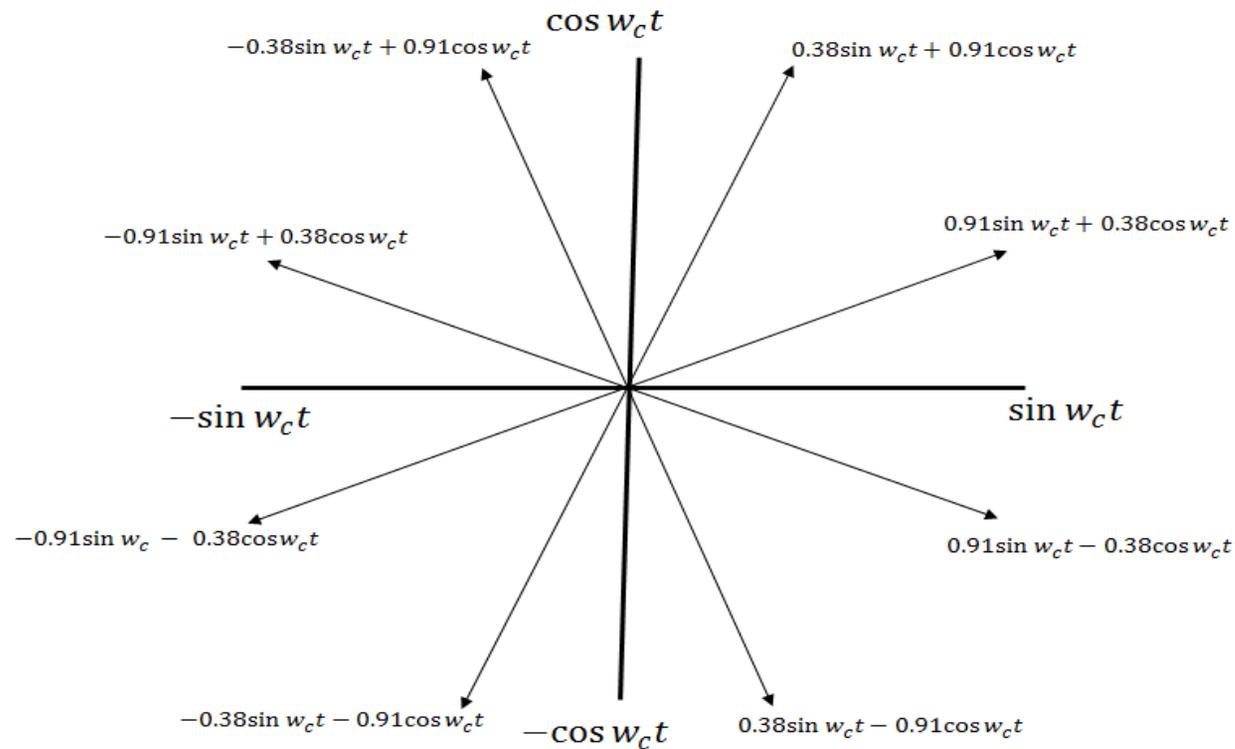


Figura 9 Posibles combinaciones de la modulación 8-PSK

Tabla 3 Salida del modulador 8-PSK dependiendo de los valores de I, Q y C

Q	I	C	Ecuación	Ángulo
0	0	0	$C\sin(\omega_c t - 112.5^\circ)$	-112.5
0	0	1	$C\sin(\omega_c t - 157.5^\circ)$	-157.5
0	1	0	$C\sin(\omega_c t - 67.5^\circ)$	-67.5
0	1	1	$C\sin(\omega_c t - 22.5^\circ)$	-22.5
1	0	0	$C\sin(\omega_c t + 112.5^\circ)$	+112.5
1	0	1	$C\sin(\omega_c t + 157.5^\circ)$	+157.5
1	1	0	$C\sin(\omega_c t + 67.5^\circ)$	+67.5
1	1	1	$C\sin(\omega_c t + 22.5^\circ)$	+22.5

Podemos comprobar que se cumpla lo planteado en la Tabla 3. Tomemos por ejemplo la combinación QIC = 000. En este caso, para el canal I, el convertidor de 2 a 4 niveles selecciona amplitud $A = 0.91$ porque $C = 0$ y polaridad negativa ya que $Q = 0$, quedando $Q = -0.91\cos\omega_c t$. Por otro lado, el convertidor de 2 a 4 niveles del canal I selecciona amplitud $B = 0.38$ y polaridad negativa por ser $I = 0$, quedando $I = -0.38\sin\omega_c t$. Nuestra señal a la salida del sumador será entonces:

$$S = -0.38\sin\omega_c t - 0.91\cos\omega_c t$$

$$S = -0.38\angle 0 - 0.91\angle 90 = -0.38 - 0.91j = \sin(\omega_c t - 112.5)$$

La constelación de la modulación 8-PSK se muestra en la Figura 10, la cual constituye la unión entre la información de la Figura 9 y la Tabla 3.

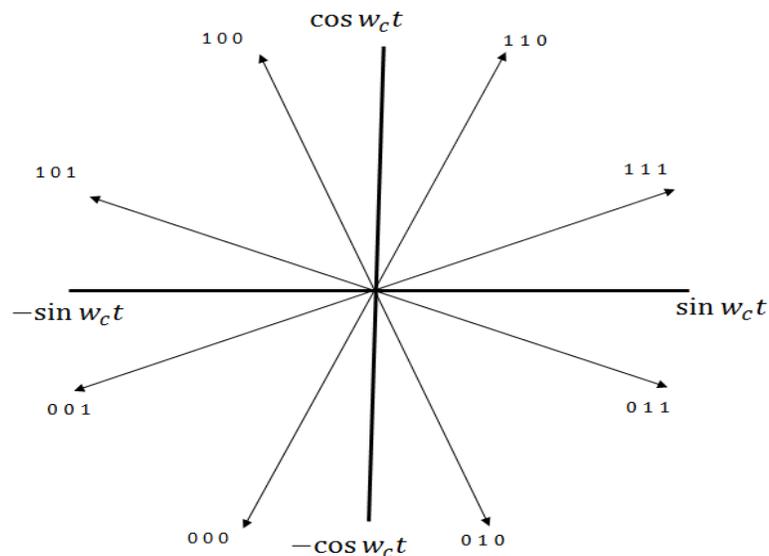


Figura 10 Constelación 8PSK

1.6.3 Modulación 16-QAM

La modulación QAM es un esquema de modulación M-ario donde la información está contenida tanto en la fase como en la amplitud de la señal portadora. En el específico de la modulación 16-QAM, $M = 16$, de ahí que los bits de entrada sean agrupados en bloques de 4 bits y cada canal de este bloque tendrá una tasa de $\frac{f_b}{4}$. El diagrama a bloques de un modulador 16-QAM se muestra en la Figura 11. Los datos de entrada son divididos en 4 canales I, I', Q y Q' cada uno con una tasa 4 veces menor a la de los datos de entrada. Los canales I y Q controlan un conversor de 2 a 4 niveles cada uno. Los canales I' y Q' controlan la magnitud de salida del conversor, mientras que los I y Q controlan la polaridad.

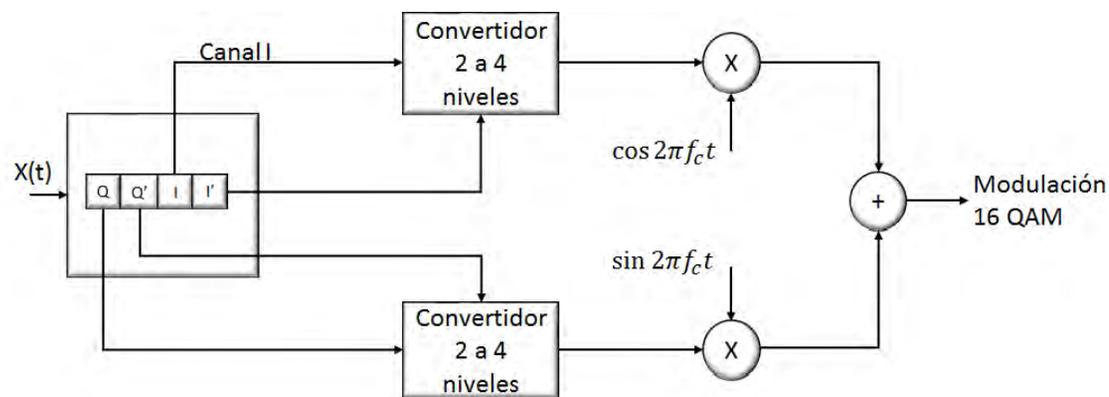


Figura 11 Modulador 16-QAM

Para obtener los valores de los coeficientes A y B del conversor de 2 a 4 niveles se usaron las propiedades geométricas del triángulo obtenido de unir los puntos de la constelación 16-QAM, como se muestra en la Figura 12.

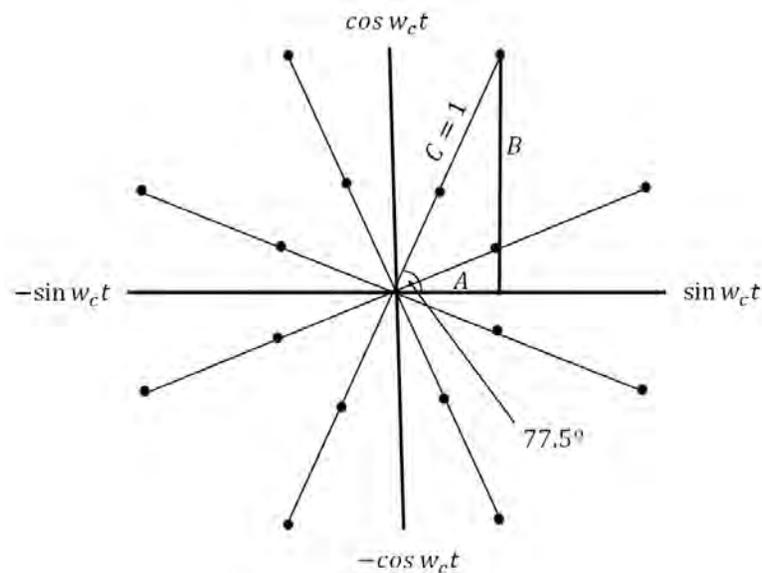


Figura 12 Triángulo obtenido de unir los puntos de la constelación 16-QAM

Usando las propiedades de los triángulos, formamos el siguiente sistema de ecuaciones

$$C = \sqrt{A^2 + B^2} \quad (1.3) \quad \frac{B}{A} = \tan 75^\circ \quad (1.4)$$

Resolviendo el sistema de ecuaciones obtenemos: $A = 0.22$; $B = 0.82$. La tabla de la verdad del conversor de 2 a 4 niveles a usar en el modulador 16-QAM se muestra en la Tabla 4. En este caso se muestra para los bits I, ya que es la misma para los bits Q.

Tabla 4 Tabla de la verdad del conversor de 2 a 4 niveles para el modulador 16QAM

I	I'	
	0	1
0	-A	-B
1	B	A

Tabla 5 Salida del sumador dependiendo de los bits I, I', Q y Q'

I	I'	Q	Q'	Salida Sumador
0	0	0	0	$B\sqrt{2}\angle - 135$
0	0	0	1	$C\angle - 135$
0	0	1	0	$B\sqrt{2}\angle - 45$
0	0	1	1	$C\angle - 15$
0	1	0	0	$C\angle - 105$
0	1	0	1	$A\sqrt{2}\angle - 135$
0	1	1	0	$C\sqrt{2}\angle - 75$
0	1	1	1	$A\sqrt{2}\angle - 45$
1	0	0	0	$B\sqrt{2}\angle 135$
1	0	0	1	$C\angle 135$
1	0	1	0	$B\sqrt{2}\angle 45$
1	0	1	1	$C\angle 15$
1	1	0	0	$C\angle 105$
1	1	0	1	$A\sqrt{2}\angle 135$
1	1	1	0	$C\angle 75$
1	1	1	1	$A\sqrt{2}\angle 45$

Las salidas de los convertidores de 2 a 4 niveles del modulador 16-QAM son multiplicadas por la señal portadora correspondiente, para luego ser combinadas mediante un sumador. La salida del sumador dependiendo de los valores de I, Q, I' y Q' se muestran en la Tabla 5.

1.7 Conclusiones del Capítulo 1

En este capítulo se presentaron los fundamentos teóricos que se usarán para la implementación del sistema. Se realizó un estudio donde se exponen los esquemas de modulación más usados en aplicaciones espaciales, partiendo del cual se realizó una comparación entre los esquemas de modulación identificados seleccionando QPSK, 8PSK y 16QAM como los esquemas a usar en nuestro sistema. Posteriormente se realizó el estudio del estado del arte de la modulación adaptativa en aplicaciones espaciales, haciendo especial énfasis en aquellas que se basan en FPGAs. Finalmente se presentó la base teórica para la construcción de los moduladores seleccionados para ser usados en nuestro sistema.

Capítulo 2. Determinación de los requerimientos para el diseño del sistema de comunicaciones

En este capítulo se realiza un modelo del canal de comunicaciones mediante simulación que nos permite determinar las principales afectaciones que sufrirá nuestro sistema, las cuáles definirán las técnicas y parámetros de diseño que serán utilizados posteriormente en el proceso de diseño e implementación. Se obtiene una aproximación del comportamiento de los parámetros de transmisión, específicamente el comportamiento de la tasa de bit erróneo, (BER, por sus siglas en inglés). Para esto usamos los requerimientos que se tienen del diseño de la misión.

2.1 Simulación del enlace de comunicaciones en STK

Para nuestra aplicación específica, seleccionamos el software STK (System Toolkit), de la empresa AGI (Analytical Graphics, Inc). El mismo es una herramienta de modelado tanto en 2D como 3D para la representación de sistemas complejos como naves espaciales, satélites y estaciones terrenas. El software es capaz de modelar, analizar y representar de forma gráfica objetos en tiempo y espacio en diferentes ambientes como el espacio, el cielo, la tierra y el mar. Los modelos creados en STK son usados por cientos de organizaciones a nivel global para modelar sistemas de vuelo y comunicaciones y evaluar su desempeño en tiempo real o tiempo simulado. Cuenta con diversos módulos que proveen al usuario de capacidades más sofisticadas en cuanto a modelado, simulación y análisis. Usamos la versión de prueba del módulo de comunicaciones de este software con el objetivo de simular el enlace de comunicaciones de nuestro microsatélite, obteniendo los valores de SNR y BER específicos para diferentes tipos de modulación.

2.1.1 Elementos de mecánica Orbital

Para la realización de dicha simulación se partió de los parámetros ya definidos de la misión [14] los cuales son:

- a. Tipo de Satélite: Microsatélite
- b. Tipo de Órbita: LEO Solar Síncrona
- c. Altura: 500 km
- d. Ubicación: UNAM 19.38 ° Latitud Norte -99.18 ° Longitud Oeste

Comenzamos insertando la estación terrena la cual supondremos se encontrará ubicada en la UNAM (Figura 13). Para esto es necesario crear el objeto *Facility* dentro de nuestro escenario de trabajo, y configurar la posición geográfica del mismo. El software STK cuenta con una base de datos tanto para satélites como para estaciones terrenas existentes, loque facilita la inserción de elementos dentro del modelo y agiliza el proceso de construcción del mismo. También cuenta con la posibilidad de insertar coordenadas específicas en caso de que el sitio que queremos analizar no se encuentre específicamente en la base de datos. En nuestro caso usaremos esta última opción, ya que proponemos que la estación terrena se encuentre en el Facultad de Ingeniería Eléctrica de la UNAM. Las coordenadas propuestas son *UNAM 19.38 ° Latitud Norte -99.18 ° Longitud Oeste*.

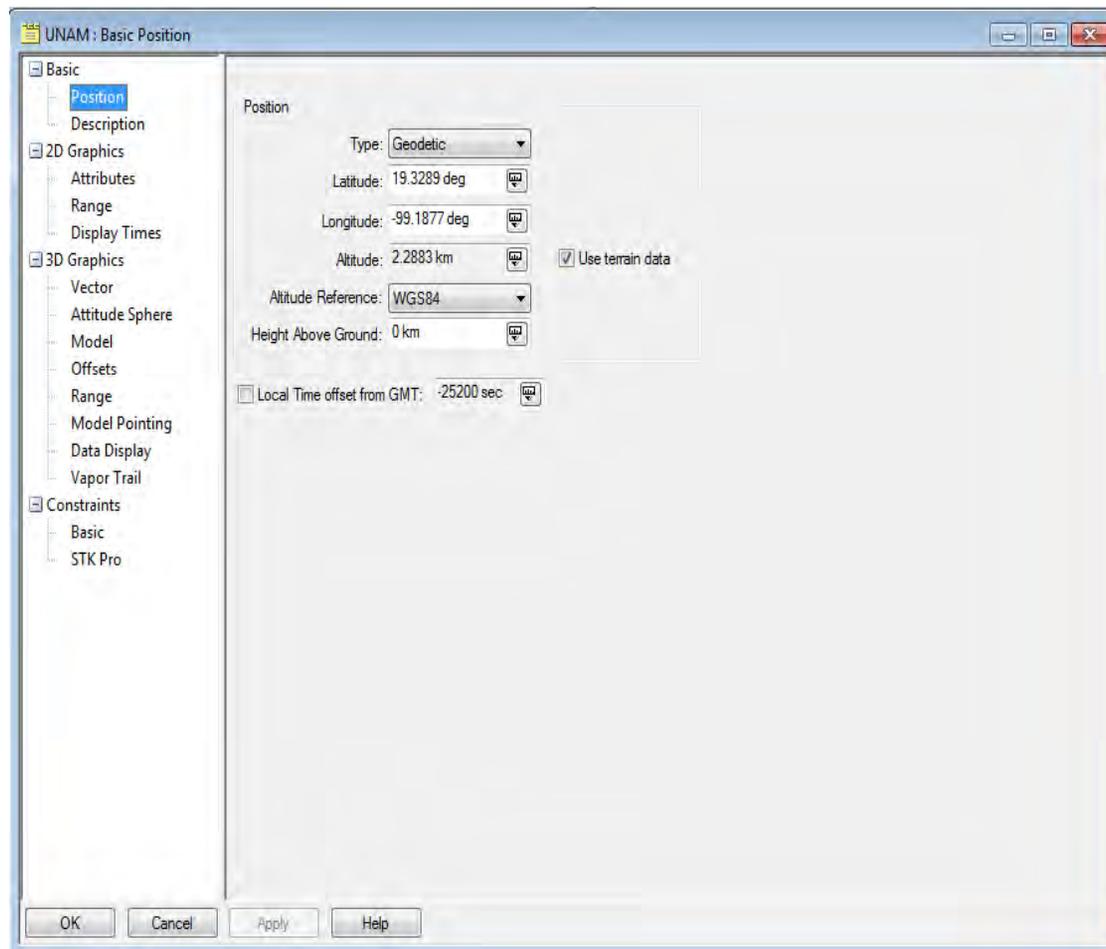


Figura 13 Inserción de la estación terrena en STK

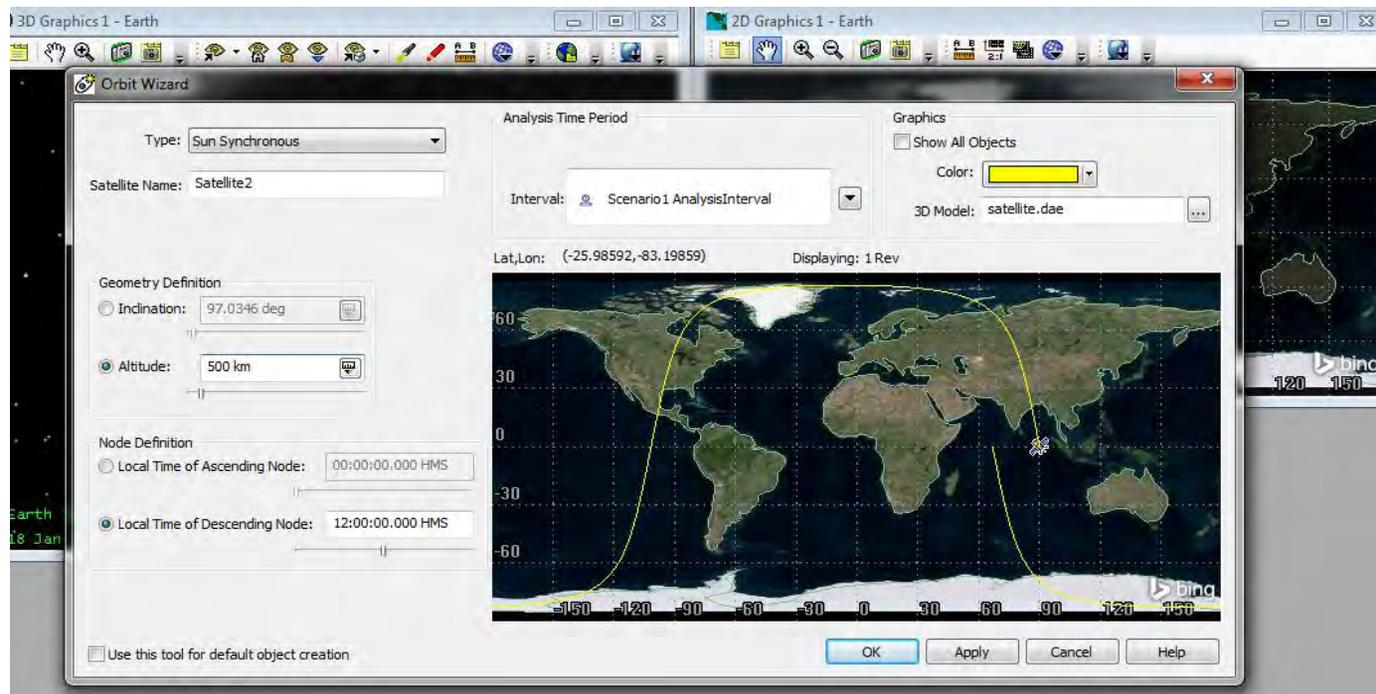


Figura 14 Inserción del satélite en STK

Posteriormente insertamos nuestro satélite mediante el Orbit Wizzard que posee el software, con los parámetros orbitales mostrados en la sección 2.1.1, tipo de órbita y altura (Figura 14).

2.1.2 Bandas de Operación

La banda de operación seleccionada se muestra en la Tabla 6. Para la selección de la misma se usó el estándar European Corporation for Space Standardization (ECSS), específicamente el ECSS-E-ST-50-05C REV. 2, oct. 2011, el cual lo podemos encontrar en la sección de Ingeniería de la página en Internet de la ECSS, apartado *Atribuciones de frecuencias al servicio de operaciones espaciales*.

Tabla 6 Bandas de Operación

Tipo de Enlace	Banda de Subida (MHz)	Banda de Bajada (MHz)
Banda S	2025,833 a 20108,708	2200 a 2290

Para determinar si existe algún tipo de restricción en México con respecto a estas frecuencias se revisa el reglamento de Radiocomunicaciones, 2012, y se determinó que no existen tales restricciones.

2.1.3 Antena de estación terrena

Tabla 7 Antena de la estación terrena

Parámetro	Valor
Frecuencia	2,2 GHz
G/T	12dB/K
Tipo de antena	Parabólica
Diámetro	3,4 m
Ganancia del Lóbulo principal	40 dB
Ángulo de Apertura	170°
Rango de Frecuencias	Bandas L, S, X, C, Ku, Ka

Fuente: TCS Space & Component Technology

Se propone el uso de la antena desarrollada por TCS Space & Component Technology, sus parámetros se muestran en la Tabla 7

2.1.4 Ancho de banda Permitido

Para la determinación del ancho de banda permitido se usó como fuente el ECSS-E-ST-50 punto 4.2 *Condiciones específicas para el uso de ciertas bandas de frecuencia*. En el mismo se especifica que se debe usar un ancho de banda de 6 MHz

Con estos parámetros podemos configurar la antena de la estación terrena (Figura 15). De las características mostradas en la Tabla 7 el software usa la frecuencia y la figura de mérito G/T para realizar la estimación de los parámetros de la antena. Para la inserción de la antena en el software es necesario seleccionar el elemento *Receiver* del módulo de comunicaciones, y asociarlo a la estación terrena.

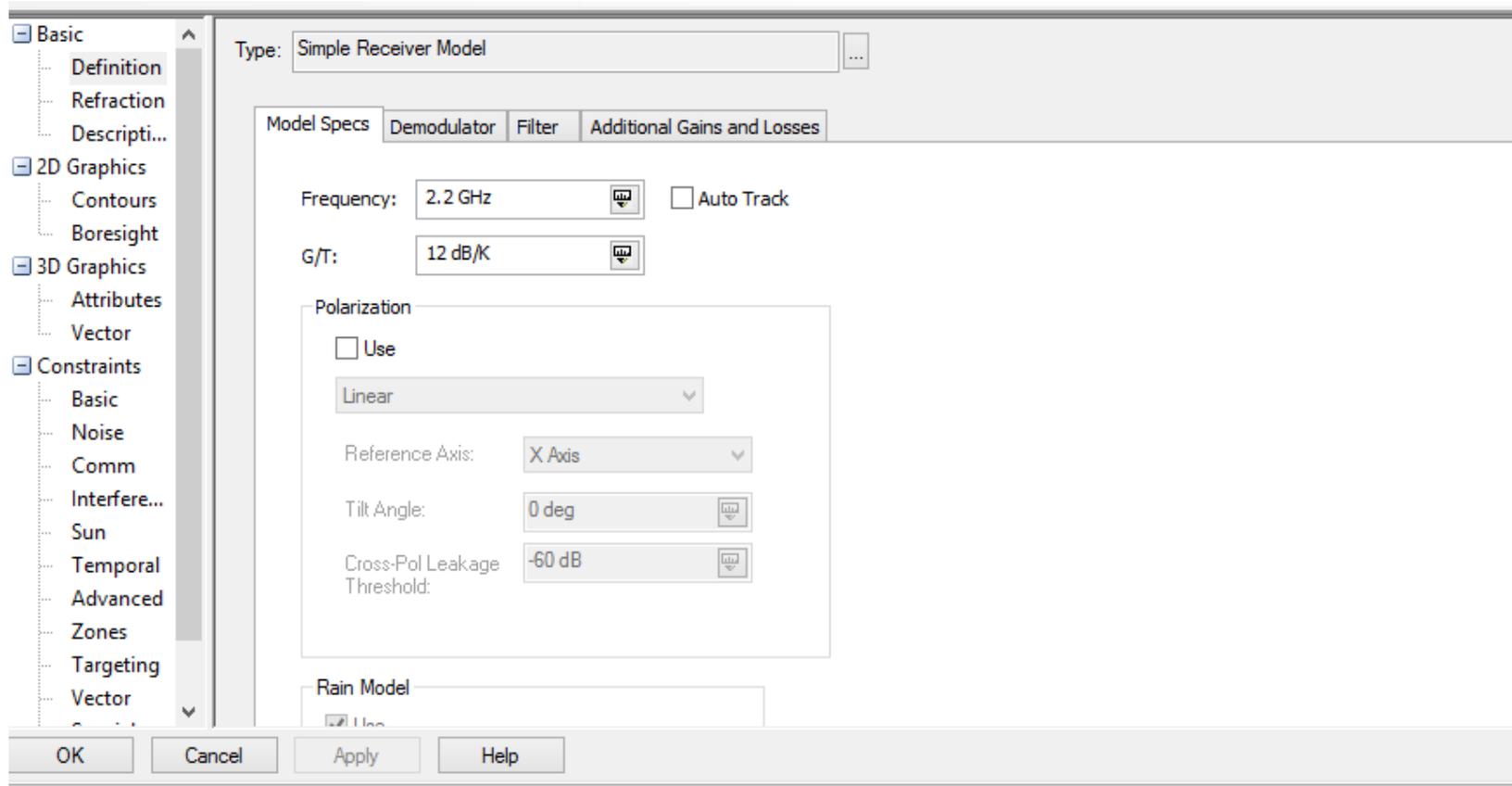


Figura 15 Configuración de la antena de la estación terrena en STK

2.1.5 Antena del Satélite

Las características de la antena del satélite se muestran en la Tabla 2.3

Tabla 8 Antena del satélite

Parámetro	Antena de Clyde Space
Banda de Frecuencia	S, 2.2 GHz
Potencia RF	5 W
Ancho de Haz	60°
Masa	50 g
Ganancia	8 dBi

Fuente: NASA/TP-2014-216648/REV

Para la inserción en STK de la antena transmisora del satélite (Figura 16), es necesario insertar primeramente un elemento *Sensor* en el satélite (para este ejercicio denominado Cándor), posteriormente insertar el elemento *Transmitter* del módulo de comunicaciones y asociarlo al sensor, en este caso del satélite. Los parámetros de diseño de la antena son configurables en el sistema, como se puede apreciar en la Figura 16.

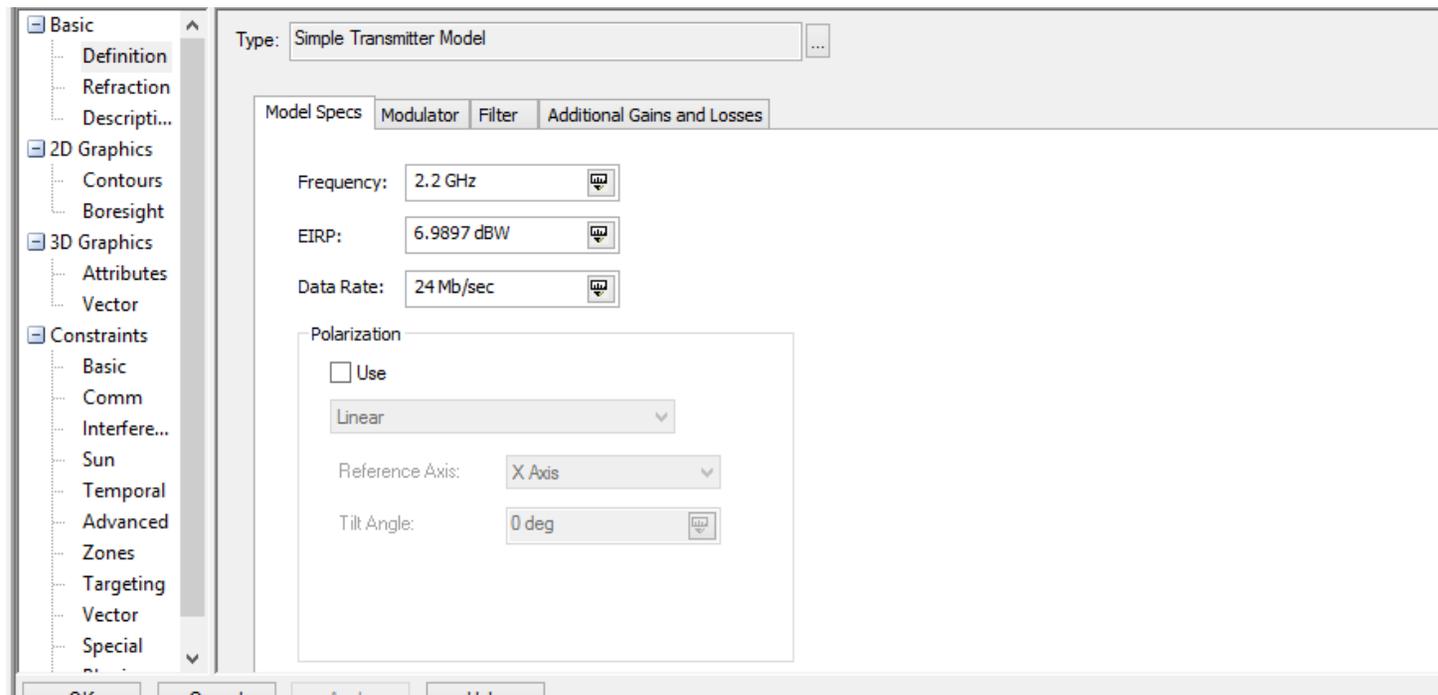


Figura 16 Configuración de la antena del satélite

Una vez configurado el sistema podemos ejecutar el simulador y obtener la vista 2D (Figura 17) donde se muestra cuántos pasajes tendrá el satélite en un período de simulación de 24 horas, en este caso se escogió la misma fecha en que se realizó la simulación, del 2 al 3 de febrero de 2017.

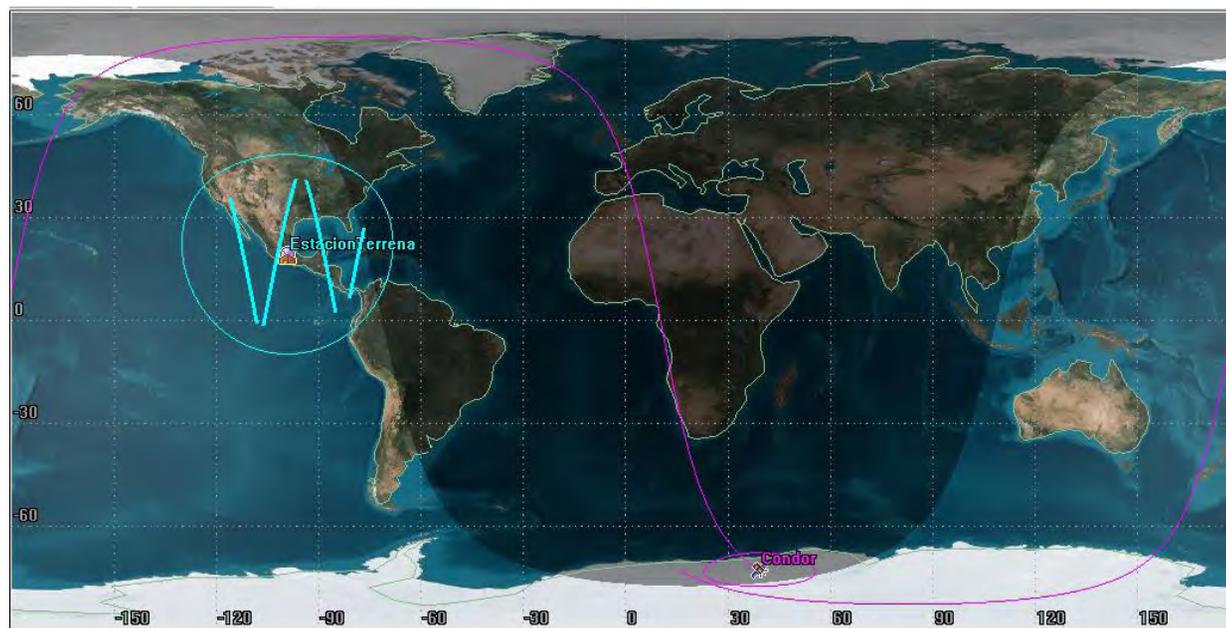


Figura 17 Vista 2D de la órbita del satélite

En la Figura 17 podemos ver que el satélite tendrá 4 eventos de línea de vista con la estación terrena en el período seleccionado. Para cada uno de estos enlaces se analizó el comportamiento del canal de transmisión, variando solamente la modulación usada. Este análisis nos permitirá comprender el comportamiento del canal de comunicaciones para obtener un diseño que se ajuste mejor al mismo.

2.2 Resultados de la simulación en STK

Con los requerimientos mencionados anteriormente se procede al desarrollo de la simulación, variando la modulación para obtener los valores de SNR y BER correspondientes. El software STK realiza la estimación del BER en función de la relación señal a ruido Eb/No. La ecuación para el cálculo del BER depende de la modulación usada, aunque todas usan la función Q [5].

$$BER_{QPSK} = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \quad (2.1)$$

$$BER_{8PSK} = 2Q\left(\sqrt{\frac{2E_s}{N_o}} * \text{sen} \frac{\pi}{M}\right) \quad (2.2)$$

$$BER_{16QAM} = \frac{2(1-L^{-1})}{\log_2 L} * Q\left(\sqrt{\frac{3 \log_2 L}{(L^2-1)^2}} * 2 \frac{E_b}{N_o}\right) \quad (2.3)$$

Donde Eb es la energía del bit y No es el ruido. En el caso de la modulación 8PSK, Ecuación 1.2, M representa la cantidad de niveles de la modulación, en este caso M = 8, y se usa Es (energía del símbolo) en lugar de Eb; estas se encuentran relacionadas por la Ecuación 2.4

$$E_s = E_b \log_2 M \quad (2.4)$$

Para la modulación 16QAM se usa L para referirse a la dimensión de la modulación, en este caso M = 16.

La función Q, usada para estimar la tasa de bit erróneo en todas las modulaciones, tiene la siguiente expresión

$$Q = \frac{1}{\sqrt{\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du \quad (2.5)$$

EL software de simulación STK estima la relación Eb/No mediante la Ecuación 2.6 [15]

$$\frac{E_b}{N_o} [dB] = \frac{C}{N_o} [dB * Hz] - R_b [dB * bits/s] \quad (2.6)$$

Donde C/No es la relación portadora a ruido y Rb es la tasa de transmisión. Para el cálculo de la relación portadora a ruido se usa la Ecuación 2.7

$$\frac{C}{N_o} = \frac{(EIRP)L_p L_a G_r}{k T_{eq}} \quad (2.7)$$

Donde:

EIRP es la potencia isotrópica equivalente radiada

Lp son las pérdidas por propagación entre el transmisor y el receptor

La son las pérdidas por atenuación atmosférica

Gr es la ganancia de la antena de recepción

k es la constante de Boltzmann (-228.6 dB)

T_{eq} es la temperatura de ruido

Si expresamos los términos de la ecuación 1.8 en decibeles, obtenemos

$$\frac{C}{N_o} [dB * Hz] = EIRP[dBW] + L_{prop}[dB] + \frac{G}{T} \left[\frac{dB}{K} \right] + k[dB] \quad (1.9)$$

Donde

L_{prop} es la totalidad de pérdidas en el enlace

G/T es la ganancia de la antena de recepción con respecto a la temperatura total del sistema, también conocido como figura de mérito

Los resultados obtenidos en la simulación se muestran en la Figura 18, la cual corresponden al comportamiento del BER para un evento de línea de vista entre el satélite y la estación terrena, para las modulaciones QPSK, 8PSK y 16QAM, en donde se observa que la cantidad de errores durante un evento de línea de vista tiene grandes variaciones, siendo mayor al inicio y al final del enlace, mientras que sus menores valores se encuentran siempre hacia el medio del enlace. Esto se debe a que a medida que se progresa en el tiempo dentro del enlace, el satélite se acerca a la estación terrena, alcanzando su distancia mínima alrededor de la mitad del enlace, momento a partir del cual comienza a alejarse nuevamente. Este comportamiento es característico de todos los eventos de línea de vista evaluados

Comportamiento del BER Enlace 1

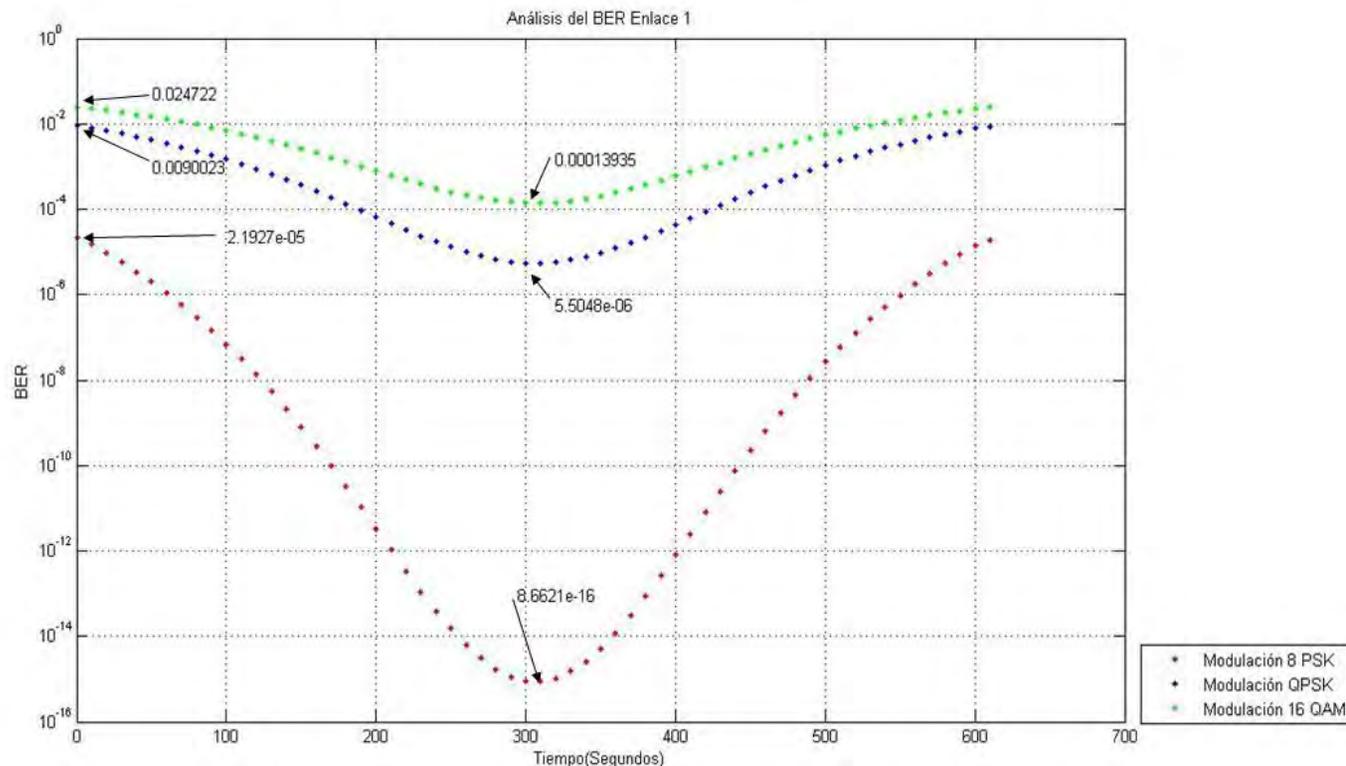


Figura 18 Comportamiento del BER para un evento de línea de vista

Este comportamiento sugiere la aplicación de esquemas de modulación adaptativos, de manera que se puedan aprovechar los puntos donde la cantidad de errores es muy baja, usando un esquema de modulación que permita, con estas condiciones, alcanzar la mayor velocidad de transmisión posible, sin sobrepasar una cantidad de errores prefijada. Esto significa que existirán puntos de la trayectoria donde, para cumplir con el requerimiento de BER, solo podremos seleccionar un tipo de modulación, en otros puntos podremos escoger entre dos tipos y en las mejores condiciones del canal podríamos escoger cualquier modulación. En cada caso se seleccionará siempre la modulación que, cumpliendo con el requisito de BER, nos permita obtener la mayor velocidad de transmisión posible.

2.3 Metodología de conmutación del sistema

Para determinar el protocolo de conmutación del sistema de comunicaciones, se realizó una simulación del enlace satelital durante todo un año, con el objetivo de tener una mejor caracterización del canal de comunicaciones. Primeramente, se obtuvo el histograma de duración de los eventos de línea de vista entre el satélite y la estación terrena, el cual se muestra en la Figura. 19, donde podemos observar que la mayor parte de los enlaces tiene una duración superior a los 500 segundos u 8 minutos.

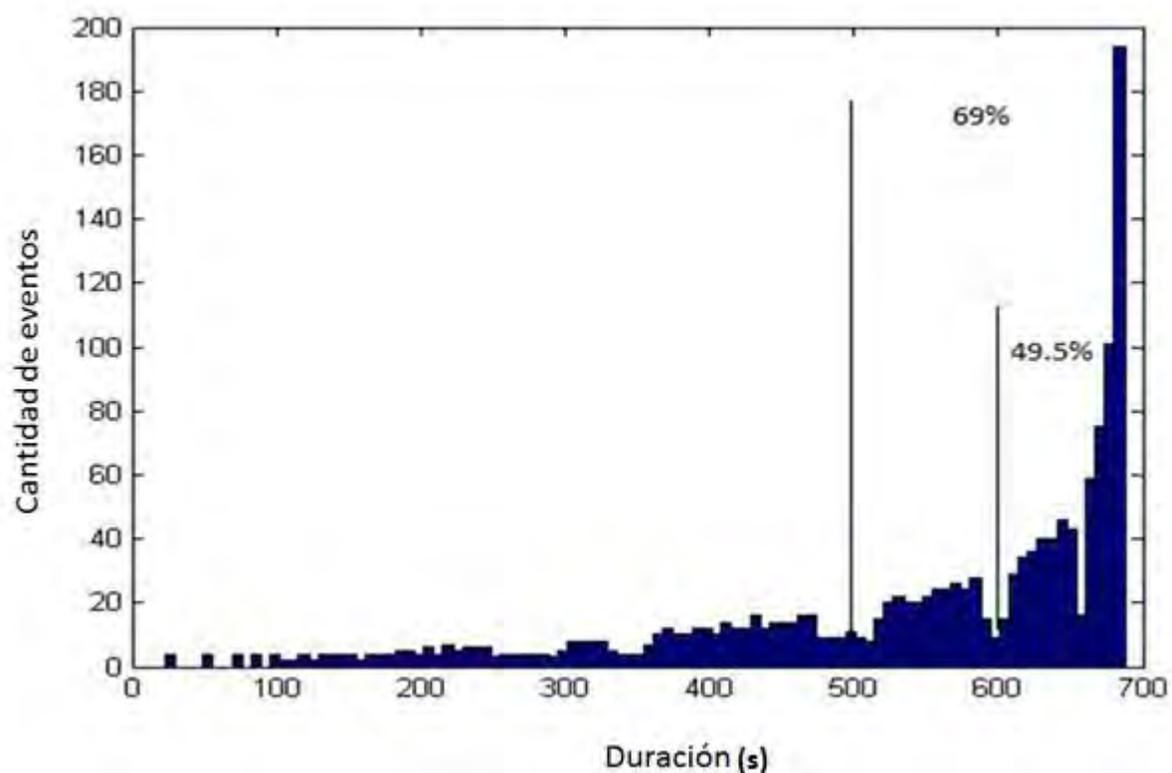


Figura 19 Histograma de duración de los eventos de línea de vista

El comportamiento del BER para las modulaciones propuestas se muestra en la Figura 20. Podemos ver que, tanto al inicio como hacia el fin del enlace, solo es posible el uso de una modulación, ya que el resto no cumple con el requerimiento de BER. Llega un punto donde es posible seleccionar entre dos modulaciones, y posteriormente es posible escoger entre cualquiera de las modulaciones propuestas, en este lapso se puede escoger la que mayor velocidad de transmisión permita.

Se realizó entonces un análisis estadístico de los datos obtenidos donde se pudo dimensionar la cantidad de tiempo que transcurre luego de que el satélite comienza la comunicación con la estación terrena, hasta que las condiciones del enlace permiten iniciar el proceso de conmutación entre modulaciones. Para esto se simularon todos los eventos de línea de vista durante un año, obteniéndose 1470 eventos. Para cada evento se evaluó el tiempo que transcurre entre el inicio del enlace y la primera conmutación entre modulaciones.

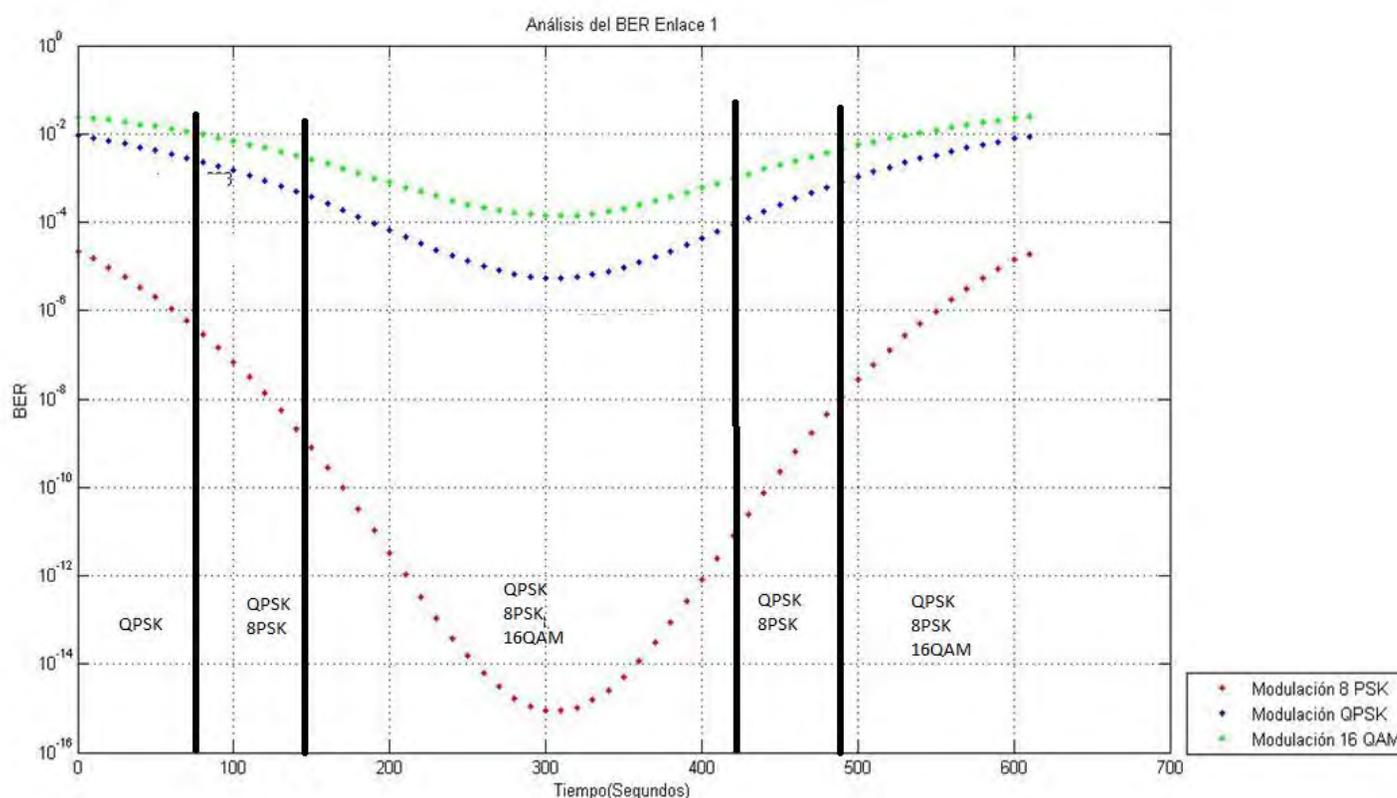


Figura 20 Comportamiento del BER para un evento de línea de vista

En la Figura 21 se muestra el histograma de los valores de tiempo obtenidos entre el inicio del enlace y la primera conmutación entre modulaciones, se puede apreciar que alrededor de 800 de un total de 1076 enlaces, presentan condiciones favorables para el inicio del proceso de conmutación 180 segundos después de iniciado el enlace.

Debido a que los valores de BER obtenidos se encuentran separados a intervalos de 60 segundos, por características propias del software de simulación, podemos concluir que a partir de los 120 primeros segundos del enlace existe la posibilidad de que éste presente condiciones favorables para la conmutación, y en el 80% de los casos, se puede realizar la conmutación en el minuto 3, por tanto, proponemos que en un enlace que dure por lo menos 7 minutos el sistema de conmutación comience a explorar la posibilidad de conmutación a partir de los 2 primeros minutos luego de iniciado el enlace, esto considerando que en este caso tendremos por lo menos 1 minuto en el que se tendrá la seguridad de que podremos aplicar la conmutación entre modulaciones.

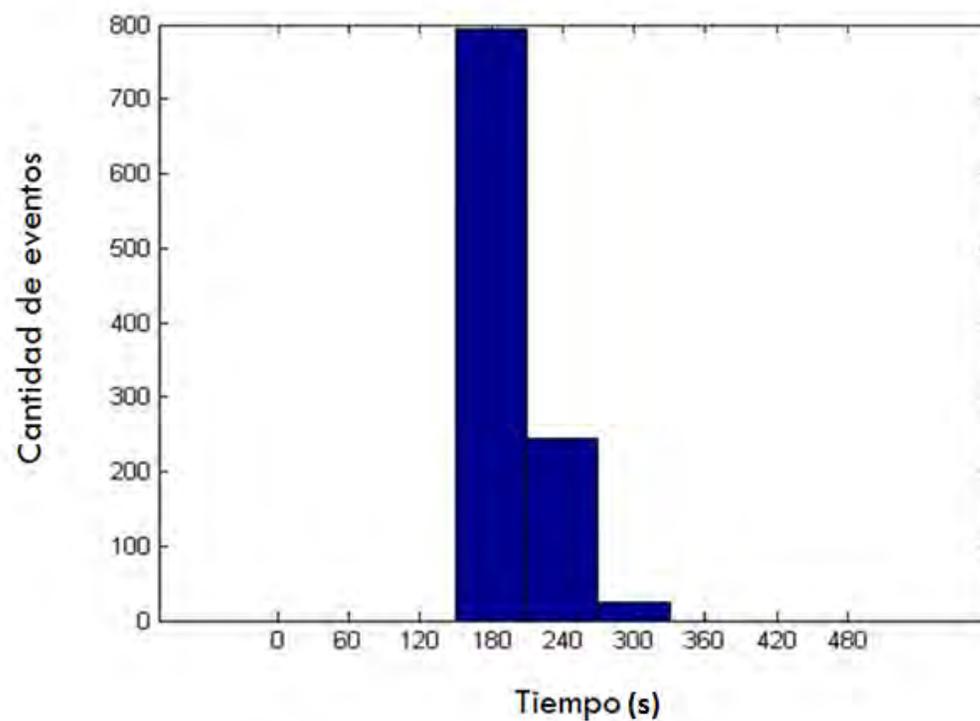


Figura 21 Histograma del tiempo que transcurre antes de que esté listo el enlace para la conmutación

El comportamiento del sistema de conmutación para un evento de línea de vista de duración T se muestra en la Figura 22.

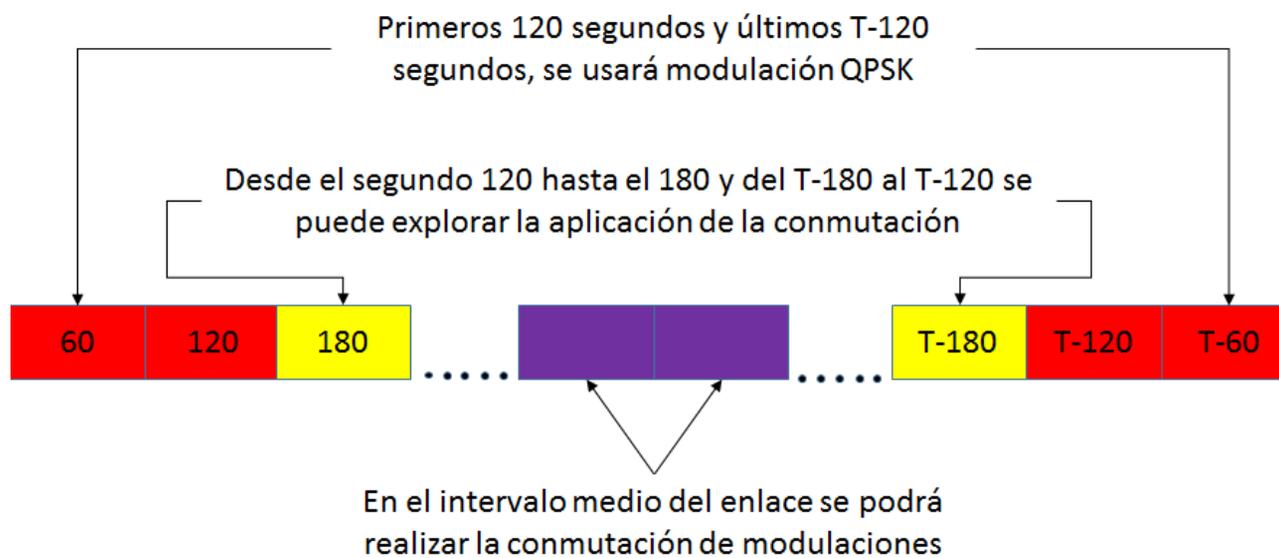


Figura 22 Intervalo de aplicación de la conmutación entre modulaciones

De la Figura. 22 podemos determinar que la duración mínima que debe tener un enlace para considerar la aplicación de la conmutación de las modulaciones debe ser de 7 minutos, para garantizar tener al menos 1 minuto en el cual se usen modulaciones de orden superior a la modulación QPSK

2.4 Estimación de las demoras de transmisión

Otro elemento a tener en cuenta es la demora que tiene una señal en llegar al receptor en tierra luego de ser transmitida. Para realizar este cálculo se obtuvo de las simulaciones las distancias a las que se encuentra el satélite de la estación terrena. Primeramente, se analizaron la totalidad de los eventos de vista, y posteriormente los eventos de línea de vista que duren más de 420 segundos, duración mínima propuesta para la aplicación de la conmutación, durante un período de un año. Para este análisis se usa la posibilidad que tiene el STK de exportar los datos en formato Excel, y posteriormente se realizó el análisis de la información para su presentación en forma de histograma.

En la Figura 23 se muestra el histograma para las distancias de todos los eventos de vista. Podemos ver que el valor mínimo de distancia entre el satélite y la estación terrena es de 500 km mientras que el valor máximo es de 2890 km. La distancia media para todos los eventos de línea de vista fue de 1900 km, con un 55% de los enlaces cuya distancia estuvo por encima de la media.

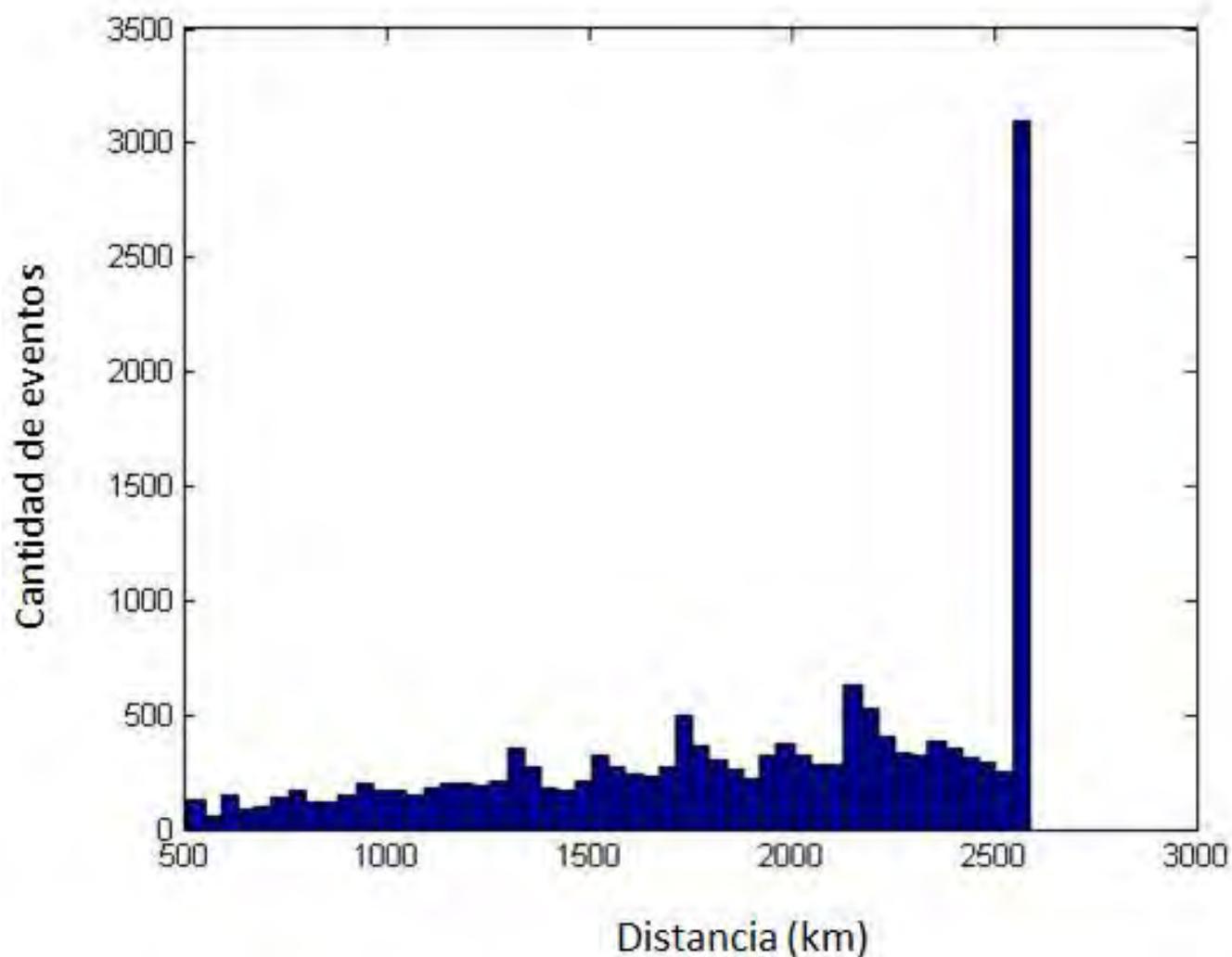


Figura 23 Histograma de distancias durante un año

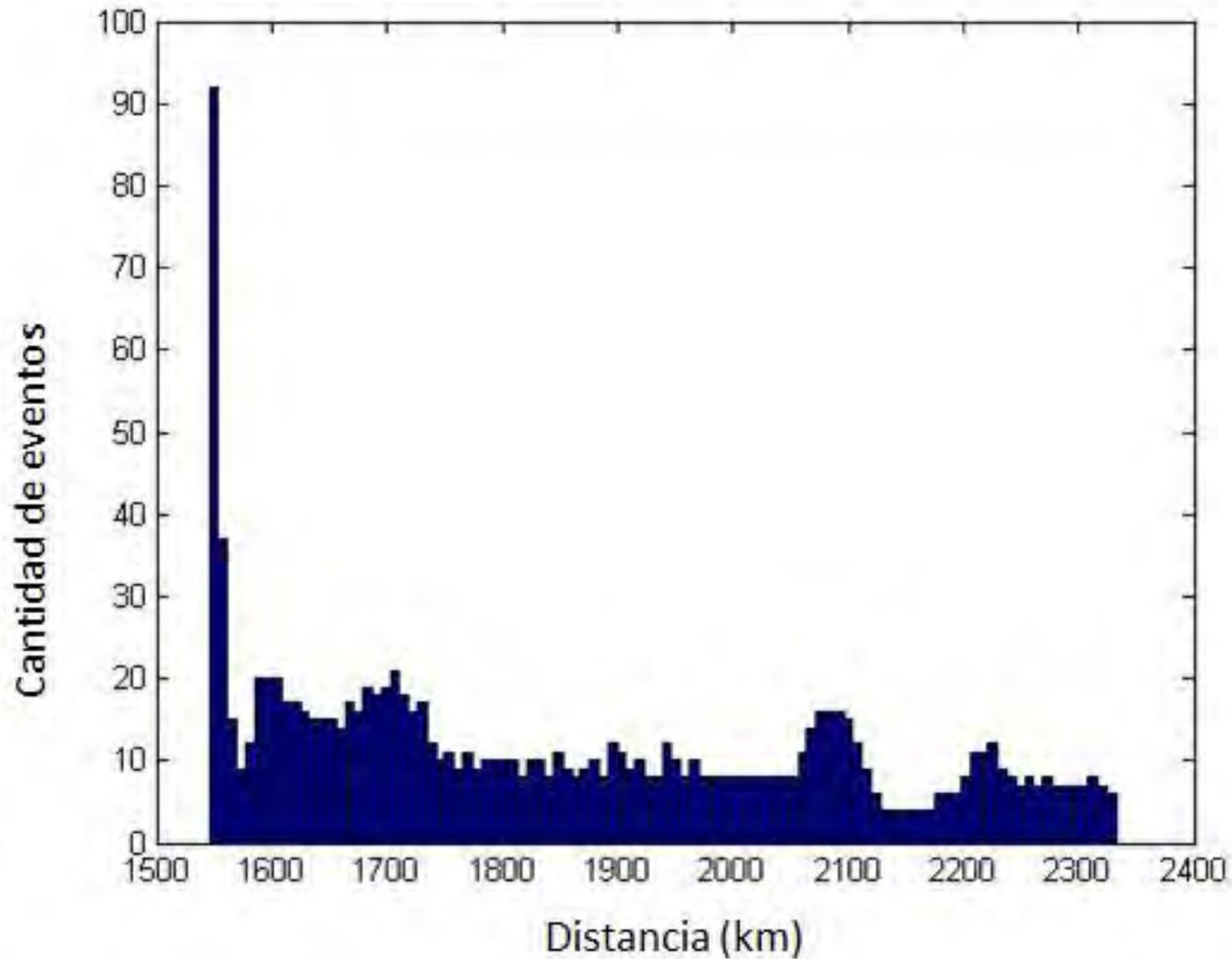


Figura 24 Histograma de distancias para eventos de duración superior a los 420 segundos

En la Figura 24 se muestra el histograma para las distancias de los eventos que tienen una duración superior a los 420 segundos, en este caso podemos apreciar una concentración alrededor de los valores entre 1500 y 1800 km, obteniéndose como promedio una distancia de 1850 km. Para calcular la demora de propagación de la señal transmitida, usamos una aproximación dividiendo la distancia a recorrer entre la velocidad de la luz. Las demoras del enlace para estos valores de distancia se muestran en la Tabla 9

Tabla 9 . Demoras de transmisión

Distancia (km)	Demora (ms)
Mínima 1545	5.15
Máxima 2331	7.7
Promedio 1850	6.1

Para nuestros análisis usaremos el peor caso de demora, 7,7 milisegundos.

2.5 Estimación del BER en el canal de transmisión

Para la estimación del BER y la posterior selección de la modulación a usar, proponemos el uso de una técnica conocida como Modulación Asistida por Símbolo Piloto (PSAM, por sus siglas en inglés) [16], [17]. Esta técnica consiste en la inserción por parte del modulador de símbolos piloto en la trama de información, de manera que el receptor puede usar estos símbolos conocidos para realizar una estimación del canal. El esquema general de esta técnica se muestra en la Figura 25.

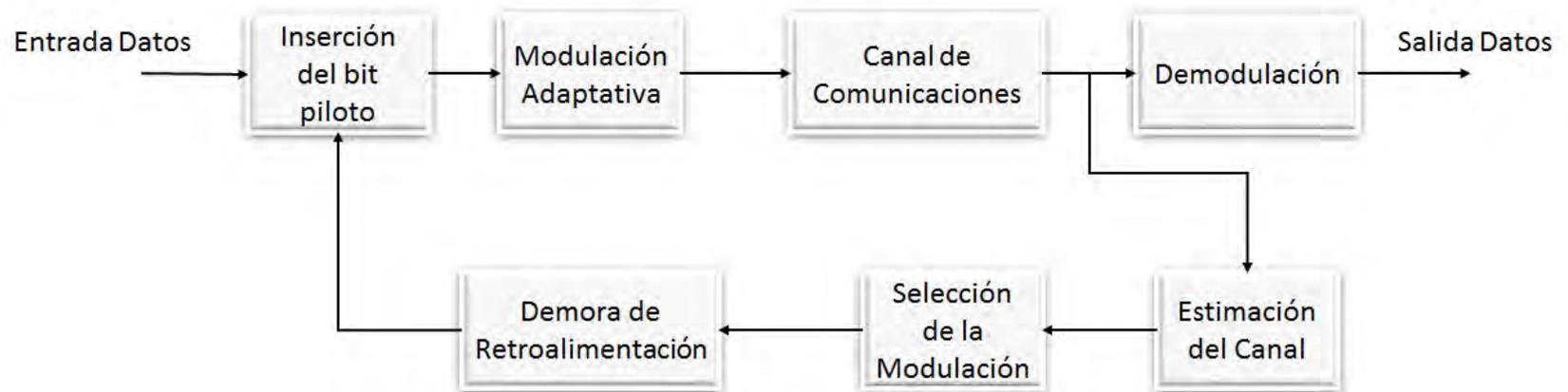
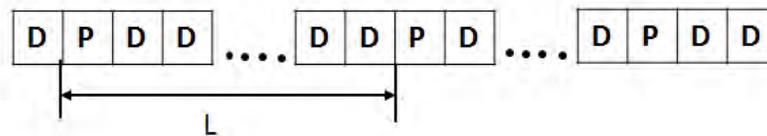


Diagrama a bloques del sistema PSAM



Estructura de la trama

Figura 25 Modelo del sistema PSAM Adaptado de [17]

En el receptor, un estimador de canal extrae los símbolos piloto para realizar la estimación del canal y luego realiza la selección de la modulación que mejor se ajuste a dicha estimación. Los elementos principales a tener en cuenta en este sistema son la longitud de la trama, la cual denotaremos L , la cantidad de símbolos piloto necesarios para realizar la predicción, la cual denotaremos K , y la demora en retroalimentar al transmisor con la información generada en el proceso de estimación en el receptor, que en nuestro caso será de 7,7 milisegundos, como se mostró en la Tabla 9.

En [16] se muestra que los valores de L y K deben tener un valor tal que se logre un equilibrio entre la cantidad de redundancia y la estimación del canal. Mucha redundancia en busca de una mejor y más precisa estimación del canal se traduce en desperdicio de potencia, mientras que muy poca redundancia con el objetivo de minimizar el consumo de potencia se traduce en una estimación errónea del canal.

El valor de L se debe escoger de forma que cumpla con la siguiente relación: $L > 1/2f_dT_s$; donde f_d es el desplazamiento por efecto Doppler, y T_s es el período del símbolo. En el caso específico de los microsátélites, el desplazamiento Doppler se encuentra

caracterizado alrededor de los 200 Hz. La mayor tasa de transmisión que se manejará en el sistema es de 24 Mbps, usando la modulación 16 QAM, en la cual se necesitan 4 bits para formar un símbolo, por tanto, en nuestro caso $T_s = 0.16 \mu s$, con el cual obtenemos un L máximo aproximado de 15000. De esta manera la cantidad de información redundante que se agregará será de $\frac{1}{15000} = 0.0000066\%$.

Para escoger el valor de K, en [16] y [17] se toma como parámetro de diseño la demora de retroalimentación, estableciéndose que se debe escoger un valor de K tal que el tiempo que transcurre entre estimaciones debe ser superior a la demora de retroalimentación. El tiempo que transcurre entre una estimación y la siguiente está determinado por la relación KLT_s , como se muestra en la Figura 2.14. Podemos ver que cada trama está conformada por L bits, de los cuales 1 se usa para formar la cadena con la que se realizará la estimación del canal. De aquí que la cantidad de bits necesarios para realizar la estimación determina el número de tramas que se ocuparán, y el tiempo que transcurre entre una estimación y otra está dado por el tiempo que se necesita para transmitir KL bits, le cual se obtiene multiplicando la cantidad de bits por la duración de los mismos.

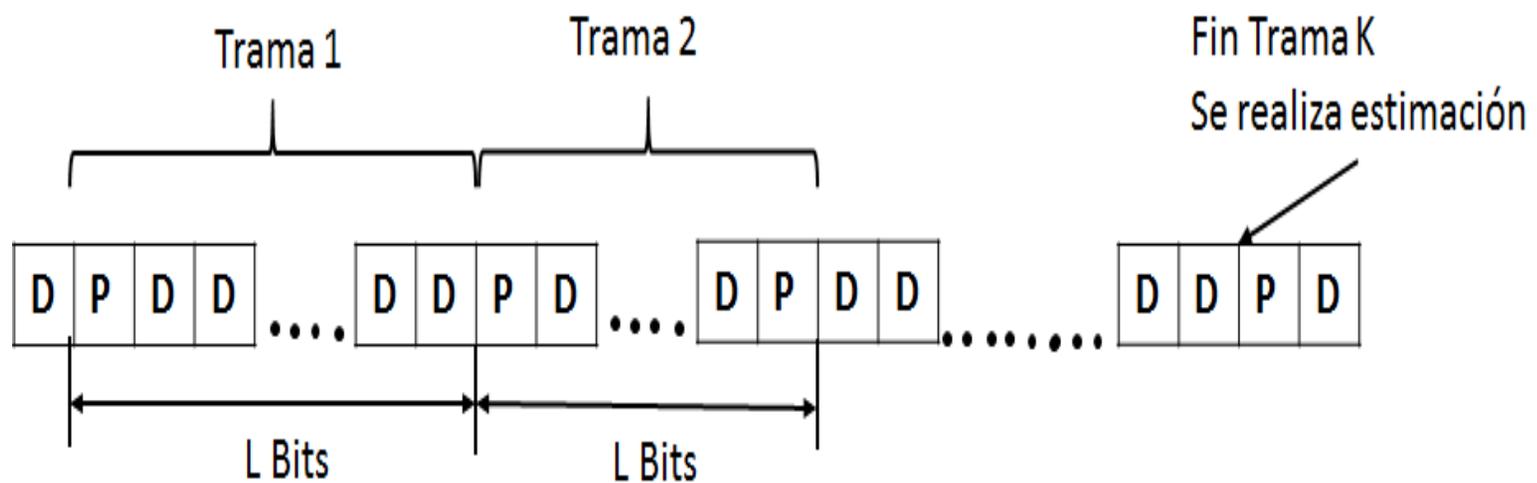


Figura 26 Determinación del tiempo entre estimaciones Adaptado de [17]

En nuestro caso seleccionaremos $K = 15$ ya que la estimación se realizaría cada KMT_s , lo cual arroja 20 ms para QPSK, 13 ms para 8PSK y 10 ms para 16 QAM, valores que se encuentran todos por encima a la demora de transmisión seleccionada, la cual es de 7.7 ms.

Para seleccionar el esquema de modulación a usar, en [17] se propone el uso del valor estimado del canal y el error de estimación del canal para encontrar el BER condicional para modulaciones M-PSK y M-QAM. De esta forma, dado un valor estimado del BER para una modulación específica, y conociendo la medida del error cometido en la estimación, el receptor puede estimar el BER que se obtendría usando este canal para otras modulaciones, seleccionando entonces la de mejor desempeño que cumpla con los requerimientos establecidos previamente. Asimismo, el resultado de la estimación obtenida se puede usar para seleccionar la técnica de demodulación a utilizar.

2.6 Comparación de las velocidades obtenidas usando modulación adaptativa vs modulación fija

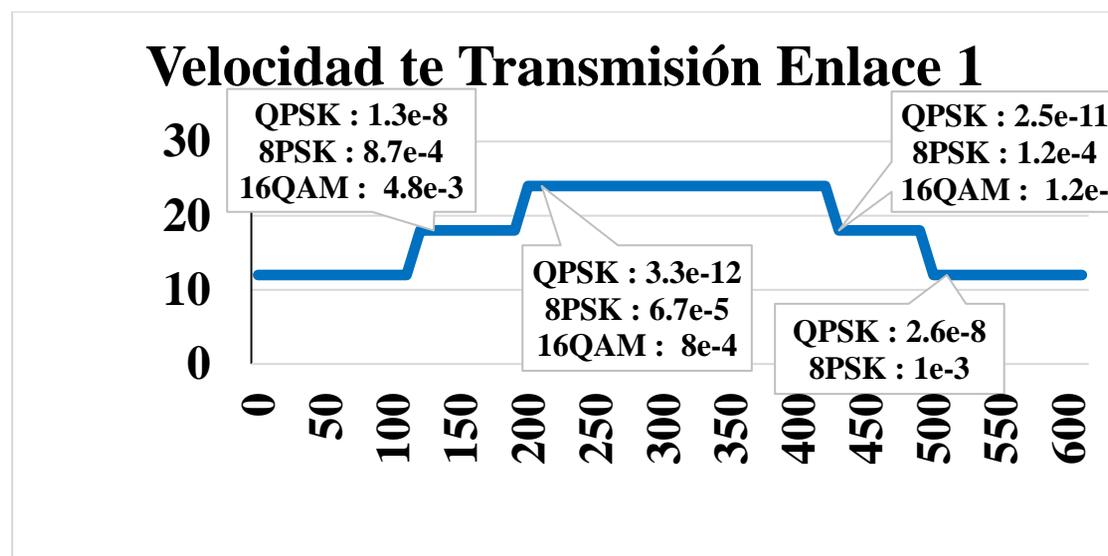


Figura 27 Comportamiento de la velocidad de transmisión durante un evento de línea de vista

En la Figura. 27 podemos apreciar el comportamiento del sistema de transmisión a lo largo del tiempo, donde se observa que el sistema cambia de modulación a lo largo del enlace permitiendo velocidades de transmisión variables. En caso de que el sistema de comunicaciones se hubiese diseñado con un esquema de modulación fijo, y teniendo como limitante la cantidad de errores de 1 bit erróneo por cada 1000 transmitidos, hubiese sido necesaria la selección de la modulación QPSK ya que modulaciones de orden superior no hubiesen cumplido con el parámetro cantidad de errores durante todo el enlace. El sistema de conmutación aprovecha el hecho de que dichas modulaciones si cumplen con el requisito de cantidad de errores permitidos en algunas etapas del enlace, y explota esta característica para obtener mayores velocidades de transmisión manteniendo un límite en cuanto a cantidad de errores.

Para este enlace específico, se obtiene una velocidad de transmisión promedio de 17,9 Mbps, mientras que si se hubiese seleccionado un esquema de modulación fijo se hubiese obtenido una velocidad fija de 12 Mbps, por tanto, concluimos que la aplicación del sistema de conmutación adaptativo tendría un gran impacto en el aprovechamiento de las capacidades de nuestro sistema de comunicaciones, maximizando en este caso la velocidad de transmisión que se puede alcanzar, y por tanto la cantidad de información que se puede transmitir.

2.7 Resumen del Capítulo 2

En este capítulo se realizó un modelo del canal de comunicaciones usando los requerimientos de diseño de la misión presentados en el Capítulo 1. La simulación se desarrolló usando el software STK, y nos permitió determinar las principales afectaciones que sufrirá nuestro sistema, las cuáles definirán las técnicas y parámetros de diseño que serán utilizados en el proceso de diseño e implementación presentado en el Capítulo 3. Se obtuvo una aproximación del comportamiento de los parámetros de transmisión, específicamente el comportamiento de la tasa de bit erróneo, (BER, por sus siglas en inglés). Se evaluó dicho comportamiento para tres tipos de modulaciones (QPSK, 8PSK y 16QAM), determinando el tiempo de utilización que tendrían las mismas en un enlace de duración

promedio, lo que permitió arribar a la conclusión de que dichas modulaciones son adecuadas para cumplir con los requisitos que se tienen para el sistema. Usando los resultados de todo este análisis se propone un esquema de control de la conmutación, el cual queda como trabajo futuro para seguir desarrollando el sistema.

Usando los elementos anteriormente planteados podemos definir las características y funcionalidades que debe tener nuestro diseño. Debemos implementar circuitos digitales que permitan obtener las señales moduladas propuestas, y deben contar con un mecanismo que permita la selección de la modulación deseada. De forma general el sistema contará con una entrada de datos, señales de reloj y reset y una entrada de selección que permita controlar la modulación que se requiera. Entregará a su salida un valor digital el cual, al alimentar un conversor digital analógico, generará las señales moduladas en frecuencia intermedia, para su posterior procesamiento por el módulo de alta frecuencia. Como requisitos adicionales tenemos la necesidad de que el sistema genere internamente varias señales de reloj a partir del reloj externo, esto para la obtención de las señales portadoras a la frecuencia deseada para cada una de las modulaciones. Es necesario además el control del flujo de bits de entrada, ya que el mismo debe ser agrupado en bloques de dimensión variable dependiendo de la modulación seleccionada.

Capítulo 3. Desarrollo del modulador adaptativo

En este capítulo se describe el proceso de diseño, descripción e implementación del modulador adaptativo, auxiliándose de la base teórica y los requerimientos de diseño establecidos en el Capítulo 2

3.1 Lenguaje VHDL, características principales

Para realizar la descripción de nuestro sistema usamos el lenguaje VHDL, cuyas siglas provienen de la combinación entre los términos Circuitos Integrados de Muy Alta Velocidad (VHSIC, por sus siglas en inglés) y Lenguaje de Descripción de Hardware (HDL, por sus siglas en inglés). En [18] se define VHDL como “*un lenguaje orientado o modelado de sistemas digitales*”. Las siglas HDL provienen del inglés Hardware Description Language, mientras que la V proviene de VHSIC (Very High Speed Integrated Circuits). VHDL es un lenguaje que permite la integración de sistemas digitales de diferente complejidad en un solo dispositivo lógico programable. [18]

3.2 Selección de la tecnología FPGA a utilizar

En la actualidad el uso de los FPGA COTS para el desarrollo de sistemas espaciales ha cobrado un gran auge debido al enorme potencial que estos dispositivos presentan como son gran velocidad de procesamiento, capacidad de reconfiguración, flexibilidad y un costo mucho más bajo que sus análogos de calidad militar o espacial. [20]

El elemento básico de un FPGA desde el punto de vista no-lógico, es decir que no tiene una función digital lógica, es la celda de configuración. Esta celda es la que va determinar la configuración de cada elemento lógico, por ejemplo, si un flip-flop se va usar o no. La celda de configuración también determina la configuración de los elementos de ruteo y de las interconexiones. [21]

Existen en la actualidad tres tipos de celdas de configuración de un FPGA. Una está basada en una celda SRAM, que como su nombre indica se usa una pequeña celda SRAM para mantener la configuración de cada parte configurable del FPGA. Otro tipo de celda involucra una celda llamada anti-fuse (anti-fusible), que consiste en una estructura la que, a diferencia de un fusible regular, esta normalmente abierta. Para configurarla se hace circular una cierta cantidad de corriente durante la configuración del dispositivo, cerca de los 5mA, y causa una gran potencia de disipación en un área muy pequeña, lo que provoca el derretimiento de un aislante dieléctrico entre dos electrodos formando una unión permanente muy fina. El tercer tipo de celda de configuración se basa en celdas tipo Flash, que, a diferencia de las SRAM, permiten mantener la configuración aún después de desconectada la alimentación del dispositivo.

En la tabla 16 se realiza un estudio comparativo entre tres dispositivos FPGAs, cada uno de los cuales es representativo de las técnicas de construcción anteriormente mencionadas. Los elementos que se tuvieron en cuenta para realizar la comparación fueron:

- velocidad de reloj
- densidad
- potencia consumida
- tolerancia a la radiación

Se puede observar que, a pesar de ser la peor en cuanto a tolerancia a la radiación y consumo de potencia, la tecnología FPGA SRAM permite alcanzar velocidades de procesamiento muy superiores a las demás alternativas, y la densidad de elementos lógicos en un ship es también mayor. Esto unido a la posibilidad de diseñar sistemas reconfigurables, hacen de la tecnología FPGA SRAM una opción muy atractiva a la hora de usar FPGAs para el diseño de sistemas electrónicos para aplicaciones espaciales.

Tabla 10 Comparación entre tecnologías de fabricación de FPGAs

	SRAM(XCVU440)*	Antifuse(Axcelerator Microsemi)**	Flash(ProASICL Microsemi)***
Velocidad	741 MHz	500 MHz	350 MHz
Potencia	13 W	300 mW	1.2 mW
Densidad	5 065 920 CLB Flip Flop 2 532 960 CLB LUT	2 000 000 internal gates	3 000 000 gates
Tolerancia a radiación	Peor	Mejor	Intermedia

*Fuente: *UltraScale Architecture and Product Overview, Xilinx, ** Microsemi Axcelerator Family FPGAs, *** Microsemi ProASICL Low Power Flash FPGAs*

3.3 FPGAs SRAM

El uso de FPGAs basados en SRAM para aplicaciones de computación en misiones espaciales se ha convertido en un área de investigación muy activa en los últimos tiempos. [22]

Este incremento en la popularidad de dichos dispositivos se debe a su capacidad de implementar complejos circuitos en muy corto tiempo y por su gran versatilidad para implementar gran cantidad de tareas. [23]

Si analizamos el aspecto de resistencia a la radiación, los FPGA SRAM tienen un menor rendimiento que sus pares anti fuse. Aun así, la ventaja que ofrecen los FPGAs SRAM es su posibilidad de reconfiguración, exhibiendo menores costos de desarrollo a pesar de la necesidad de usar una memoria externa no volátil para almacenar la configuración del sistema. Además, esta arquitectura provee la posibilidad de comprar grandes volúmenes de producto a un costo inferior al de los FPGAs anti fuse. [24]

El procesamiento de datos a bordo de un satélite realizado en una plataforma FPGAs SRAM adaptable, ofrece ventajas únicas sobre los FPGAs anti fuse, posibilitando la creación de un instrumento adaptable con ahorros considerables en masa, costo, recursos y flexibilidad. [25]

Por las razones anteriormente expuestas, y basándonos en el análisis mostrado en la Tabla 16, se propone en uso de FPGAs SRAM para el desarrollo de nuestro sistema, específicamente FPGAs del fabricante Xilinx, debido a que existen referencias de misiones espaciales que han usado dispositivos de este fabricante con buenos resultados [22] [23] [24].

3.4 Herramienta Vivado Design Suite HL WebPack

Para realizar la descripción de los elementos de nuestro sistema digital nos auxiliamos de la herramienta Vivado Design Suite HL WebPack, la cual es una versión libre de la plataforma Vivado Design Suite perteneciente a Xilinx, y contiene muchas de las funcionalidades de la versión profesional, aunque se encuentra limitada en cuanto a la cantidad de dispositivos que contienen sus librerías. Algunos de los dispositivos soportados por la plataforma son: Artix®-7 (XC7A15T, XC7A35T, XC7A50T, XC7A75T, XC7A100T, XC7A200T), Kintex®-7 (XC7K70T, XC7K160T), Kintex UltraScale™ (XCKU025 – XCKU035), Zynq®-7000 All Programmable SoC (XC7Z007S – XC7Z7030). [19]

Entre las funcionalidades de la plataforma podemos encontrar:

- Síntesis e implementación del código VHDL
- Ubicación y ruteo de los componentes del sistema
- Generación del bitstream para la programación del dispositivo
- Programador de dispositivo
- Simulación funcional del sistema

3.5 Metodología de diseño jerárquico en VHDL

El diseño jerárquico es una herramienta que nos permite dividir un sistema complejo y extenso en bloques más simples de implementar y simular, para luego unir estos en a través de un algoritmo de integración [18]. Una de las ventajas de este tipo de diseño es que permite el trabajo en paralelo ya que cada programador puede desarrollar ciertos módulos o componentes y una vez listos todos los componentes del sistema se procede a la integración. La metodología recomendada en [18] para desarrollo de sistemas extensos es la siguiente:

1. Analizar con detalle el problema y descomponer en bloques individuales la estructura global
2. Diseñar y programar módulos individuales (componentes)
3. Crear un paquete de componentes
4. Diseñar el programa de alto nivel (Top Level)

En nuestro caso, decidimos por cuestiones de comodidad alterar el orden propuesto en [18], planteando primeramente el bloque Top Level de mayor jerarquía, y descomponiendo el mismo en bloques más pequeños hasta describirlo completamente. Siguiendo esta metodología, el diseño de nuestro Top Level se muestra en la Figura 28. A este nivel solamente definimos los elementos que tendremos como entrada al sistema y lo que esperamos obtener como respuesta a la salida del mismo. Se puede apreciar que en el nivel de abstracción más general nuestro sistema contará con cuatro señales de entrada las cuales son el reloj de entrada (clk), la señal de datos a transmitir (datos), la señal de reset (reset) y por último la señal selección(1:0) la cual es un bus de dos bits y se utiliza para seleccionar la modulación deseada. A la salida del sistema tenemos dos señales: salida_modulada la cual es la señal resultante del proceso de modulación, específicamente la señal modulada, y la señal RDY la cual es una señal de control que indica la estabilización del PLL interno del sistema.

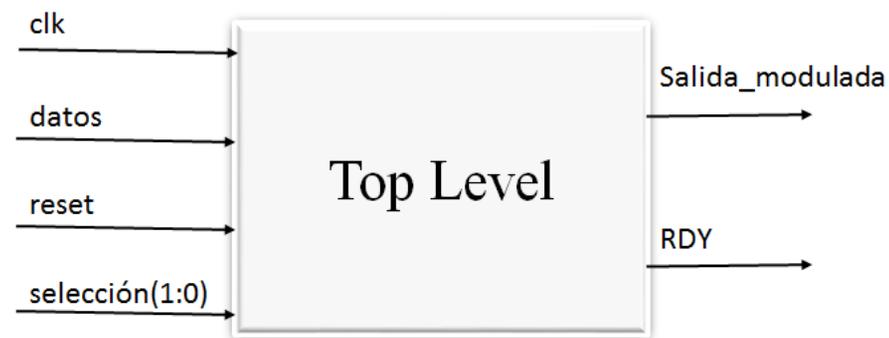


Figura 28 Top Level del sistema digital

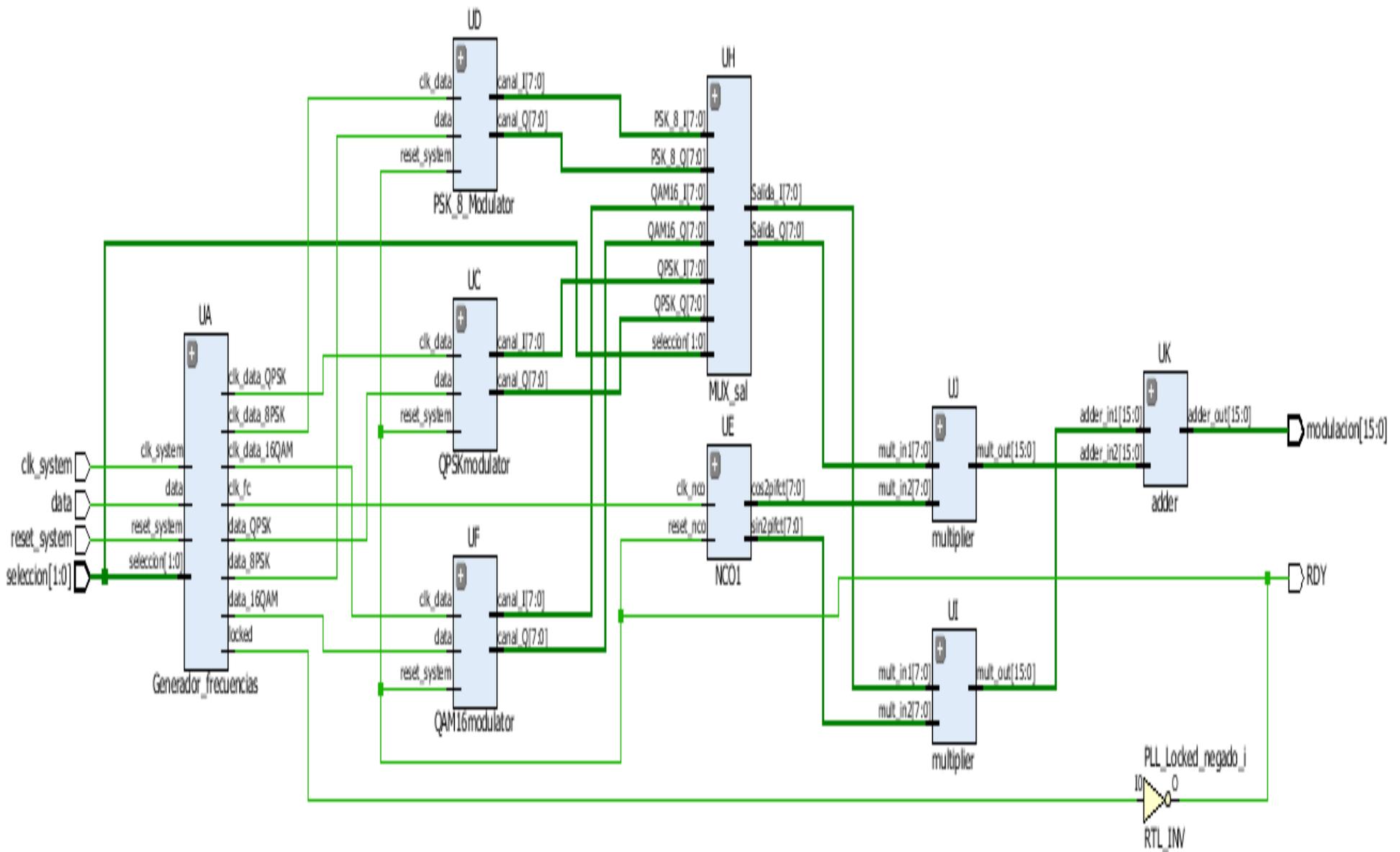


Figura 29 Segundo nivel de abstracción

El siguiente nivel de abstracción se muestra en la Figura 29. Los componentes que forman el sistema son:

1. UA: Generador de frecuencias
2. UC: QPSKmodulator
3. UD: PSK_8_Modulator
4. UF: QAM16modulator
5. UH: MUX_sal
6. UE: NCO1
7. UI, UJ: Multiplier
8. UK: Adder

3.6 Bloque Generador de Frecuencias

Este bloque tiene como objetivo generar todas las señales de reloj que internamente necesita nuestro sistema para su funcionamiento. Debido a que estamos planteando un sistema de modulación adaptativo, esto implica que nuestra fuente de datos tendrá una velocidad que se ajustará a la modulación seleccionada. Específicamente se tendrá tres velocidades, 12, 18 y 24 Mbps, para las modulaciones QPSK, 8PSK y 16QAM respectivamente. Pero ya que el sistema una sola entrada de datos, es necesario multiplexar esta señal para que dependiendo del valor de la entrada **selección** se conecte la señal de datos al bloque correspondiente.

En la Figura 30 podemos ver el diseño Top Level del bloque Generador de Frecuencias

El bloque tiene como entradas la señal de reloj primario del sistema, la señal de datos, el reset y la entrada de selección. A la salida tenemos:

- Las tres señales de reloj de datos para las modulaciones QPSK, 8PSK y 16QAM, `clk_data_QPSK`, `clk_data_8PSK` y `clk_data_16QAM` las cuales tienen una frecuencia de 12, 18 y 24 MHz respectivamente.
- La señal `clk_fc` usada para la generación de la frecuencia portadora
- `PLL_locked` el cual es un pin que indica que el bloque está estabilizado
- Las señales `data_QPSK`, `data_8PSK` y `data_16QAM`, las cuales llevan la señal de datos a los bloques que correspondan para su procesamiento

Podemos apreciar que el bloque Generador de Frecuencias está conformado por dos bloques; PLL y `sel_salida_pll`, los cuales describimos a continuación.

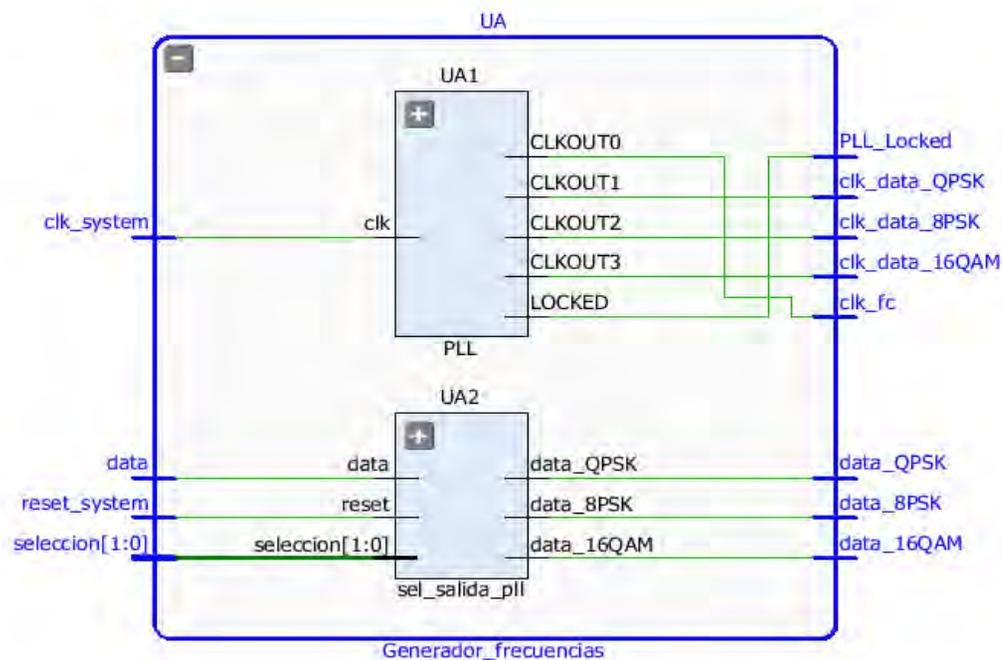


Figura 30 Top Level del bloque generador de frecuencias

3.6.1 Componente sel_salida_pll

El bloque **sel_salida_pll** tiene como funcionalidad el multiplexar la señal de datos dependiendo del valor de la señal de selección, con el objetivo de que solo se active la salida de datos que corresponda a la modulación seleccionada y las otras salidas de datos tengan nivel bajo. La tabla de la verdad del componente se muestra en la Tabla 10.

Tabla 11 Tabla de la verdad del bloque sel_sal_pll

reset	Selección	data_QPSK	data_8PSK	data_16QAM
1	X	0	0	0
0	00	datos	0	0
0	01	0	datos	0
0	10	0	0	datos
0	11	0	0	datos

En la Figura 31 podemos ver la estructura del bloque luego de la implementación.

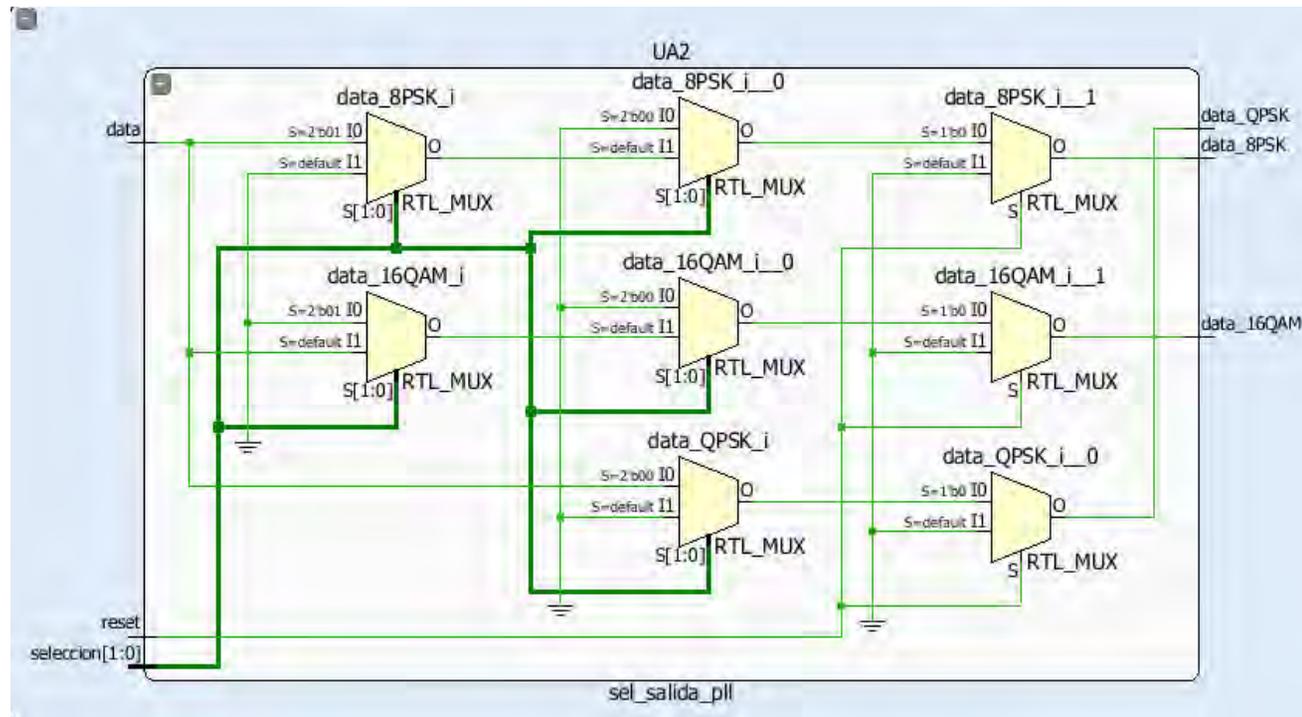


Figura 31 Estructura del bloque sel_salida_pll

El software Vivado toma el lenguaje descriptivo VHDL y lo interpreta construyendo el circuito lógico digital mostrado en la Figura 31, el cual está compuesto por una serie de multiplexores controlados por las señales de **reset** y **selección**.

3.6.2 Componente PLL

Este componente tiene como funcionalidad principal la generación de las señales de reloj que el sistema requiere. Podemos apreciar en la Figura 32 que el bloque tiene como entrada el reloj primario del sistema, y a partir del mismo genera varias señales de reloj. En su interior está compuesto por una instancia del bloque PLL2_BASE, el cual es un PLL que viene integrado dentro del FPGA. Para hacer uso del mismo es necesario instanciar el componente y configurar sus parámetros de forma que se generen las frecuencias que requerimos. En nuestro caso, conectamos la señal de reloj de entrada a la entrada CLKIN1 del PLL, la salida CLKFBOUT es conectada a la entrada CLKFBIN y tomamos las salidas CLKOUT0, CLKOUT1, CLKOUT2 y CLKOUT3 como las señales de reloj de salida, las cuales son mapeadas a las señales clk_fc, clk_data_QPSK, clk_data_8PSK y clk_data_16QAM del bloque Generador de Frecuencias. La señal LOCKED es mapeada a la señal PLL_locked del componente Generador de Frecuencias. Para obtener las frecuencias deseadas es necesario configurar durante el proceso de instancia las constantes CLKFBOUT_MULT y CLKOUTx_DIVIDE de manera que la frecuencia de salida será:

$$CLKOUTx = (CLKIN1 * CLKFBOUT_MULT) / CLKOUTx_DIVIDE$$

Donde x puede variar desde 0 a 5.

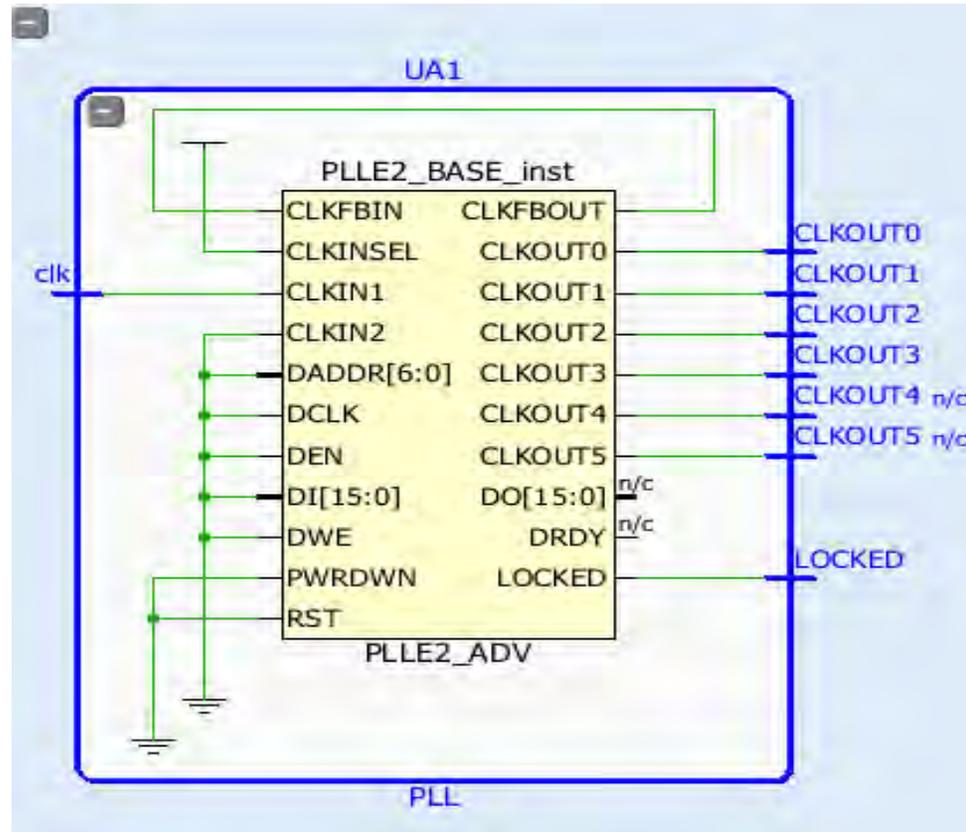


Figura 32 Top Level del componente PLL

3.7 Bloque QPSKModulator

El componente **QPSKModulator** tiene como funcionalidad entregar a su salida los canales I y Q correspondientes a la modulación QPSK, los cuales modularán a las portadoras en cuadratura. La estructura del Top Level del componente se muestra en la Figura 33

Este bloque tiene como entradas la señal de datos a modular, el reloj de datos y el reset, entregando a la salida los canales I y Q.

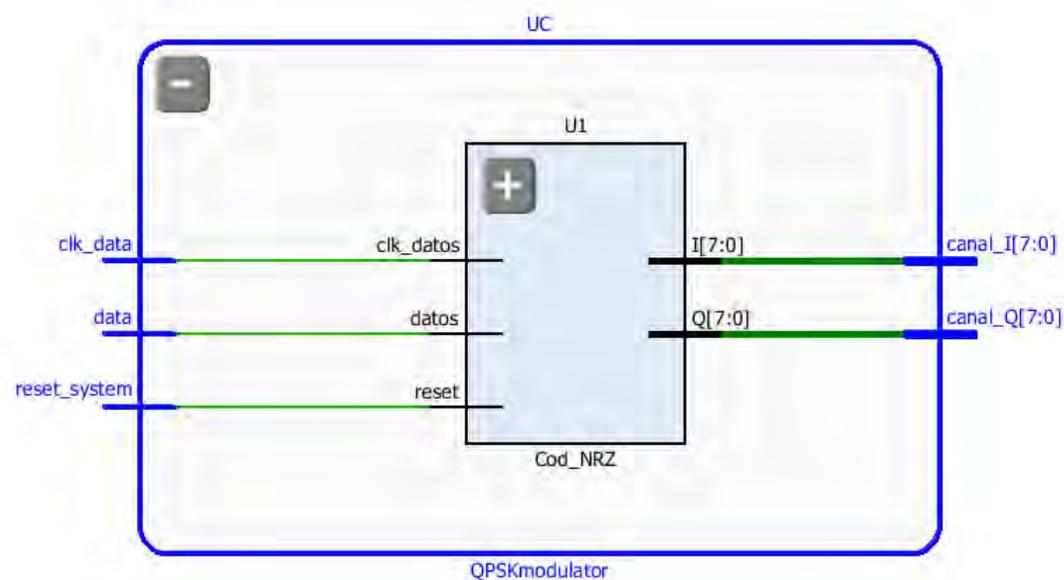


Figura 33 Top Level del componente QPSKmodulator

El siguiente nivel de abstracción del bloque se muestra en la Figura 34

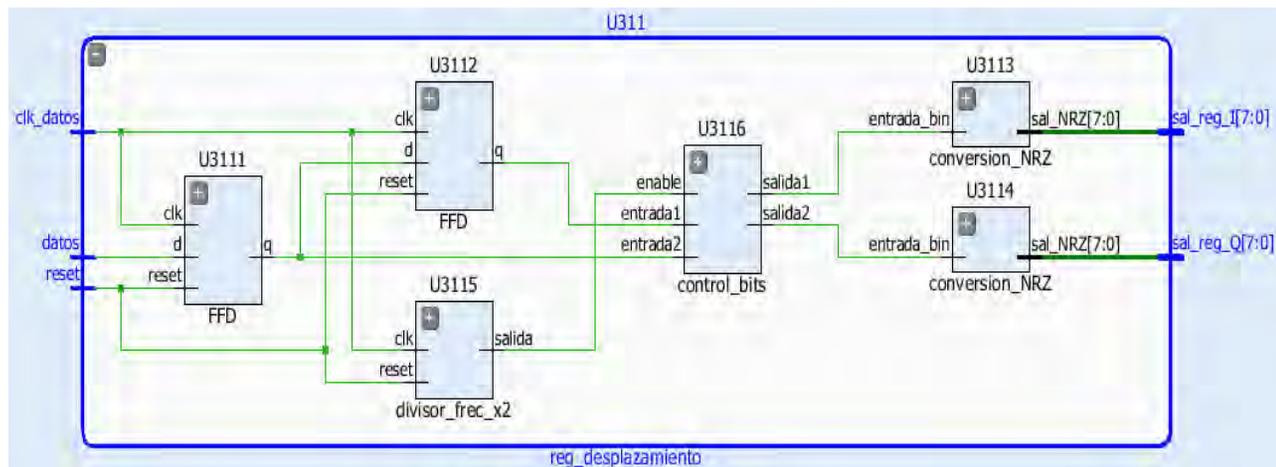


Fig 34 Segundo nivel de abstracción del componente QPSKmodulator

El segundo nivel de abstracción del componente QPSKmodulator está formado por los bloques:

- FDD
- Divisor_frec_x2
- Control_bits
- Conversión_NRZ

3.7.1 Componente FDD

Para la descripción de este componente se usó el lenguaje funcional, y el software Vivado lo interpretó como un flip-flops tipo D. El resultado de la interpretación se muestra en la Figura 35

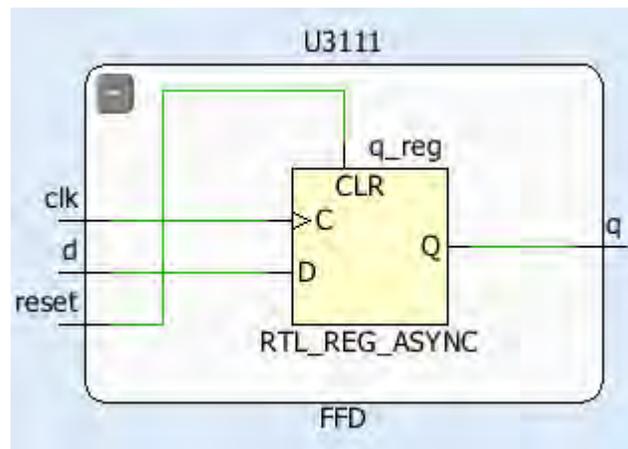


Figura 35 Interpretación del componente FDD por Vivado

Este bloque funciona como un elemento de memoria, ya que mantiene a su salida el valor anterior presente en la entrada. El objetivo de este bloque en nuestro sistema es el de crear un registro de desplazamiento, ya que en el caso de la modulación QPSK es necesario acumular dos bits para generar un símbolo. Para lograr esto se conectaron dos flip-flops tipo D en serie, de manera que, a la entrada del

primero colocamos la señal de datos, mientras que la salida del primero es la entrada del segundo. De esta forma cada dos ciclos de reloj podemos tomar el último par de bits generados.

3.7.2 Componente divisor_frec_x2

El objetivo de este componente es dividir la señal de reloj de datos de entrada por dos, para detectar cuando se hayan generado los bloques de dos bits, y habilitar la generación del símbolo correspondiente. La señal generada a la salida de este componente servirá como señal de reloj para el bloque que realice el cambio de los bits de entrada de serial a paralelo. La interpretación que realizó el software Vivado del componente se muestra en la Figura 36.

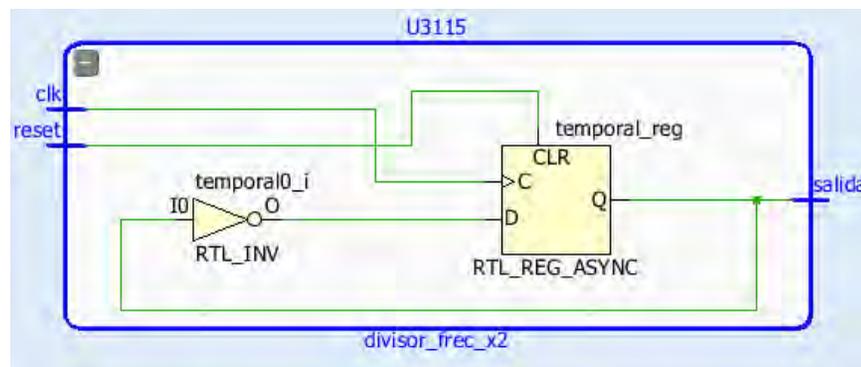


Figura 36 Interpretación del componente divisor_frec_x2 por Vivado

3.7.3 Componente control_bits

Este componente toma los bloques de dos bits provenientes de la red de flip-flops y los pasa a los canales I y Q según corresponda dependiendo de la salida del bloque **divisor_frec_x2**. Sus salidas mantienen la última combinación de dos bits a pesar de que las entradas cambien, y solo cambia su salida con la señal de control proveniente del divisor de frecuencia.

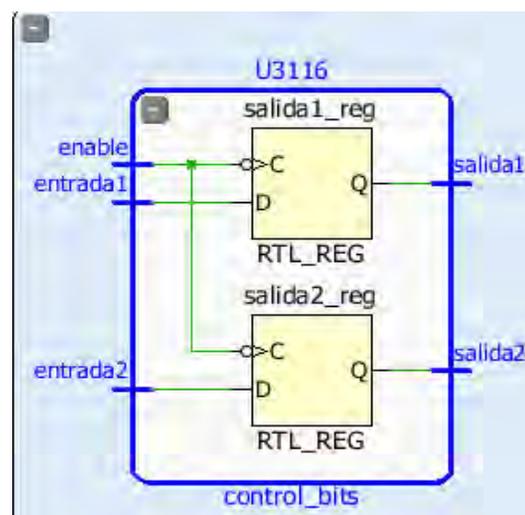


Figura 37 Interpretación del componente control_bits por Vivado

En la Figura 37 podemos ver que este bloque está compuesto por dos flip-flops los cuales tienen entradas independientes y comparten la señal de control. Esto permitirá que la salida de los dos flip-flops se activen al mismo tiempo, entregando a la siguiente etapa la última combinación de dos bits recibida.

3.7.4 Componente conversión_NRZ

Este componente tiene como objetivo pasar de código binario (1 y 0) a código NRZ (1 y -1). Debido a que la siguiente etapa requiere que estos valores sean de 8 bits, realiza la conversión de valor y cantidad de bits al mismo tiempo. La lógica de funcionamiento del bloque se muestra en la Tabla 12.

Tabla 12 Lógica de funcionamiento del bloque conversión_NRZ

Entrada_binaria	Salida_NRZ
0	10000001
1	00000001

En la Figura 38 se muestra la interpretación que realizó el software Vivado del código VHDL con que se describió este bloque

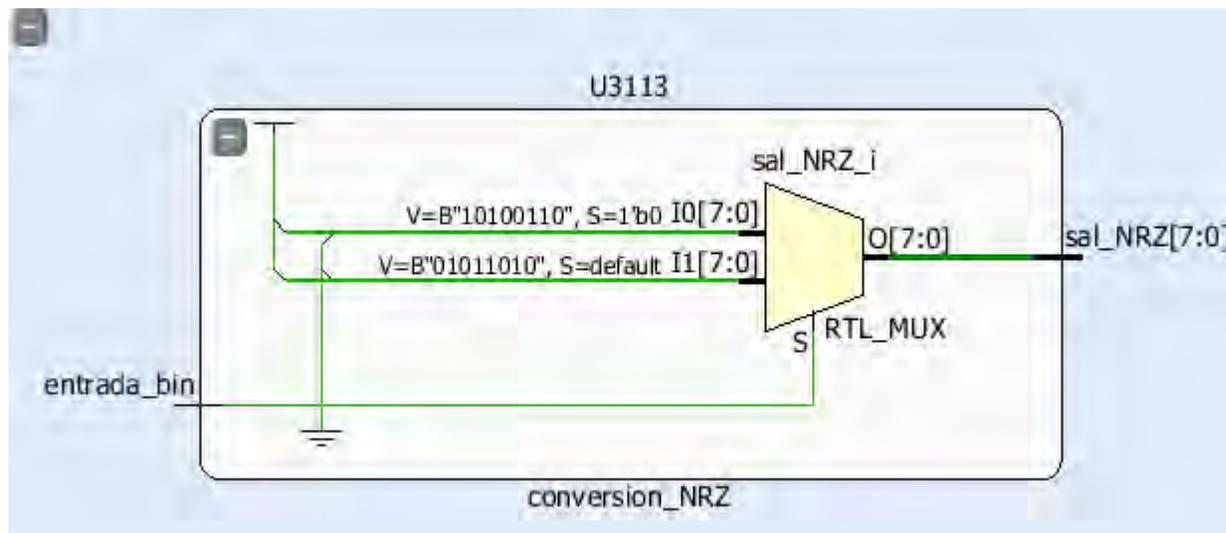


Figura 38 Interpretación del componente conversión_NRZ por Vivado

Podemos ver que el bloque está compuesto por un multiplexor, cuya salida es controlada por el valor de la entrada binaria, y sus entradas tienen asignados valores constantes que corresponden a la conversión de binario a NRZ.

3.8 Componente PSK_8_Modulator

Este es el bloque responsable de generar los canales I y Q necesarios para obtener la modulación 8PSK. La estructura interna del bloque **PSK_8_Modulator** se muestra en la Figura 39. De forma similar a los demás moduladores, tenemos como entradas el reloj de datos, la señal de datos a modular y el reset, mientras que a la salida se generan los canales I y Q, en este caso de la modulación 8PSK.

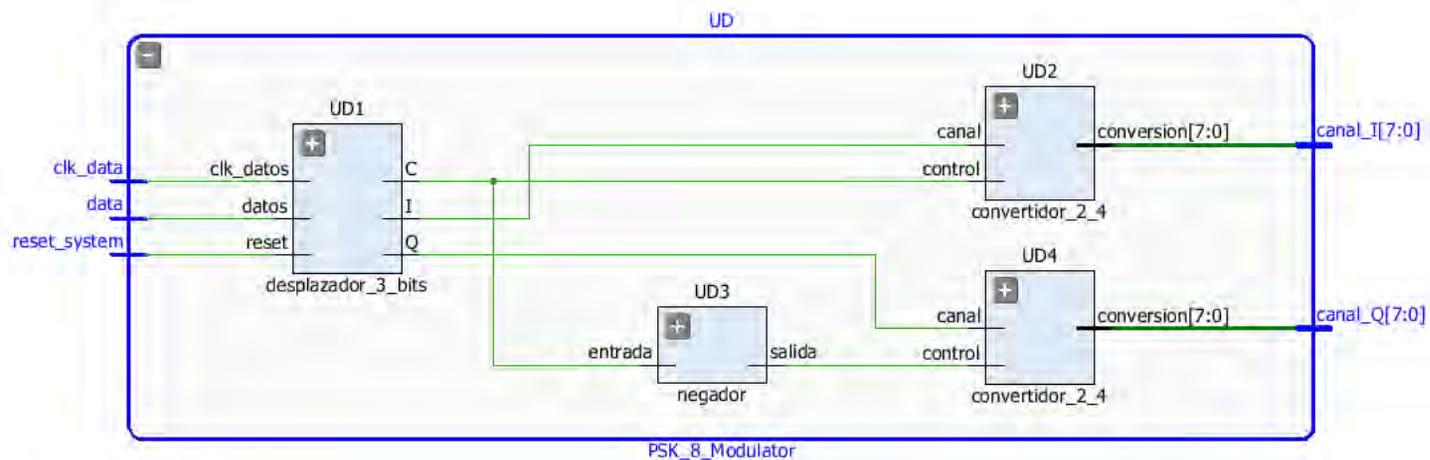


Figura 39 Componentes que conforman el TopLevel del bloque PSK_8_Modulator

El bloque está compuesto por los siguientes elementos:

- Desplazador de 3 bits
- Negador
- Convertidor 2 a 4

3.8.1 Componente desplazador_3_bits

Este bloque tiene como funcionalidad acumular los bits de entrada agrupándolos en bloques de 3 bits, ya que en la modulación 8PSK se genera un símbolo por cada 3 bits. La estructura interna del bloque se muestra en la Figura 40.

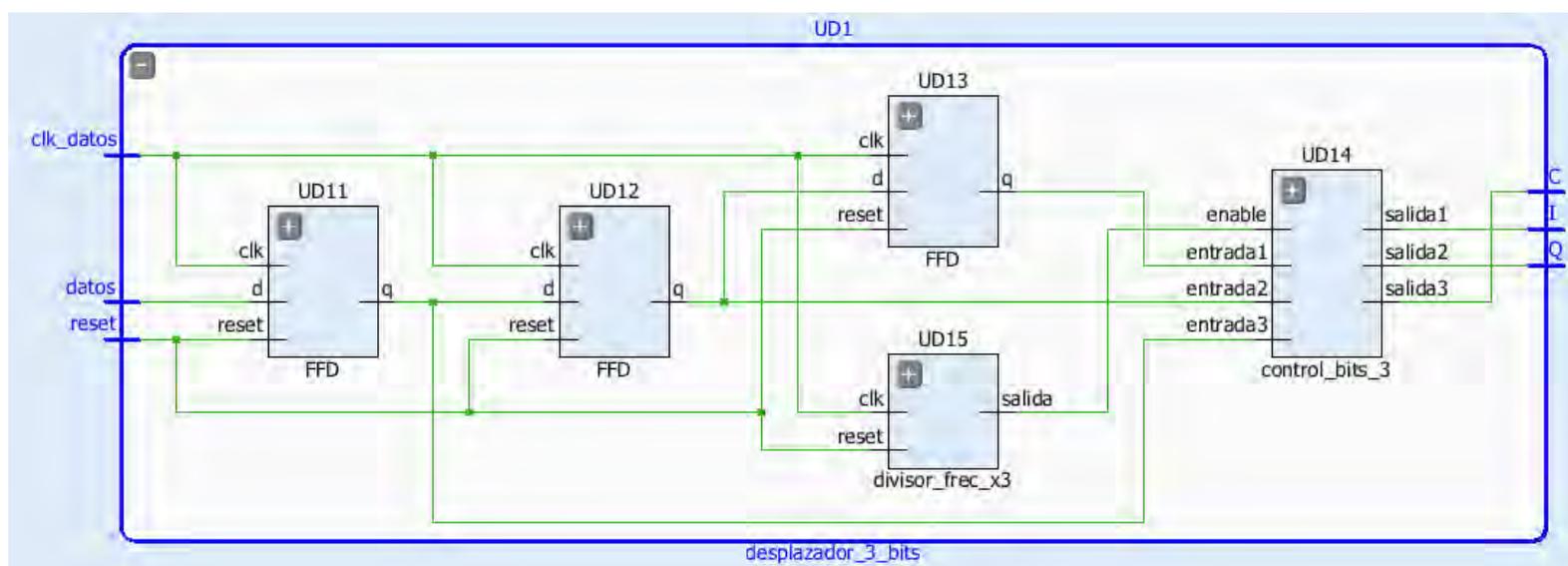


Figura 40 Componentes que conforman el bloque desplazador_3_bits

Podemos apreciar que el bloque cuenta con tres secciones fundamentales; una red de flip-flops tipo D que funcionan como elementos de memoria para almacenar los bits de entrada, un divisor de frecuencia y una etapa de control de la salida. A continuación, describiremos cada una de estas etapas.

En el caso de la red de flip-flops, se utilizó el componente **FFD**. En este caso, ya que es necesario almacenar 3 bits, se necesitan 3 flip-flops, conectados en serie de manera que formen un registro de desplazamiento de 3 bits. De esta forma, la salida Q del flip-flop UD11 corresponde a la salida I del bloque, la salida Q del flip-flop UD12 a la salida Q del bloque y la salida Q del flip-flop UD13 a la salida C.

Para controlar que los 3 bits se presenten a la salida del bloque al mismo tiempo, es necesario realizar un conteo de la cantidad de eventos de reloj de datos y realizar la habilitación de la salida al alcanzarse los 3 bits, por lo cual se utiliza el bloque **divisor_freq_x3**.

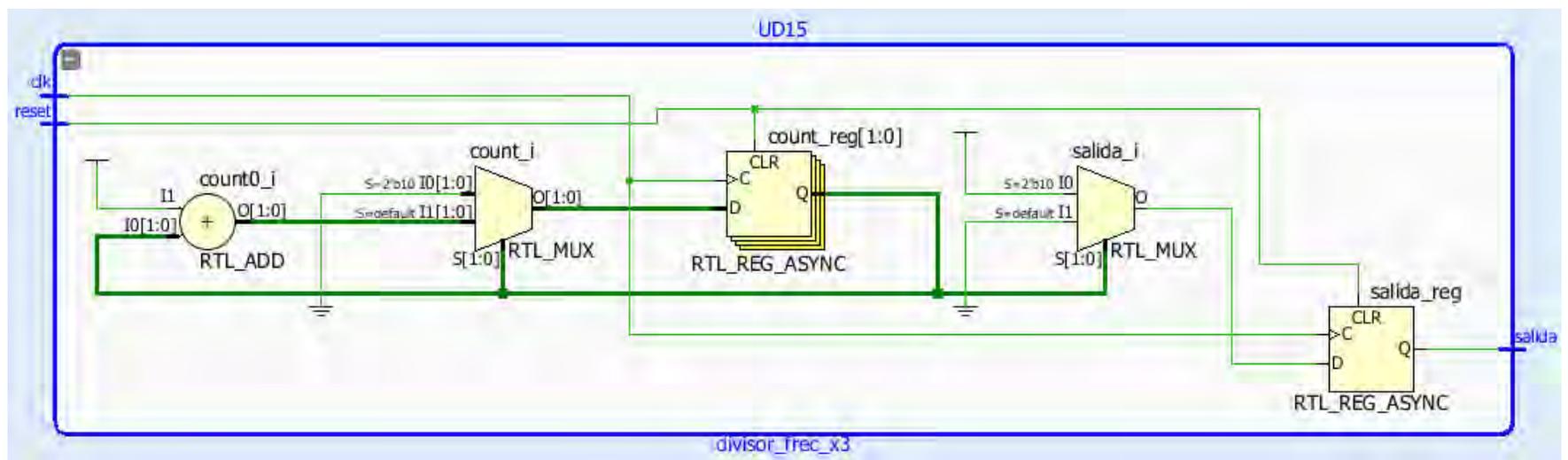


Figura 41 Interpretación del bloque divisor_freq_x3 por Vivado

En la Figura 41, podemos ver la interpretación que realizó el software Vivado de la descripción en VHDL del bloque **divisor_freq_x3**. El bloque contiene una red formada por un sumador, un multiplexor y un acumulador, los cuales realizan el conteo de la cantidad de eventos del reloj de entrada. La salida de este acumulador controla el multiplexor **salida_i**, el cual tiene sus entradas conectadas a valores constantes (0 y 1). La salida de este multiplexor pasa por un registro que funciona como unidad de memoria y su función es mantener el valor generado por el multiplexor a la salida del bloque.

La salida del bloque **divisor_freq_x3** controla la habilitación del bloque **control_bits_3**. La función de este bloque es habilitar la salida cuando se hayan acumulado los 3 bits necesarios para la generación del símbolo. Su estructura se muestra en la Figura 42.

La estructura del bloque **control_bits_3** es muy similar a la descrita en el punto 3.5.3 para el bloque **control_bits**, pero en este caso se necesita un flip-flop adicional ya que se tienen 3 bits a la salida. Este bloque está compuesto por tres flip-flops los cuales tienen entradas independientes y comparten la señal de control. Esto permitirá que la salida de los tres flip-flops se activen al mismo tiempo, entregando a la siguiente etapa la última combinación de tres bits recibida. Cada vez que entra un nuevo bit de forma serial este se almacena en el último flip flop, y el siguiente bit provoca el corrimiento de los bits anteriores. De esta forma se transforma el flujo de bits de entrada serial a un formato paralelo donde los bits están agrupados en bloques de 3 bits o tribits.

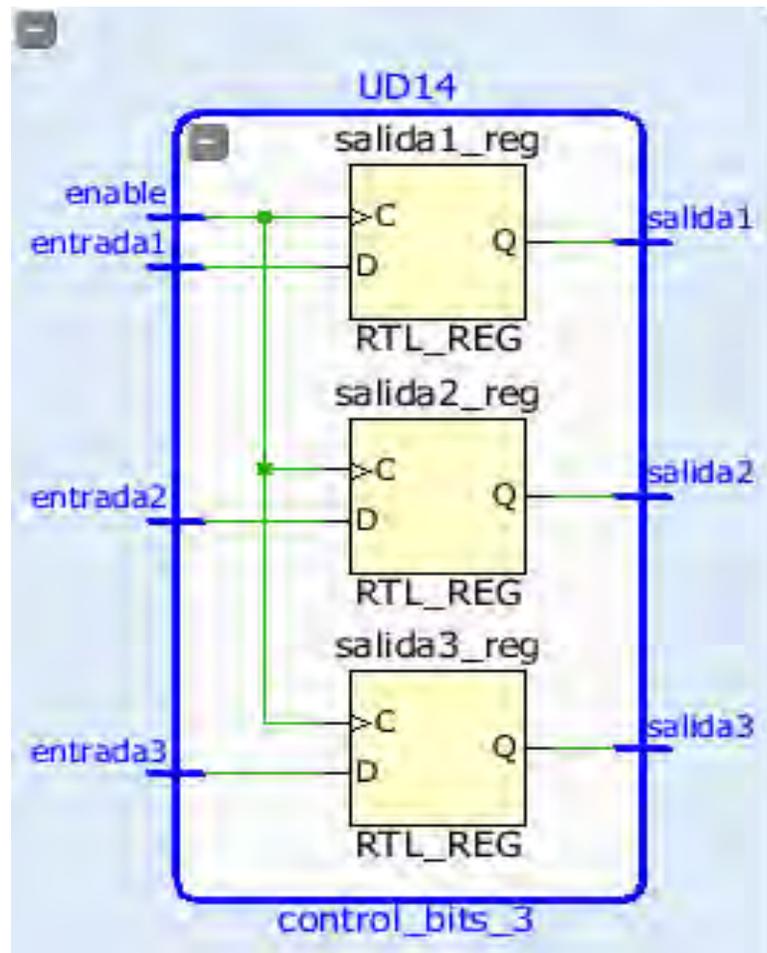


Figura 42 Estructura del componente control_bits_3

3.8.2 Componente convertidor 2 a 4

En el caso de la modulación 8PSK es necesario el uso del bloque **convertidor_2_4**, para realizar la conversión de los bits que se encuentran en código binario, a ciertos niveles y polaridades. El bloque toma dos bits, uno de los cuales controla la polaridad de la conversión (positivo o negativo) y el otro controla la amplitud de la conversión (mayor o menor). El comportamiento del bloque se muestra en la Tabla 13. A pesar de que estamos trabajando en un dispositivo digital, y la conversión realizada por el bloque transforma un valor digital en otro, el proceso realizado por este bloque se puede interpretar de manera funcional como una conversión de digital a analógico.

Tabla 13 Lógica de funcionamiento del bloque conversión 2 a 4

canal	control	salida
0	0	-0.91 (10001100)
0	1	0.91 (00001100)
1	0	-0.38 (11010000)
1	1	0.38 (01010000)

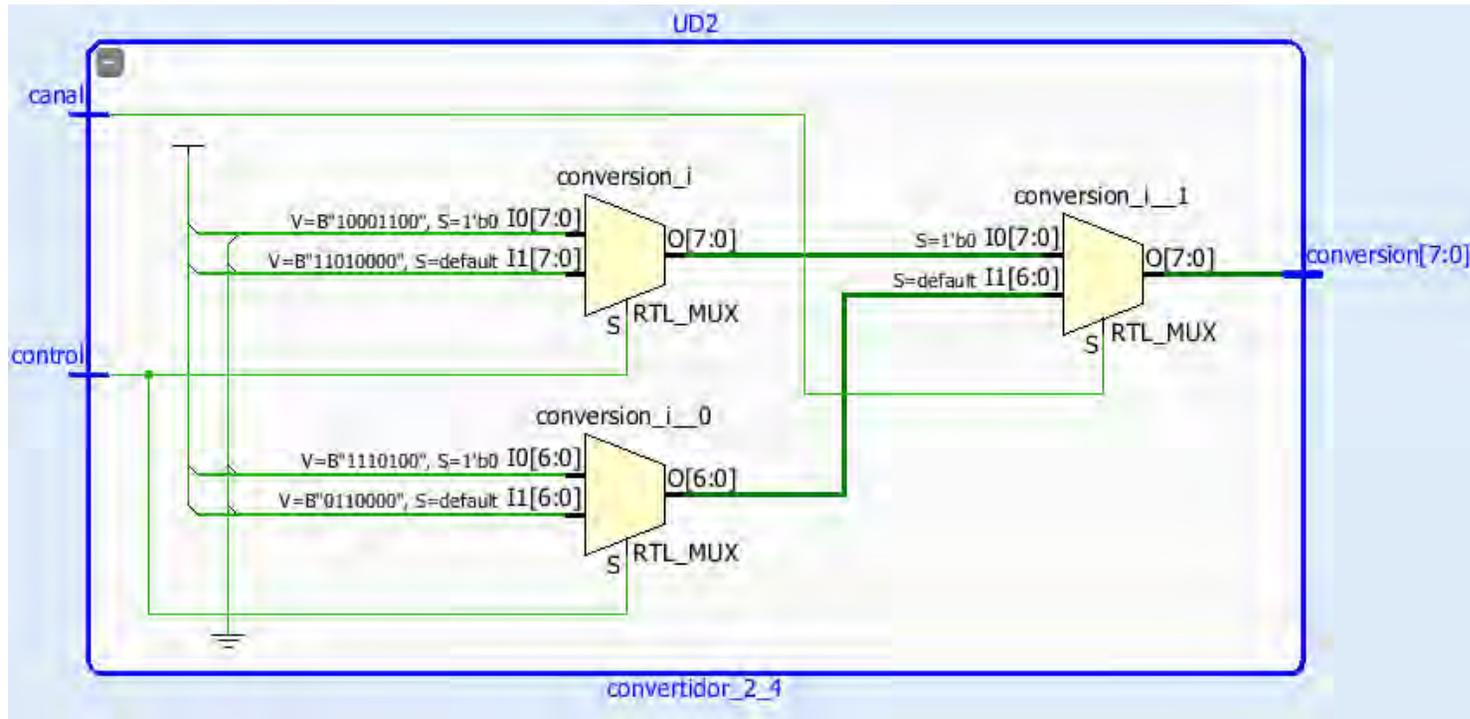


Figura 43 Estructura interna del bloque conversión_2_4

En la Figura 43, podemos ver como el software Vivado interpretó el código VHDL con el que se describió la lógica mostrada en la Tabla 3.3. El bloque está compuesto por una red de multiplexores. Los multiplexores **conversión_i** y **conversión_i_0**, tienen fijadas sus entradas a los valores mostrados en la Tabla 12, y su entrada de selección es controlada por el bit control. Las salidas de estos multiplexores van conectadas a las entradas de otro multiplexor, cuya entrada de selección es controlada por el bit canal. El sistema cuenta con dos de estos conversores, uno para el canal I y otro para el canal Q, la diferencia entre la conexión de los dos bloques es que uno recibe el bit de control o “C” proveniente del bloque **desplazador_3_bits**, mientras que el otro bloque recibe este valor negado. Para esto se usa el bloque **negador**.

3.8.3 Componente negador

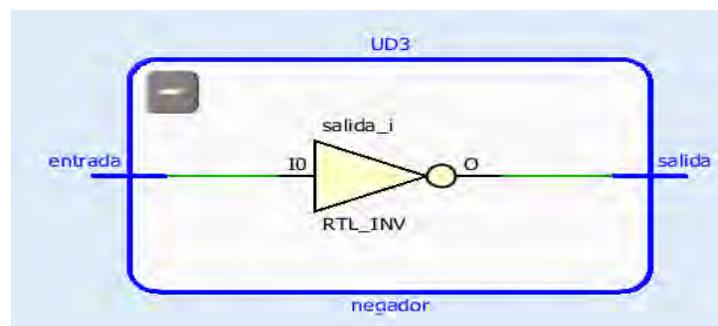


Figura 44 Estructura interna del bloque negador

Este bloque es implementado por el software Vivado como un negador, el cual tiene como salida el valor de la entrada invertido, como se muestra en la Figura 44.

3.9 Componente QAM16modulator

Este componente tiene como funcionalidad obtener los canales I y Q para realizar la modulación 16QAM. La estructura interna del Top Level del componente se muestra en la Figura 45.

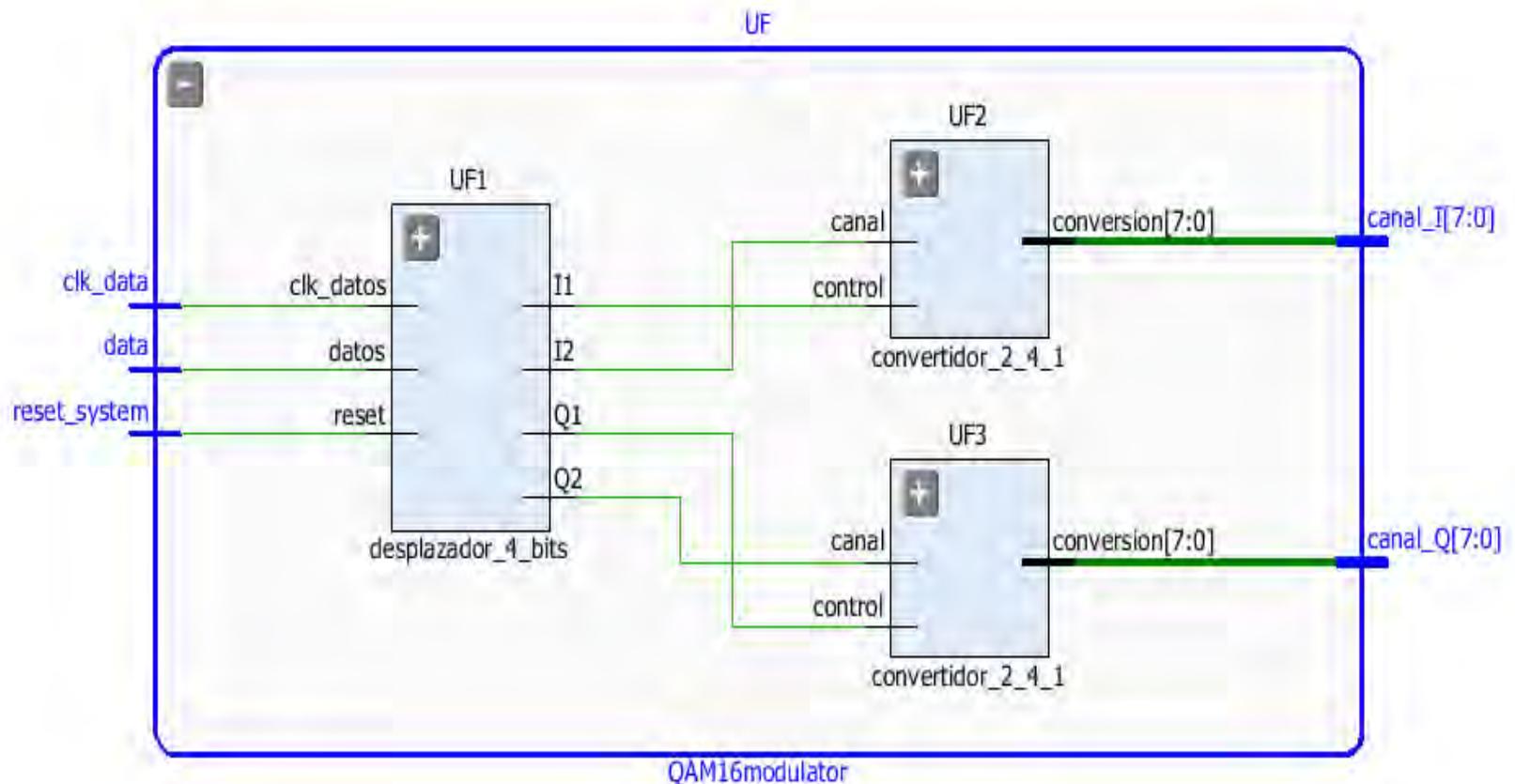


Figura 45 Top level del bloque QAM16modulator

Al igual que los demás moduladores, este componente tiene como entradas la señal de reloj de datos, los datos a modular y el reset, entregando a la salida los canales I y Q, en este caso para realizar la modulación 16QAM. A continuación, describiremos los elementos que componen el bloque.

3.9.1 Componente desplazador_4_bits

Este componente tiene como funcionalidad acumular los bits de entrada en bloques de 4 bits, su estructura se muestra en la Figura 46. En este caso tenemos una red de flip-flops generada con el componente **FFD**, la cual forma un registro de desplazamiento de 4 bits. Cada una de las salidas Q de los flip-flops corresponde a una de las salidas del bloque **desplazador_4_bits**, como se muestra en la Tabla

Tabla 14 Correspondencia entre las salidas de la red de flip-flops y el bloque desplazador_4_bits

Flip-Flop	desplazador_4_bits
UF11	Q1
UF12	Q2
UF13	I2
UF14	I1

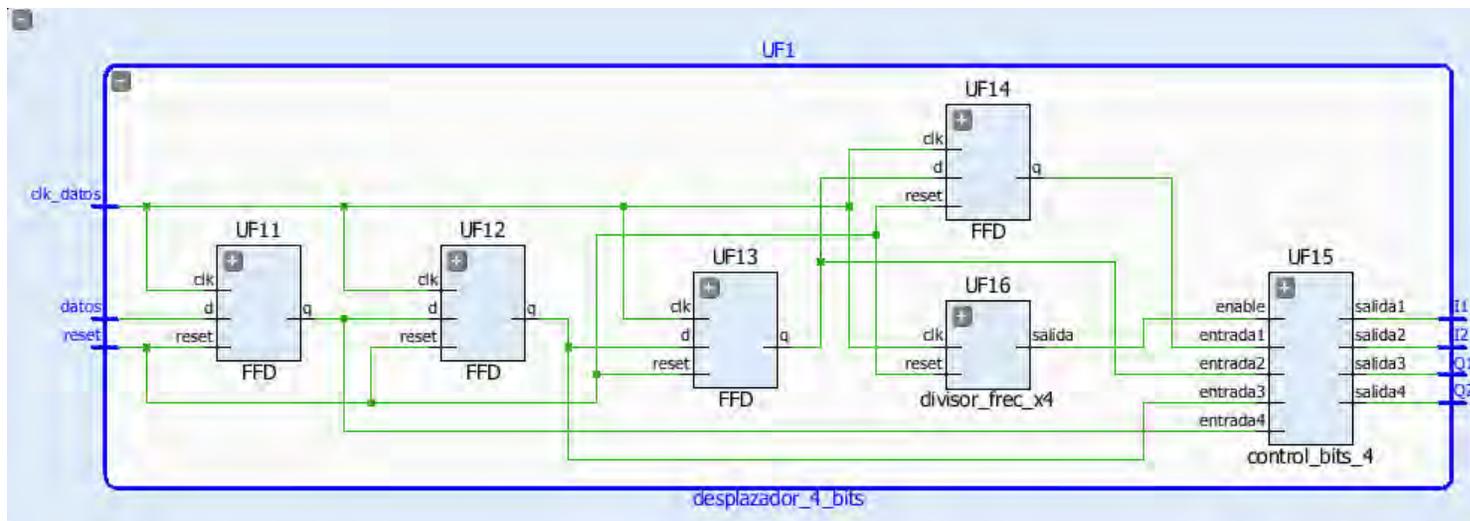


Figura 46 Estructura interna del bloque desplazador_4_bits

3.9.2 Componente divisor_frec_x4

Este bloque controla la habilitación de las salidas del bloque control_bits_4, para lograrlo, realiza un conteo de la cantidad de eventos del reloj de datos, activando las salidas cuando se han acumulado 4 bits. La estructura del bloque se muestra en la Figura 47.

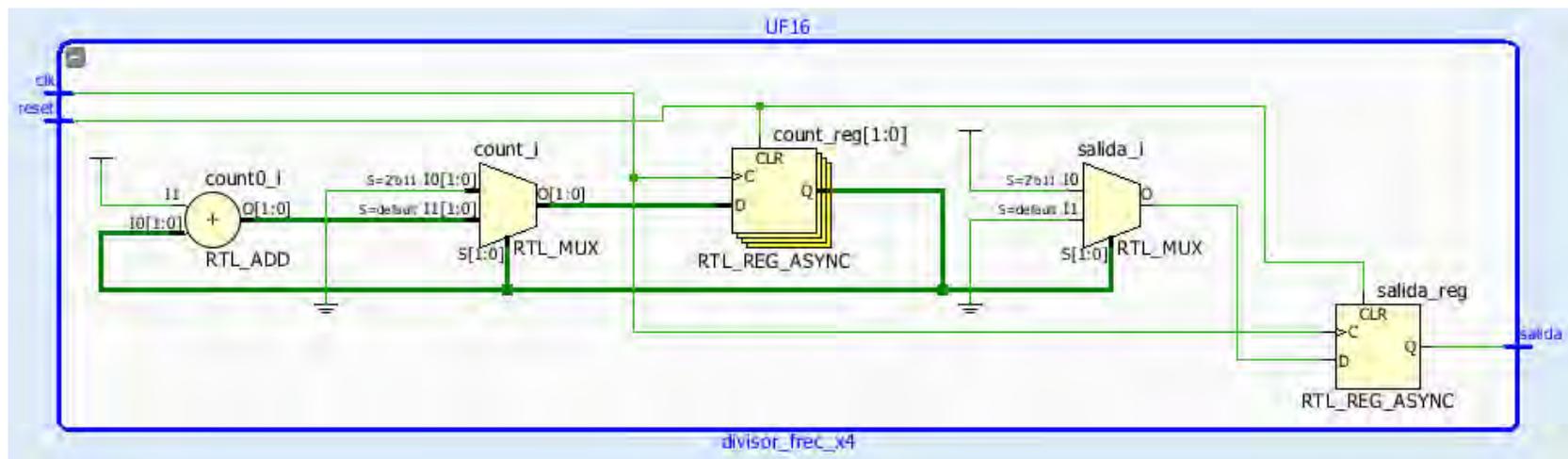


Figura 47 Estructura interna del bloque divisor_frec_4

La estructura interna del bloque **divisor_frec_x4** se muestra en la Figura 46. El bloque contiene una red formada por un sumador, un multiplexor y un acumulador, los cuales realizan el conteo de la cantidad de eventos del reloj de entrada. La salida de este acumulador controla el multiplexor **salida_i**, el cual tiene sus entradas conectadas a valores constantes (0 y 1). La salida de este multiplexor pasa por un registro que funciona como unidad de memoria y su función es mantener el valor generado por el multiplexor a la salida del bloque. Debemos notar que en el **divisor_frec_x3** la salida del multiplexor **salida_i** se activa con el valor de control “10”, mientras que en este caso se activa con el valor “11”.

3.9.3 Componente convertidor_2_4_1

En el caso de la modulación 16QAM también es necesaria la conversión de niveles, en este caso la lógica de conversión se muestra en la Tabla 15

Tabla 15 Lógica de funcionamiento del bloque conversión_2_4_1

canal	control	
	0	1
0	-0,22 (11100100)	-0,82 (10011000)
1	0,82 (00011000)	0,22 (01100100)

La interpretación que realiza el software Vivado del código VHDL que describe la lógica mostrada en la Tabla 14, se muestra en la Figura 48. El bloque está compuesto por una red de multiplexores. Los multiplexores **conversión_i** y **conversión_i_0**, tienen fijadas sus entradas a los valores mostrados en la Tabla 14, y su entrada de selección es controlada por el bit control. Las salidas de estos multiplexores van conectadas a las entradas de otro multiplexor, cuya entrada de selección es controlada por el bit canal. El sistema cuenta con dos de estos conversores, uno para el canal I y otro para el canal Q, en este caso los bits de información son I y Q mientras que los bits de control para la salida del multiplexor final son I' y Q'.

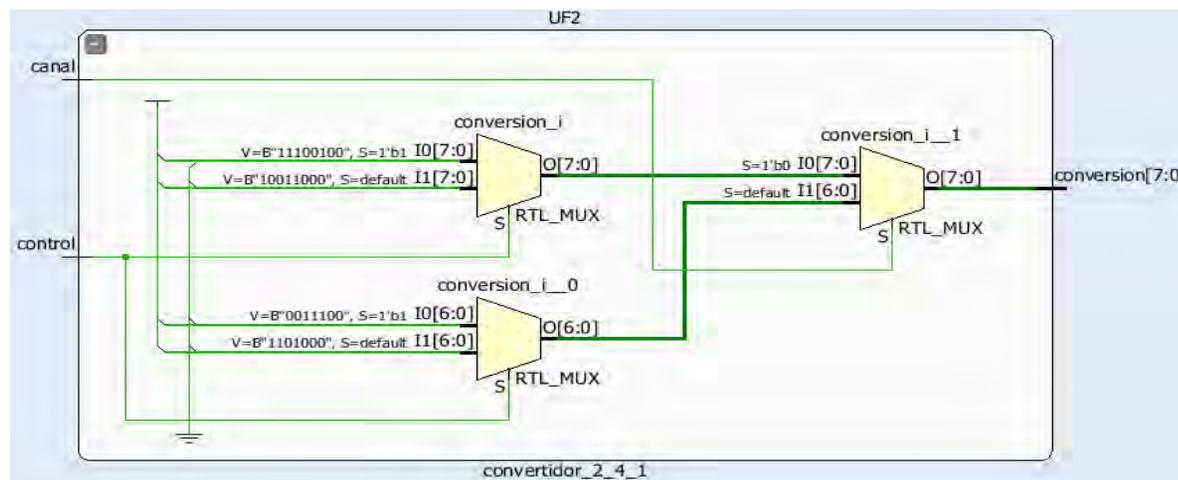


Figura 48 Estructura interna del bloque convertidor_2_4_1

3.10 Componente MUX_sal

Este componente tiene como funcionalidad seleccionar la pareja de canales I y Q que corresponda a la señal de selección. La lógica de funcionamiento del bloque se muestra en la Tabla 16

Tabla 16 Lógica de funcionamiento del bloque MUX_sal

Selección (1 downto 0)	Modulación
00	QPSK
01	8PSK
10	16QAM
11	16QAM

La interpretación que realiza el software Vivado del código VHDL generado para describir la lógica mostrada en la Tabla 15 se puede ver en la Figura 49.

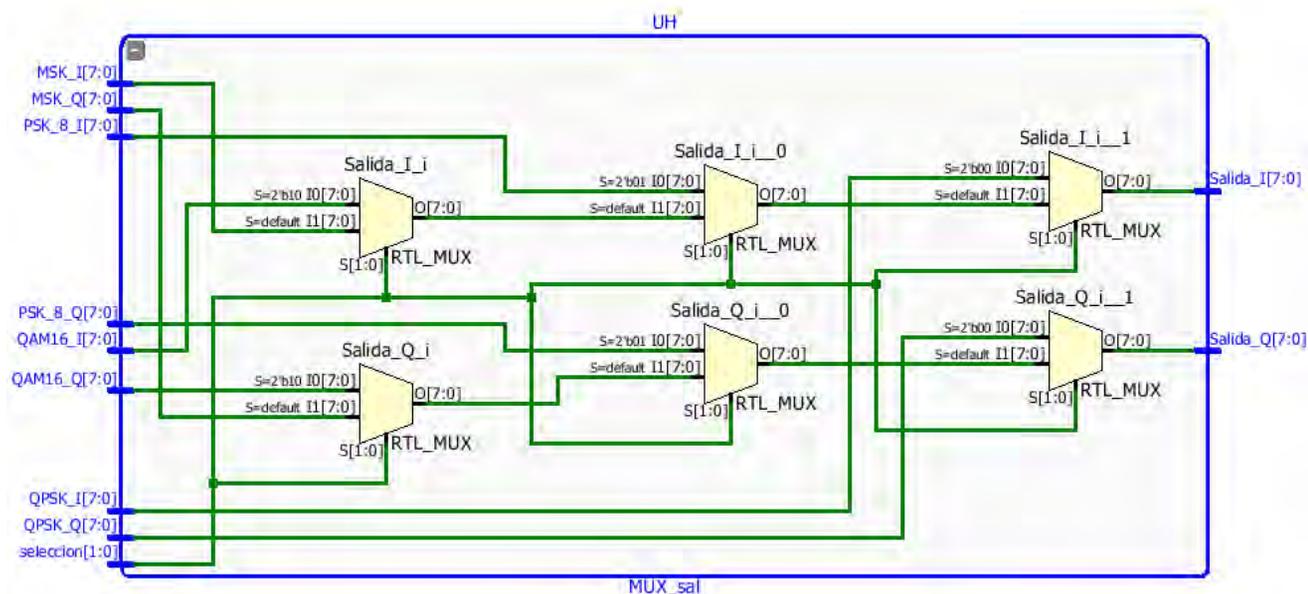


Figura 49 Estructura interna del bloque MUX_sal

Podemos ver en la Figura 49 que el bloque está compuesto por una red de multiplexores, controlados por la señal **selección**, y conectados de forma que se cumpla con la lógica mostrada en la Tabla 16.

3.11 Componente NCO1

Este componente tiene como funcionalidad generar las señales portadoras que se usarán en el proceso de modulación. En el caso que nos ocupa, se generan una señal senoidal y una cosenoidal, para los canales I y Q respectivamente.

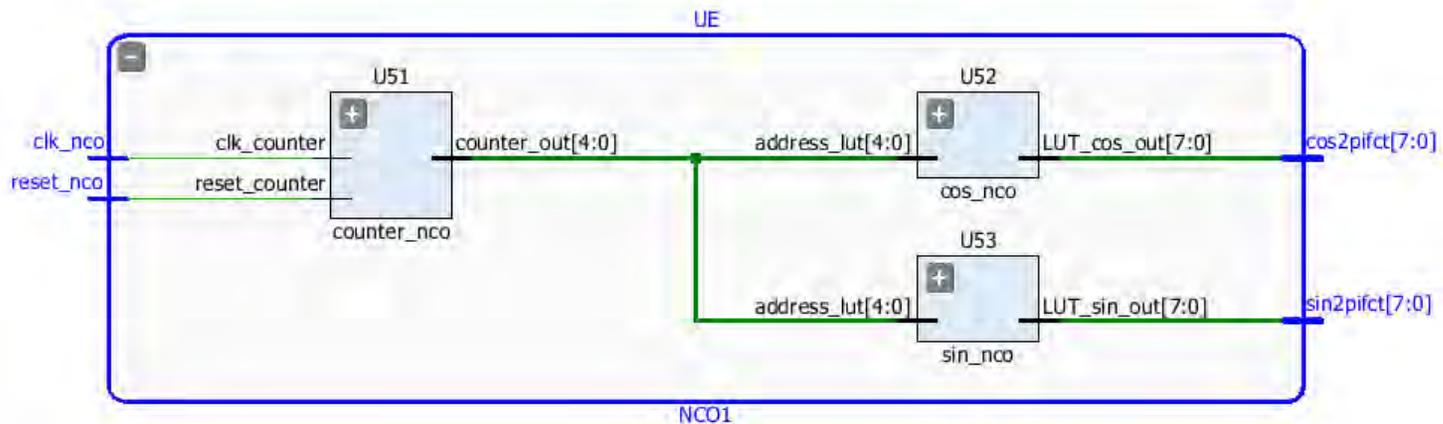


Figura 50 Estructura interna del bloque NCO1

En la Figura 50 se muestra la estructura interna del componente NCO1. Este componente funciona como un oscilador, el cual es controlado por un contador, lo cual lo hace muy flexible. El contador controla el valor de salida de las tablas de búsqueda, las cuáles contienen los valores digitalizados que conforman las señales seno y coseno, respectivamente. La señal de entrada **clk_nco** se usa como señal de reloj del bloque contador, el cual tiene a su salida el valor de conteo. Dicho valor de conteo es usado como dirección dentro de las tablas de búsqueda, las cuáles según la dirección que se encuentre en el puerto **address_lut** seleccionan el valor de salida correspondiente.

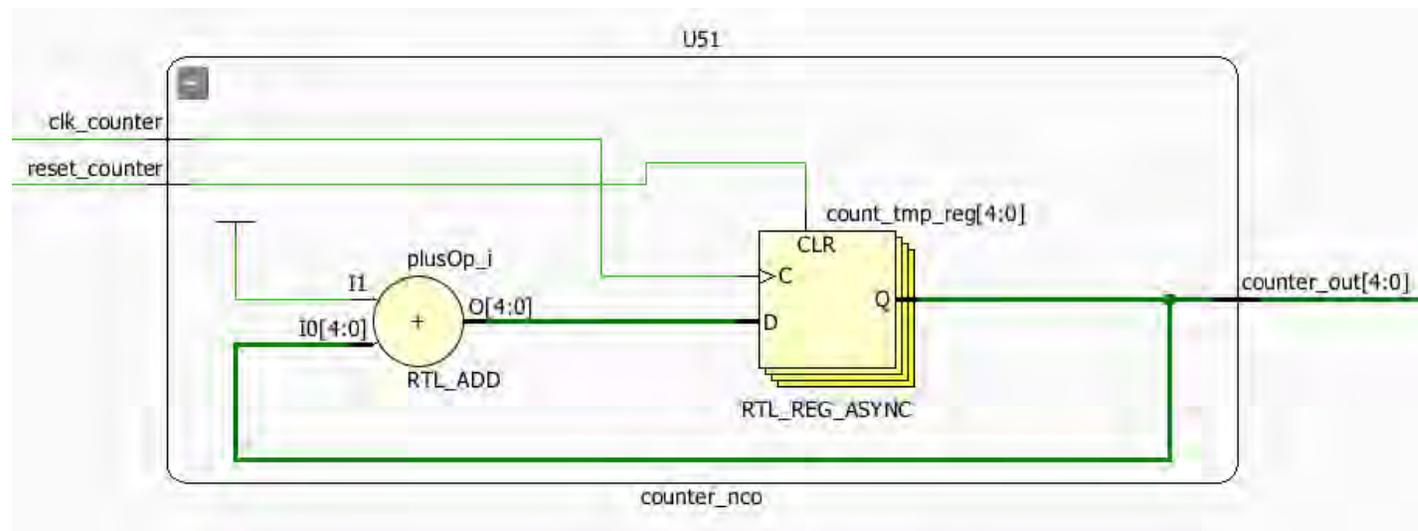


Figura 51 Estructura interna del counter_nco

En la Figura 51 se muestra la estructura del bloque **counter_nco**. Este cuenta con un flip-flop tipo D que almacena el valor de conteo y un sumador que realiza la operación de incrementar el valor de conteo cada vez que ocurre un se tiene un flanco de subida en la señal **clk_counter**.

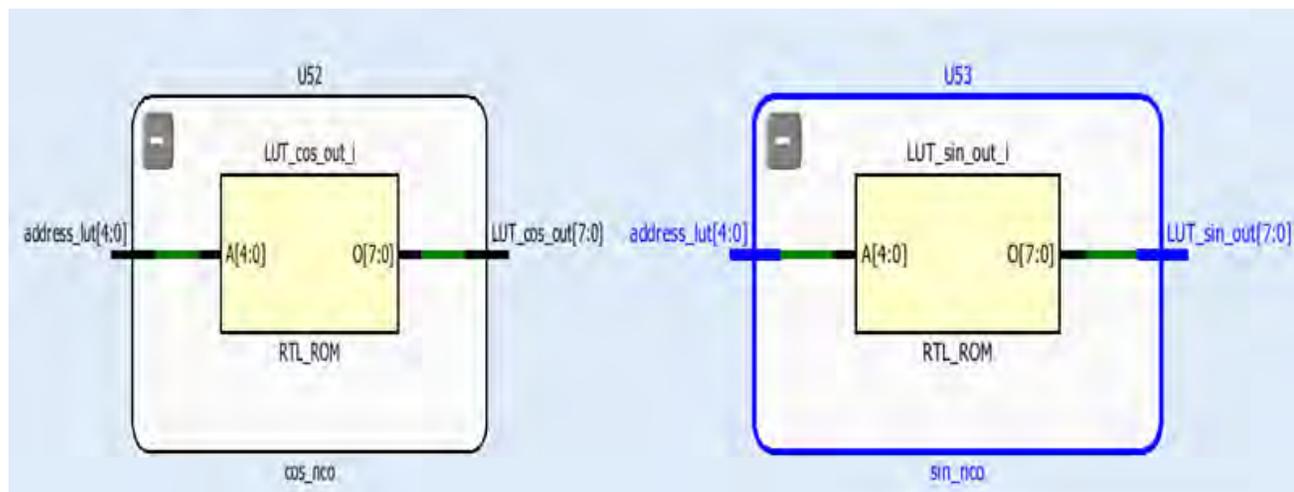


Figura 52 Composición de las tablas de búsqueda

En la Figura 52 vemos como el software interpreta el código VHDL que describe las tablas de búsqueda. Su interior está compuesto por un bloque de memoria ROM embebido dentro del FPGA, donde se encuentran almacenados los valores digitalizados que componen la señal que queremos generar.

El proceso final para obtener la modulación es multiplicar los canales I y Q por las señales portadoras seno y coseno, respectivamente, para posteriormente combinar estas dos señales mediante un sumador.

3.12 Componente multiplier

Este bloque realiza la multiplicación de dos señales de 8 bits. En la Figura 53 podemos ver la estructura interna del mismo, la cual consiste en la implementación de un multiplicador, el cual es un componente con el que cuenta el software Vivado, y es el resultado de la interpretación del código VHDLK diseñado para describir un multiplicador.

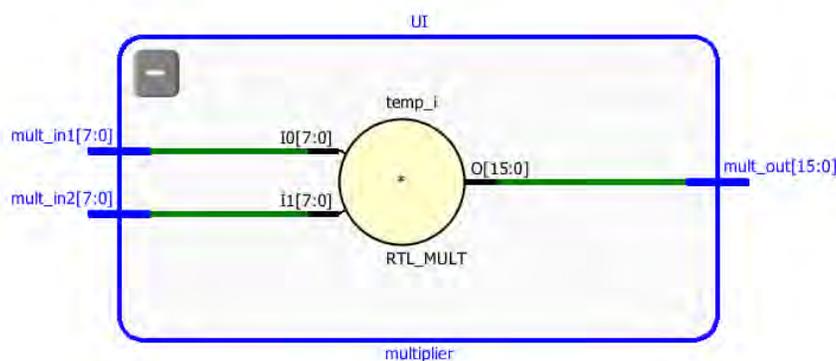


Figura 53 Estructura del componente multiplier

3.13 Componente adder

El componente final de nuestro diseño digital es el sumador, el cual combina los dos canales I y Q. EN la Figura 54 se muestra la interpretación que realizó el software Vivado del código VHDL usado para describir el mismo. Debido a que la descripción en VHDL de este componente se realizó de forma funcional, el software Vivado al realizar el proceso de síntesis usa un componente previamente definido en sus librerías, el cual es un sumador de 16 bits.

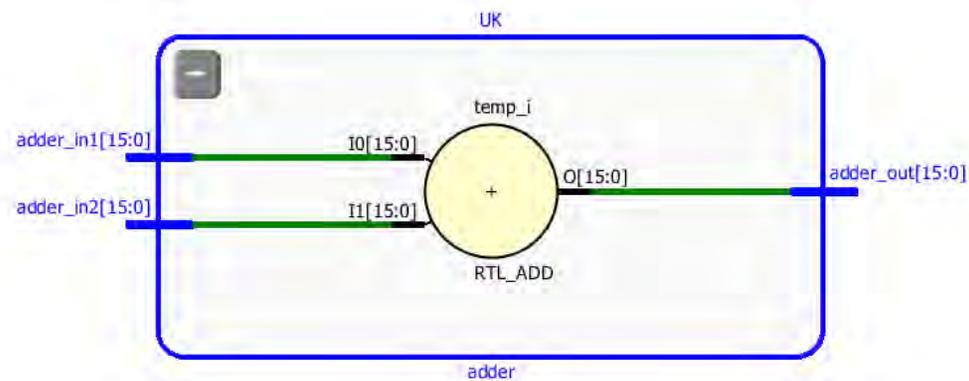


Figura 54 Estructura del componente adder

3.14 Resumen Capítulo 3

En el capítulo 3 se presentó el desarrollo del modulador adaptativo usando el software Vivado y el lenguaje VHDL. Se utilizó la técnica de diseño Top Down, separando un bloque general con funcionalidades muy abarcadoras en bloques de complejidad inferior con funcionalidades más específicas. Durante el capítulo se describe el proceso de diseño y descripción de cada uno de estos bloques. Este diseño se toma como el sistema base, pero el mismo no cuenta con tolerancia a fallas, por lo que en el capítulo siguiente se realiza una implementación de tolerancia a fallas a modo de ejemplo, usando el sistema planteado en este capítulo.

Capítulo 4. Aplicación de técnicas de tolerancia a fallas

En este capítulo se realiza un análisis del efecto que tienen las partículas de alta energía presentes en el ambiente espacial sobre los bloques sensibles del FPGA SRAM, y se estudia de forma documental las técnicas de tolerancia a fallas más utilizadas en sistemas espaciales, con el objetivo de brindar protección contra efectos indeseables como los SEU. Se realiza un análisis comparativo entre varias herramientas que permiten la implementación de sistemas tolerantes a fallas de forma automática, y se selecciona una de ellas para realizar un ejemplo demostrativo de su uso.

4.3 Caracterización de los ambientes radioactivos. Daños que estos producen en los FPGAs SRAM

La actividad del sol genera una gran variedad de partículas de alta energía. Estas pueden ser clasificadas en dos tipos principales, el primero partículas cargadas como electrones, protones e iones pesados y le segundo radiación electromagnética en forma de fotones, que puede ser rayos-x, rayos gamma o luz ultravioleta. La fuente principal de partículas cargadas que contribuyen en los efectos de la radiación son protones y electrones atrapados en el cinturón de Van Allen, iones pesados atrapados en la magnetosfera y erupciones solares. Las partículas cargadas interactúan con los átomos del silicón causando excitación e ionización de los electrones. [4]

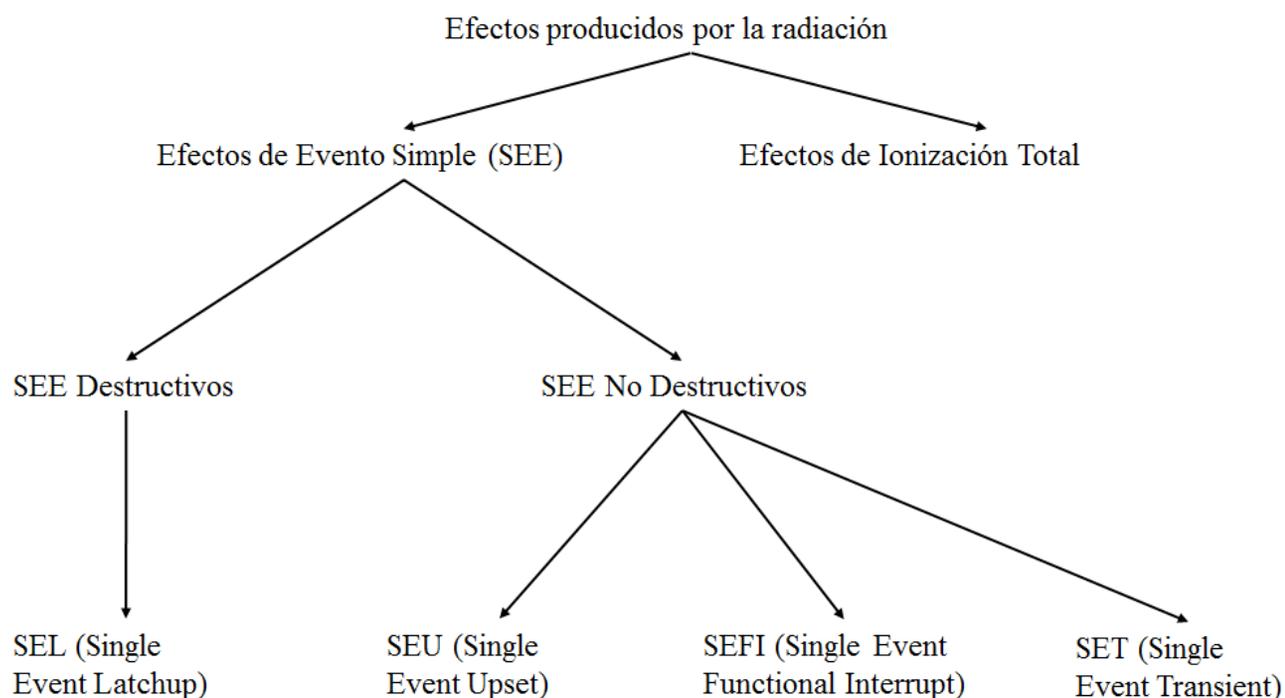


Figura 55 Clasificación de los efectos producidos por la radiación en los FPGA SRAM Adaptado de [20]

Los posibles daños debidos a las radiaciones se pueden separar en dos categorías, tal como se muestra en la Figura 54: Los efectos de un evento simple (SEE, por sus siglas en inglés) y los efectos debidos a una dosis completa de ionización (TID, por sus siglas en inglés). El primer caso se debe al impacto de una única partícula sobre el material, de forma que se deposita suficiente energía como para causar algún efecto en el dispositivo. El segundo caso es el resultado de la acumulación de partículas debido a una larga exposición a las radiaciones [26], conllevando a la producción de electrones libres y pares electrón hueco. La acumulación de este efecto se mide

en rad donde $1 \text{ rad} = 10^{-2} \text{ Gy} = 6.24 \cdot 10^7 \text{ MeV/g}$ [4]. Este fenómeno puede enfrentarse blindando nuestra electrónica, y el grosor del blindaje depende de la cantidad de TID que se estime estará sometido el sistema.

A su vez, los efectos SEE pueden ser destructivos, conocidos como Single Event Latchup (SEL) o no destructivos. En el caso de los SEL, es un tipo de SEE que puede activar estructuras tiristor PNP parásitas en el dispositivo, provocando picos de corriente en el dispositivo que podrían llevar a su destrucción. Para combatir este fenómeno se usan detectores de picos de corriente que apagan el dispositivo en caso de detectar un sobreconsumo de corriente, evitando así que ocurran daños al mismo.

Los efectos SEE no destructivos se dividen en tres grandes grupos:

- Interrupción Funcional por Evento Simple (SEFI, por sus siglas en inglés): Consiste en un tipo de fallas provocadas por el efecto de partículas de alta energía que afecta registros de configuración de dispositivos lógicos, lo cual afecta el funcionamiento del dispositivo. A medida que el dispositivo se hace más complejo, la probabilidad de ocurrencia de este efecto aumenta [27].
- Transitorio por Evento Simple (SET, por sus siglas en inglés): Este consiste en una perturbación de corriente o voltaje momentáneos la cual se puede propagar a través de circuitos subsiguientes.
- Alteración por Efecto Simple (SEU, por sus siglas en inglés): Es una alteración en el estado lógico de una celda de memoria, en el caso de este tipo de efecto profundizaremos en la caracterización del mismo ya que es el más afecta a la tecnología FPGA SRAM sometida a radiación.

4.3.1 Efecto SEU

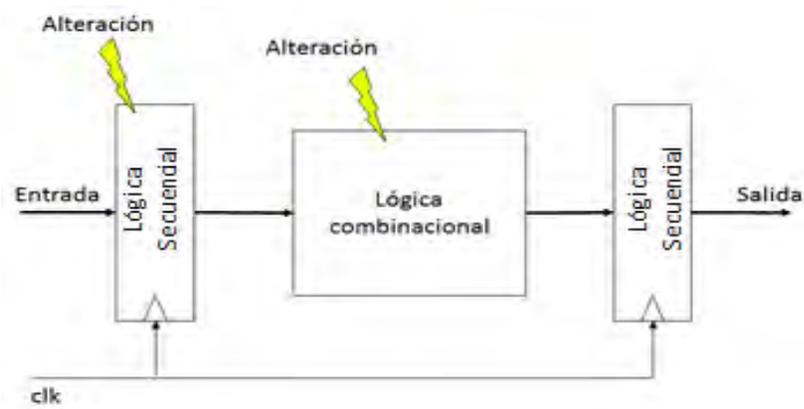


Figura 34 Alteración aplicada a la lógica combinacional y secuencial *Adaptado de [3]*

Una partícula de alta energía puede golpear tanto la lógica combinacional como la lógica secuencial del sistema, como se muestra en la Figura 55.

Cuando una partícula cargada golpea uno de los nodos sensibles de una unidad de memoria, como por ejemplo el drenador de un transistor apagado, se genera un pulso de corriente que puede activar la puerta del transistor opuesto, como se muestra en la Figura 56.

El efecto puede producir la inversión de un valor almacenado, en otras palabras, un cambio de bit en la unidad de memoria. Este fenómeno es conocido como Alteración de Evento Simple o SEU.

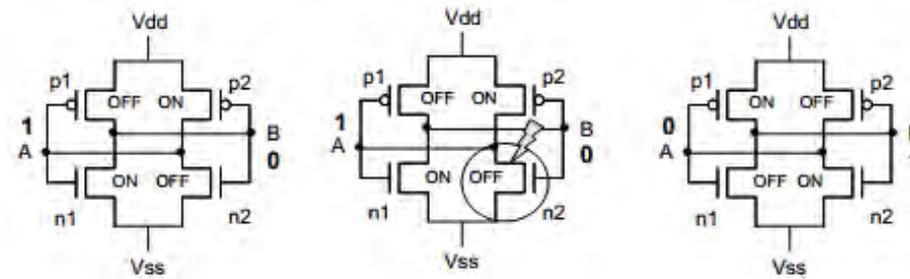


Figura 35 Efecto SEU en una unidad de memoria Adaptado de [3]

4.4 Métodos de mitigación de los efectos de la radiación en FPGAs

Las fallas pueden ser prevenidas o toleradas. En el primer caso, se realizan acciones para evitar la ocurrencia de fallas en la etapa de diseño, incluyéndose capacidades para minimizar la ocurrencia de estas. Las técnicas de enmascaramiento de fallas permiten la ocurrencia de un pequeño nivel de fallas, pero enmascara sus efectos de manera que estos no afecten el funcionamiento general del sistema. Las técnicas de recuperación ante fallas permiten que se comprometa temporalmente el funcionamiento del sistema, pero responde y se recupera luego de que la falla compromete una misión global. Las estrategias de cambio de tarea, permite que una falla comprometa la misión global del sistema, cambiando dicha misión a otra, usualmente degradada. [28]

A continuación, se presenta un resumen de las diferentes técnicas utilizadas para minimizar el efecto SEU

4.4.1 Técnicas orientadas al proceso de fabricación

Las técnicas basadas en el proceso de fabricación, también conocidas como técnicas tecnológicas, pueden reducir hasta niveles aceptables algunos de los efectos de la radiación, como TID y SEL, pero no eliminan completamente efectos SEU y SET. Entre las formas de implementarla podemos encontrar el procesamiento CMOS epitaxial y el aislamiento del sustrato de silicón.

4.4.2 Técnicas orientadas al diseño

Las técnicas basadas en el diseño, también llamadas técnicas arquitectónicas, son muy aceptadas ya que pueden ser aplicadas a varios niveles diferentes del diseño sin ningún cambio en la tecnología del proceso de fabricación. Pueden ser planeadas para solo detectar la presencia de una falla o pueden ser más complejas en aras de detectar y corregir errores del sistema ante la presencia de una falla. Todas estas técnicas están compuestas por algún tipo de redundancia, que se puede obtener a través de componentes extra (redundancia de hardware) o tiempo de ejecución extra (redundancia de tiempo). Muy frecuentemente, las técnicas usadas constan de una combinación de ambas. En la Figura 4.4 se ilustran las bases de estas técnicas.

En el caso de la redundancia por tiempo, se trata de aprovechar las características transitorias del pulso de corriente generado por las partículas, comparando las señales de salida en dos momentos diferentes. La salida de la lógica combinatorial es almacenada en dos

momentos diferentes, donde el pulso de reloj del segundo latch está demorado un tiempo d . Un comparador indica la ocurrencia del pulso detectando el error.

En el caso de la redundancia por hardware, puede ser usada para ambas lógicas, y se compara la salida de dos elementos que realizan la misma función.

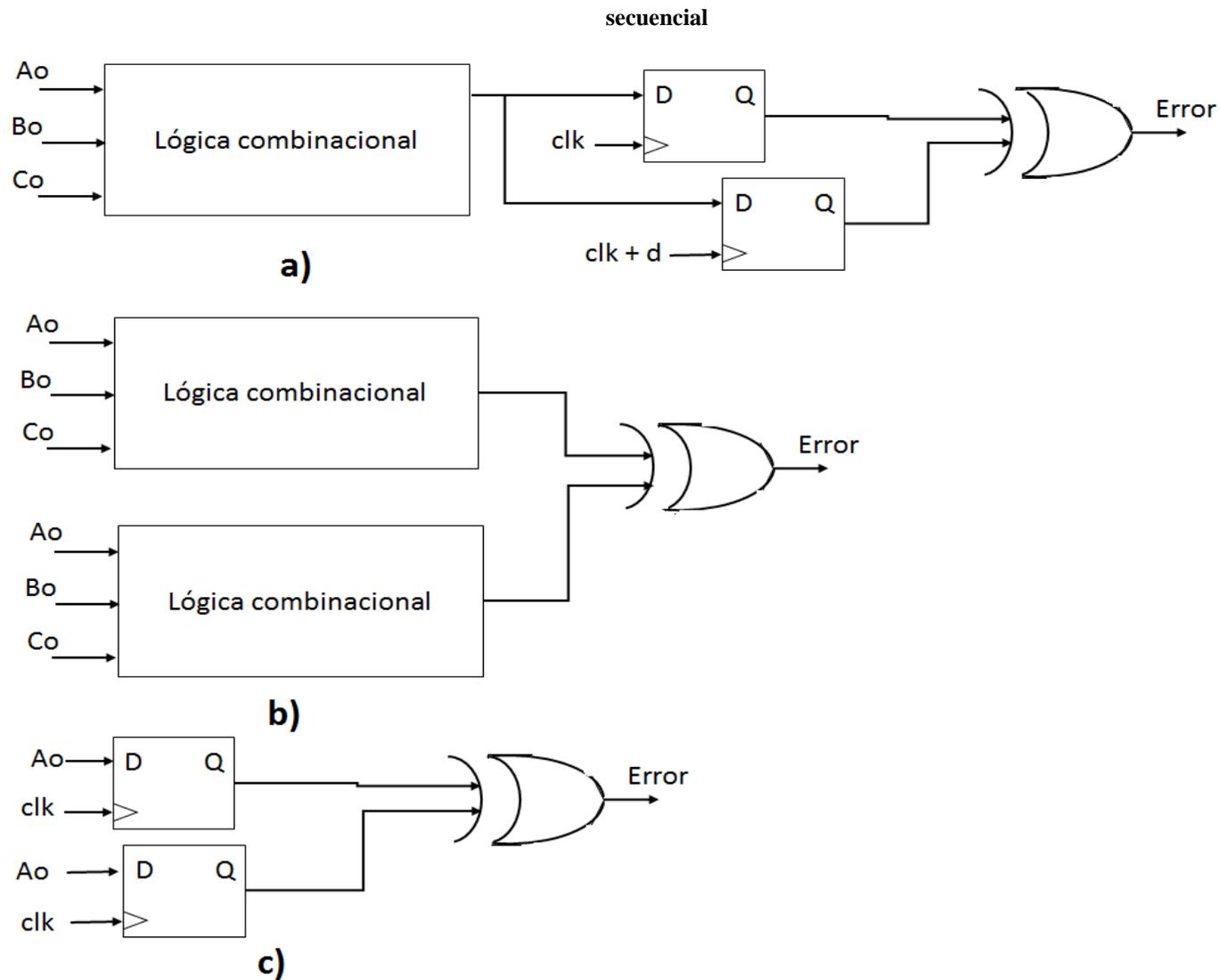


Figura 36 a) Redundancia por tiempo b) Redundancia por hardware para lógica combinacional c) Redundancia por hardware para lógica

Adaptado de [4]

La triple redundancia modular (TMR por sus siglas en inglés), consiste en usar 3 módulos de redundancia y un elemento de decisión que toma la salida de cada módulo y decide cual es la respuesta correcta. En la Figura 58 se muestra cómo implementar TMR en tiempo. La salida de la lógica combinacional es demorada en intervalos de tiempo d , y comparada en el votador para decidir la respuesta correcta.

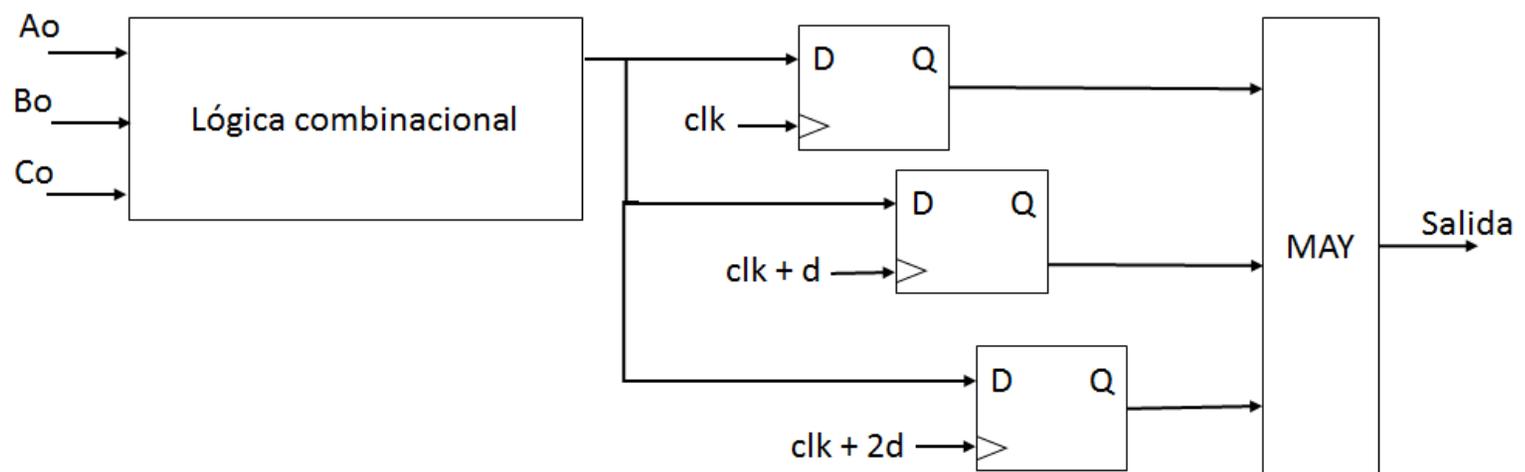


Figura 37 TMR en tiempo *Adaptado de [3]*

En el caso de la redundancia TMR por hardware, se usan bloques con la misma funcionalidad, y se compara su salida para obtener la respuesta correcta, como se muestra en la Figura 59.

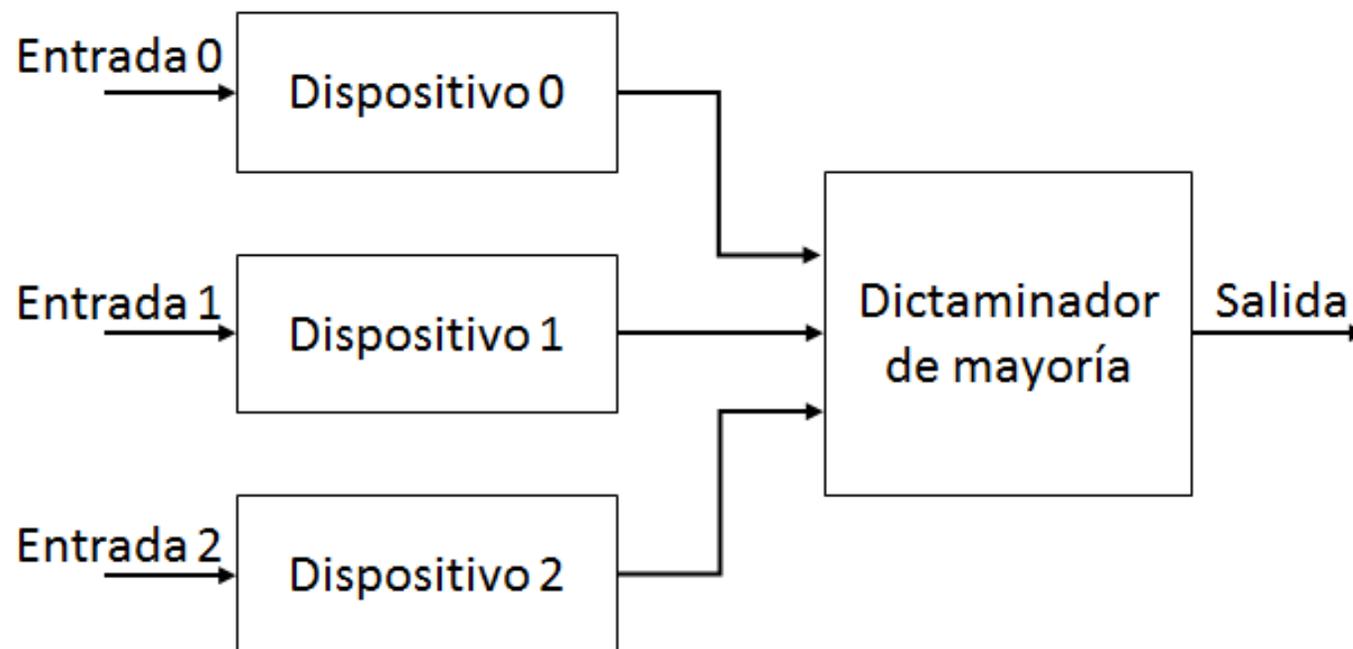


Figura 38 TMR en hardware *Adaptado de [4]*

Ya que el dictaminador de mayoría es en sí un elemento que puede fallar, este también es sometido a redundancia y retroalimentación para lograr la resincronización en caso de ocurrencia de una falla, como se muestra en la Figura 60.

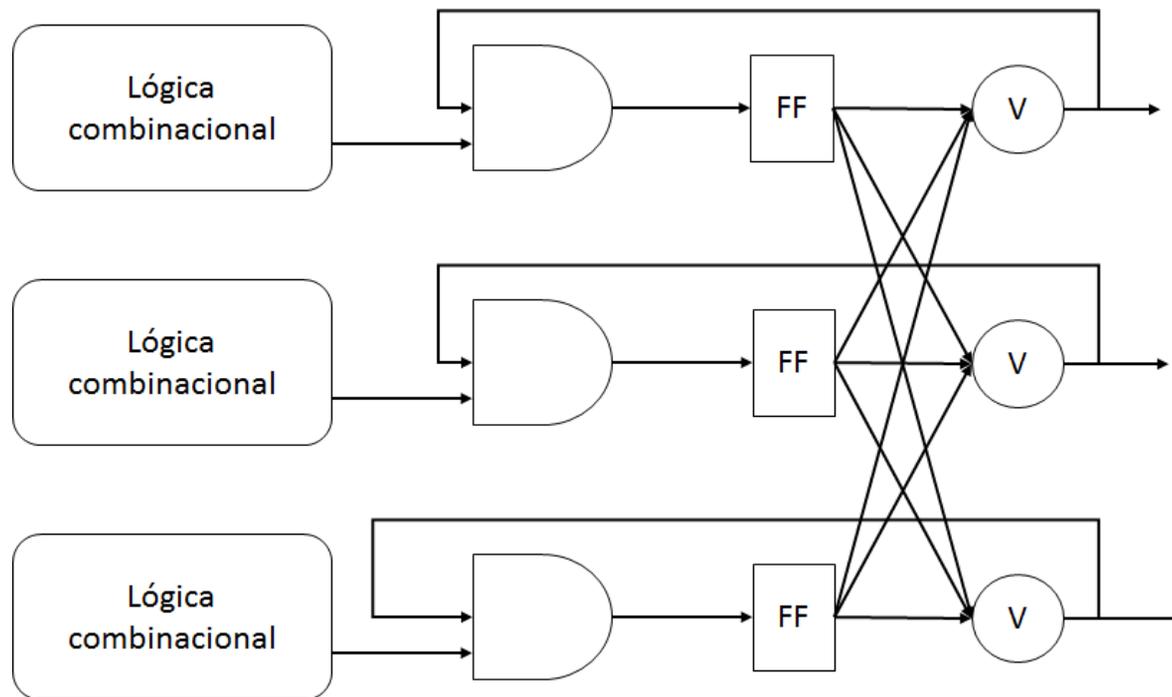


Figura 39 Votador para sincronización Adaptado de [20]

Los códigos de detección y corrección de errores o EDAC también son usados para mitigar los SEU en circuitos integrados. Hay muchos códigos usados con este fin, un ejemplo muy popular es el código Hamming, el cual es un código binario de detección y corrección de errores que puede detectar cualquier error de uno o dos bits y corregir todos los errores de un bit.

4.4.3 Técnicas de recuperación

Scrubbing ciego vs Scrubbing con retroalimentación

El refrescamiento o scrubbing consiste en la reconfiguración de un dispositivo usando parámetros de los cuales estamos seguros de su validez. Esto se realiza escribiendo en la memoria de configuración del dispositivo una secuencia de bits almacenada en una memoria externa protegida contra los efectos de la radiación, y puede realizarse de varias formas.

Entre los elementos a considerar está el método de desencadenamiento de la reconfiguración, esta puede realizarse de forma ciega, lo cual se considera un método de prevención de fallas, ya que se sobre escribe la memoria, aunque no haya ocurrido una falla. Por otra parte, se puede leer la secuencia almacenada en la memoria de configuración, y compararla con la copia almacenada como correcta o goldstream. Solo si se detectan errores, la memoria de configuración es sobre escrita.

Comparando ambos métodos, la reconfiguración a ciegas es la más simple de implementar, pero realiza una cantidad innecesaria de accesos a la memoria de configuración, mientras que la reconfiguración con retroalimentación es más compleja de implementar ya que se necesitan técnicas de comparación y detección de errores, pero es más robusta y usa menos accesos a la memoria.

Scrubbing Total vs Scrubbing por tramas

Ambas técnicas mencionadas anteriormente, se pueden llevar a cabo usando el goldstream total, o separando este por tramas. El primer método, requiere una implementación más simple, ya que la secuencia de configuración puede ser descargada directamente desde la memoria hacia la interfaz de configuración. Peor en el caso de que suceda una falla durante el proceso de descarga del goldstream, el funcionamiento global del dispositivo quedaría comprometido. Para evitar esto, se pueden separar el goldstream en varias tramas, lo cual es más difícil de implementar ya que cada trama debe ser preparada antes de la descarga, pero tiene la ventaja de aislar el efecto de una falla a una sola trama. También trae consigo un aumento en el tiempo de reconfiguración, debido a la información adicional que se agrega a cada trama para su transmisión.

Scrubbing Periódica vs Scrubbing por Demanda

El proceso de reconfiguración, puede desencadenarse de forma periódica, o ser activado por algún método de detección de fallas. La disponibilidad de un sistema depende del tiempo en que una falla permanece sin reparar. Por tanto, mientras mayor sea la frecuencia de reconfiguración, menor será el tiempo de indisponibilidad. No obstante, reconfiguraciones excesivas pueden ser no deseadas, por tanto, es más viable la implementación de mecanismos de detección de fallas que desencadenen el proceso de reconfiguración inmediatamente después de detectada la falla.

Scrubbing Externa vs Scrubbing Interna

El almacenamiento del goldstream se puede realizar tanto en una memoria externa, usualmente fortalecida contra efectos de radiación, o almacenarse internamente en el dispositivo. El primer caso, considerado el más robusto, tiene el inconveniente de necesitar un controlador para descargar el goldstream, y el uso de una memoria externa. Sin embargo, la reconfiguración interna, aunque prescinde de estos elementos, no puede recuperarse ante eventos SEU, si estos corrompen la secuencia de configuración.

4.5 Herramientas que permiten la implementación de tolerancia a fallas de forma automática

Aunque la aplicación de TMR es un proceso que muchas veces se aplica de forma manual al diseño, se pueden usar herramientas que realicen la inserción de la redundancia de forma automática. La inserción de votadores de forma manual es un proceso tedioso y propenso a errores, sobre todo en diseños extensos y complejos. El uso de una herramienta automática hace el proceso mucho más sencillo y rápido. [29]

Entre las herramientas que realizan esta tarea podemos citar Xilinx XTMR Tool, Precision Hi-Rel y Synplify, todas propietarias. En el ámbito de software libre encontramos ByuEdifTool, desarrollada por el NSF Center for High-Performance Reconfigurable Computing de la Universidad Brigham Young.

4.5.1 Xilinx TMR Tool

Esta herramienta permite la implementación en FPGAs de Xilinx de redundancia modular triple partiendo de un diseño sintetizado. Soporta los sistemas operativos Windows 2000/XP/7, y acepta como archivo de entrada un diseño digital sintetizado en cualquier herramienta.

Entre las características de la TMR aplicada por la herramienta podemos destacar las siguientes [29]:

- Permite triplicar todas las entradas incluyendo señales de reloj
- Permite la triplicación de las lógicas de retroalimentación, insertando votadores de mayoría en las rutas de retroalimentación, evitando así la necesidad de resetear los componentes de una red con redundancia en aras de mantener la sincronía.
- Protección de los votadores de mayoría mediante triplicación

4.5.2 Synopsis Synplify Premier

El uso de esta herramienta permite la inserción de forma automática en nuestro diseño de medios para hacerlo a fallas inducidas por la radiación u otras causas que puedan afectar el funcionamiento del sistema.

Entre las funcionalidades a destacar de la misma podemos encontrar [30]:

- Soporte para la detección de SEUs y esquemas de recuperación para una gran variedad de familias como Altera, Lattice, Microsemi y Xilinx.
- Soporte automático para la creación de monitores de errores SEU, permitiendo la implementación de sistemas de mitigación de errores basados en software para controlar, monitorear, diagnosticar y mitigar la ocurrencia de dichos errores.
- Protección de elementos de memoria mediante códigos correctores de errores (ECC, por sus siglas en inglés).
- Implementación de TMR local para la protección de registros, TMR en bloques para la protección de bloques IP y rutas específicas, TMR para lógica de retroalimentación y separación física de las partes triplicadas para obtener una protección adicional.

4.5.2 ByuEdifTool

La herramienta ByuEdifTool es capaz de insertar redundancia en un diseño digital para FPGA con el objetivo de incrementar la confiabilidad. La herramienta es capaz de insertar redundancia modular triple y duplicación con comparación, y cuenta con varias opciones configurables que determinan el método de inserción de los votadores. Entre las características a destacar de la herramienta podemos encontrar [31]:

- Soporta una gran variedad de familias de FPGAs de Xilinx
- Permite la inserción de diferentes técnicas de redundancia como TMR y DWC en nuestro diseño digital
- La implementación de la tolerancia puede ser sobre el sistema total o una parte del mismo
- Gran variedad de opciones en cuanto al método de aplicación de la redundancia como son ignorar cajas negras, la ubicación de los votadores y el método de inserción de los mismos.
- Permite la inserción de votadores en la lógica de retroalimentación

Por los elementos antes expuestos podemos llegar a la conclusión de que existe una gran similitud entre las funcionalidades de las herramientas seleccionadas, y para el caso específico de nuestro sistema todas cumplen con los requerimientos básicos los cuales son que debe ser capaz de generar un diseño digital aplicable a FPGAs de Xilinx y contener varias opciones de técnicas de tolerancia a fallas a aplicar, de manera que se pueda realizar un estudio comparativo entre las mismas. En nuestro caso seleccionaremos la herramienta ByuEdifTool, la cual cumple con nuestros requerimientos, es de código abierto y adicionalmente contamos con la colaboración por parte del equipo desarrollador de la herramienta.

4.6 Ejemplo de utilización de la herramienta ByuEdifTool

En el siguiente apartado se desarrolla un ejemplo donde se muestre el uso de la herramienta seleccionada en el punto anterior. Partimos de un sistema sin redundancia, y mostraremos como implementar dicho sistema ya con redundancia mediante el uso de la herramienta. Para este ejemplo seleccionamos realizar la comparación entre un sistema simple y dicho sistema luego de que se le aplique TMR. La disponibilidad de un sistema es la probabilidad de que el sistema sea operacional en un tiempo t dado una tasa de fallo λ , y se calcula como se muestra en las Ecuaciones 4.1 (sistema simple sin redundancia) y 4.2 (sistema con redundancia modular triple). [31]

$$R_1(t) = e^{-\lambda t} \quad (4.1)$$

En el caso de los sistemas con TMR sin scrubbing, se modelan con la Ecuación 4.2

$$R_2(t) = 3e^{-2\lambda t} + 2e^{-\lambda t} \quad (4.2)$$

Para poder apreciar las ventajas del uso de la redundancia modular triple, se realizó un análisis donde se graficaron las ecuaciones 4.1 y 4.2. Se tomaron como parámetros una tasa de fallos $\lambda = 0.001$ y se realizó la gráfica para 1000 unidades de tiempo. En la Figura 4.8 podemos apreciar el resultado de la comparación. Podemos ver que inicialmente la redundancia modular triple presenta un valor de confiabilidad muy superior al de un sistema simple. A medida que pasa el tiempo, la probabilidad de falla en ambos sistemas comienza a aumentar, pero para la cantidad de tiempo analizada siempre es superior la redundancia modular triple.

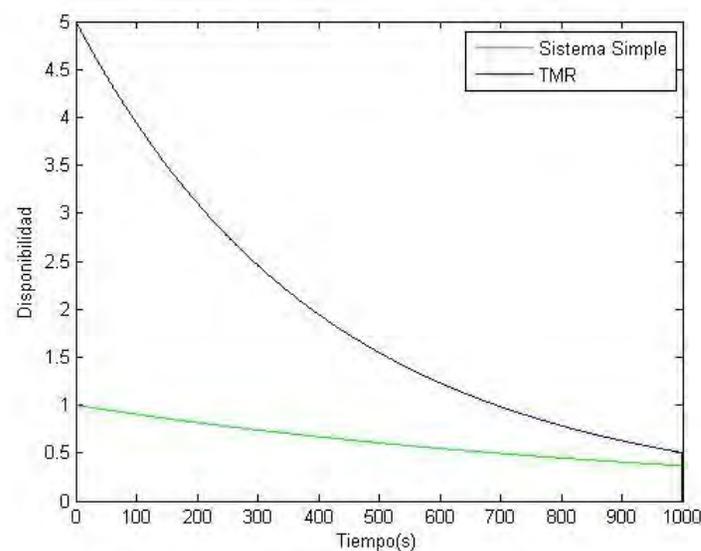


Figura 40 Gráfica de disponibilidad para un sistema simple y uno redundante

4.6.1 Descripción de la herramienta

Para obtener un diseño con redundancia, se parte del diseño simple el cual se entrega en un archivo .edif. La herramienta está dividida en varias sub-herramientas, algunas de las cuales son opcionales, dependiendo su aplicación de las necesidades del usuario. El primer paso es descargar la herramienta, la cual se encuentra en <https://sourceforge.net/projects/byuediftools/>. En este URL podemos descargar el archivo byuediftool.jar, el cual debemos guardar en nuestro sistema de archivos.

Para poder utilizar esta herramienta, se realizó una colaboración con un colega del NSF Center for High-Performance Reconfigurable Computing de la Universidad Brigham Young, el cual brindó asesoría en varios aspectos como fueron la instalación de la herramienta, la configuración de la computadora para que pudiera ejecutarla y la configuración de la herramienta para que pudiera ejecutar el tipo de análisis deseado.

En la Figura 4.9 se puede apreciar las diferentes opciones a tener en cuenta a la hora de usar la herramienta. Los recuadros en línea continua representan pasos o sub-herramientas que son de uso obligatorio, mientras que los que usan líneas discontinuas son opcionales. La herramienta está desarrollada en Java, por lo que es necesario contar con un compilador de java instalado en nuestro computador para correr los comandos. Una vez instalado dicho compilador, podemos correr desde la consola de comandos de Windows cada uno de los métodos por separado, o usar un archivo de extensión .MakeFile que englobe todos los comandos a utilizar. Es necesario desplazarse en el sistema de archivos hasta la ubicación donde guardamos la herramienta, y a partir de ahí ejecutar los comandos de java.

A continuación, describiremos el proceso para poder ejecutar las diferentes tareas mostradas en la Figura 4.9, así como una descripción de las herramientas de uso obligatorio.

JEdifBuild: Este es el primer comando a ejecutar, el cual toma nuestro diseño sin redundancia el cual se encuentra en un archivo .edf generado el software usado para la síntesis del código VHDL. En nuestro caso, usamos el software Vivado para la síntesis del código. Una vez que tenemos abierto el diseño sintetizado corremos el comando `write_edif nombre_archivo.edf`, como se muestra en la Figura 4.10, el cual exporta el netlist sintetizado. Este archivo constituye la entrada al comando JEdifBuild, el cual acondiciona nuestro diseño para su posterior procesamiento por la herramienta.

JEdif Replication Toolflow

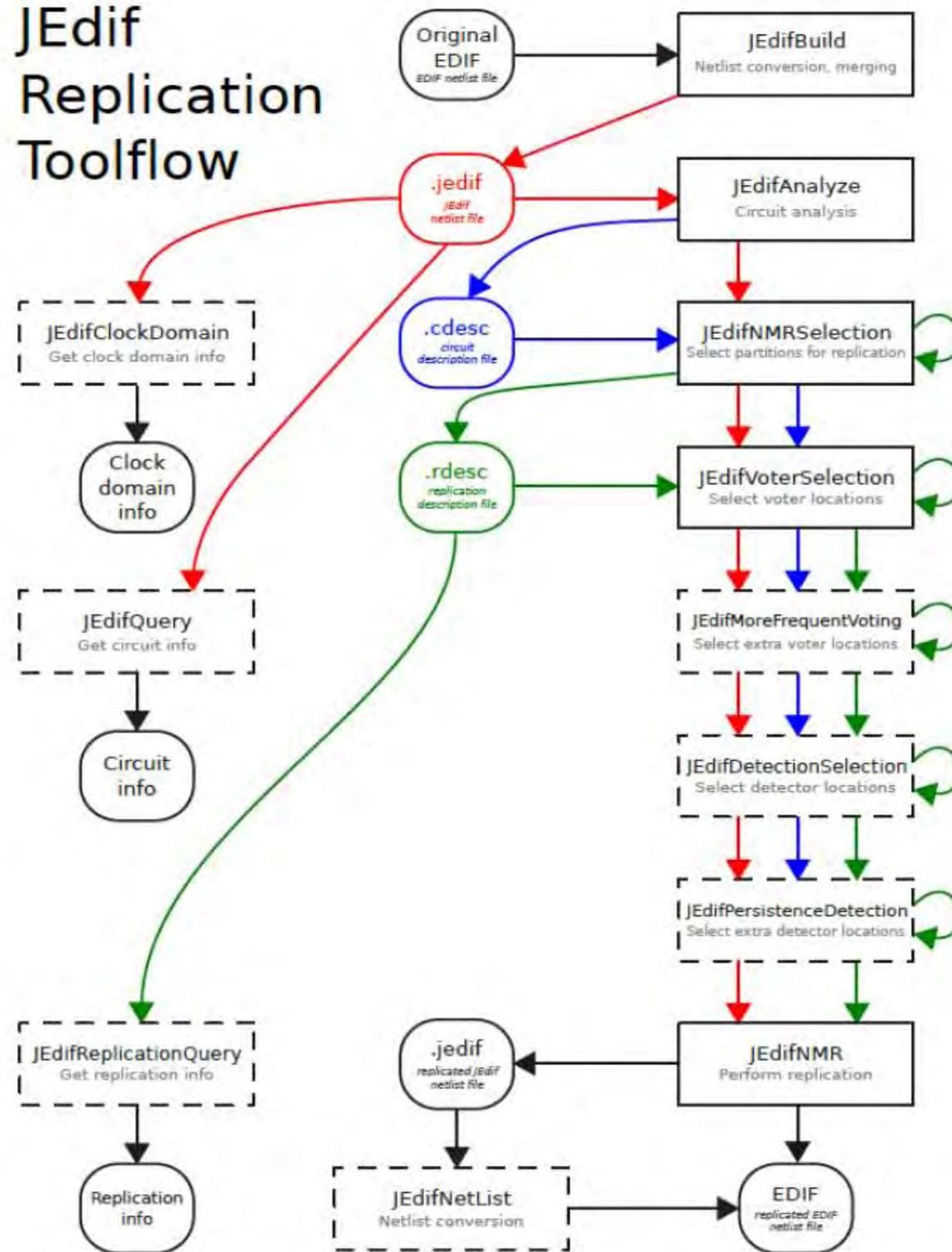


Figura 41 Flujograma de la herramienta JEdifTool Adaptado de [45]

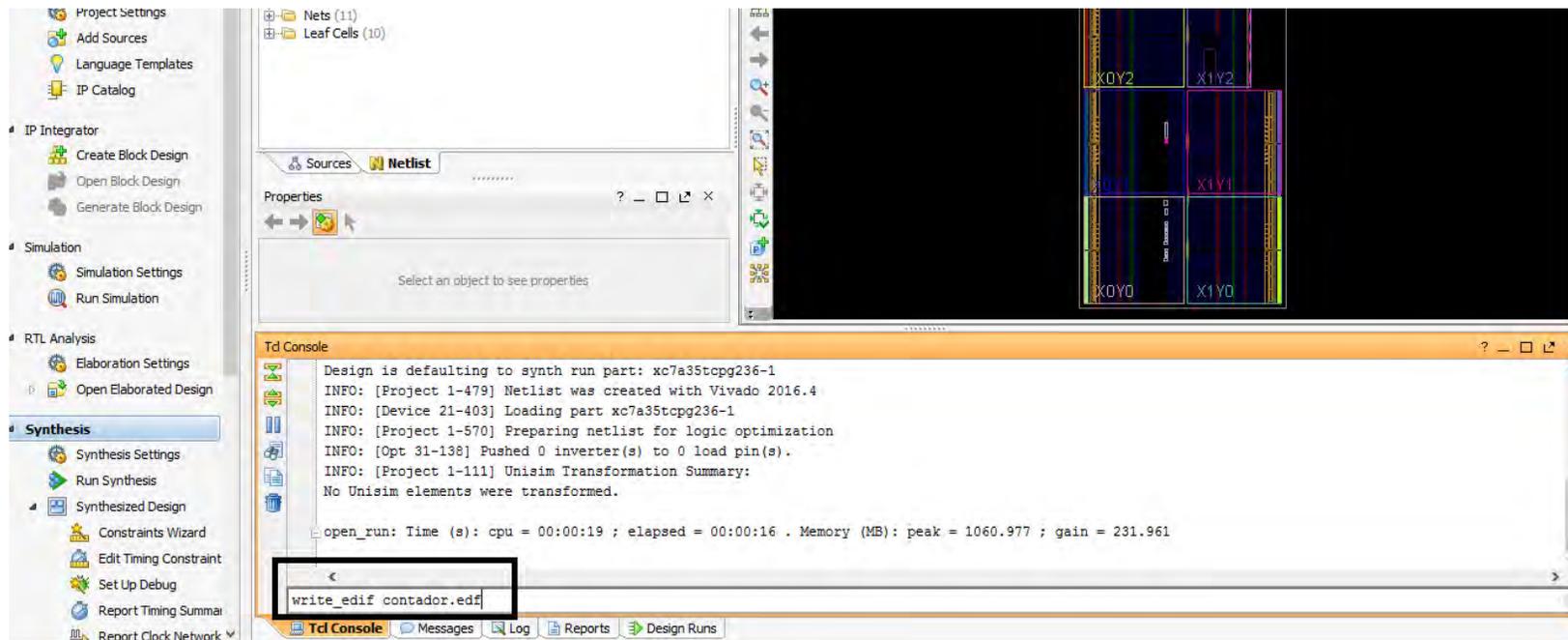


Figura 42 Exportar archivo .edf desde Vivado

Una vez que tenemos el archivo .edf, corremos el comando JEdifBuild desde la consola de comando de Windows. Para esto usamos el comando:

```
java -Xmx1G -cp byuEdiftools.jar;JSAP-2.1.jar edu.byu.ece.edif.jedif.JEdifBuild contador.edf --output contador.jedif
```

Este comando de java hace que la herramienta tome el archivo *contador.edf* el cual generamos en Vivado, lo procese y acondicione, y nos devuelve el archivo *contador.jedif*, para ser usado en etapas posteriores.

JEdifAnalyze: Esta herramienta realiza un análisis de los caminos de retroalimentación en nuestro diseño, generando un archivo de extensión .cdesc requerido por etapas posteriores. Para ejecutar la herramienta usamos el comando de java:

```
java -Xmx1G -cp byuEdiftools.jar;JSAP-2.1.jar edu.byu.ece.edif.jedif.JEdifAnalyze contador.jedif --c_desc contador.cdesc
```

Con este comando la herramienta toma el archivo *contador.jedif* generado por *JEdifBuild* y genera el archivo de descripción *contador.cdesc* para ser usado por etapas posteriores.

JEdifNMRSelection: Esta herramienta determina que partes del diseño serán replicadas, y determina el tipo de replicación. El comando en java para ejecutar la misma es:

```
java -Xmx1G -cp byuEdiftools.jar;JSAP-2.1.jar edu.byu.ece.edif.jedif.JEdifNMRSelection contador.jedif --c_desc contador.cdesc --r_desc contador.rdesc --replication_type triplication --full_nmr
```

En este caso, partimos del archivo *contador.jedif* y la información presente en el archivo *contador.cdesc* para generar el archivo *contador.rdesc* el cual contiene una descripción de la redundancia a aplicar. La opción *replication_type* se configura como *triplication*, lo cual significa que nuestro diseño será triplicado, mientras que *full_nmr* implica que se aplicará la redundancia a todo el diseño.

JEdifVoterSelection: Determina en que parte de nuestro diseño serán ubicados los votadores. El resultado se agrega al archivo *contador.rdesc*. Entre las opciones para la inserción de los votadores encontramos *before_ff_cutset* y *after_ff_cutset*, y como sus nombres

lo indican ahcen que lso votadores se inserten antes o después de cada flip flop. En nuestro ejercicio aplicaremos la opción *after_ff_cutset*. El comando en java quedaría

```
java -Xmx1G -cp byuEdiftools.jar;JSAP-2.1.jar edu.byu.ece.edif.jedif.JEdifVoterSelection contador.jedif --c_desc contador.cdsc --r_desc contador.rdesc --after_ff_cutset
```

JEdifNMR: El último comando a ejecutar, toma la información almacenada en el archivo *contador.rdesc* acerca de que parte replicar y donde insertar los votadores e inserta la redundancia en el diseño.

```
java -Xmx1G -cp byuEdiftools.jar;JSAP-2.1.jar edu.byu.ece.edif.jedif.JEdifNMR contador.jedif --r_desc contador.rdesc contador_nmr.edif
```

Podemos ver que la herramienta toma los archivos *contador.jedif* y *contador.rdesc* y genera la salida *contador_nmr.edif*. Este archivo ya contiene nuestro diseño redundante, y puede ser usado para crear un proyecto post-síntesis en Vivado, para continuar con la implementación.

En la Figura 64 se muestra el resultado de la síntesis de un circuito sencillo, en este caso un contador de 0 a 9, mientras que en la Figura 65 se muestra el resultado de abrir el archivo *contador_nmr.edif* en Vivado.

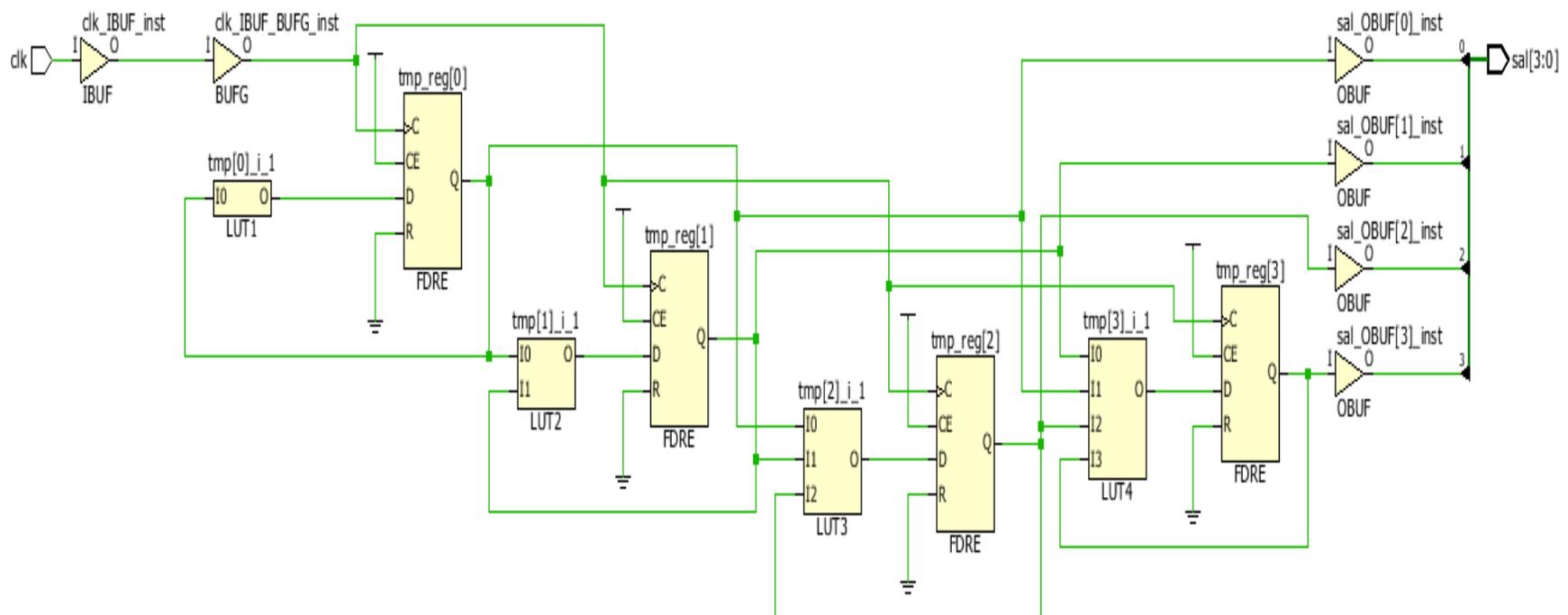


Figura 43 Resultado de la síntesis del diseño simple

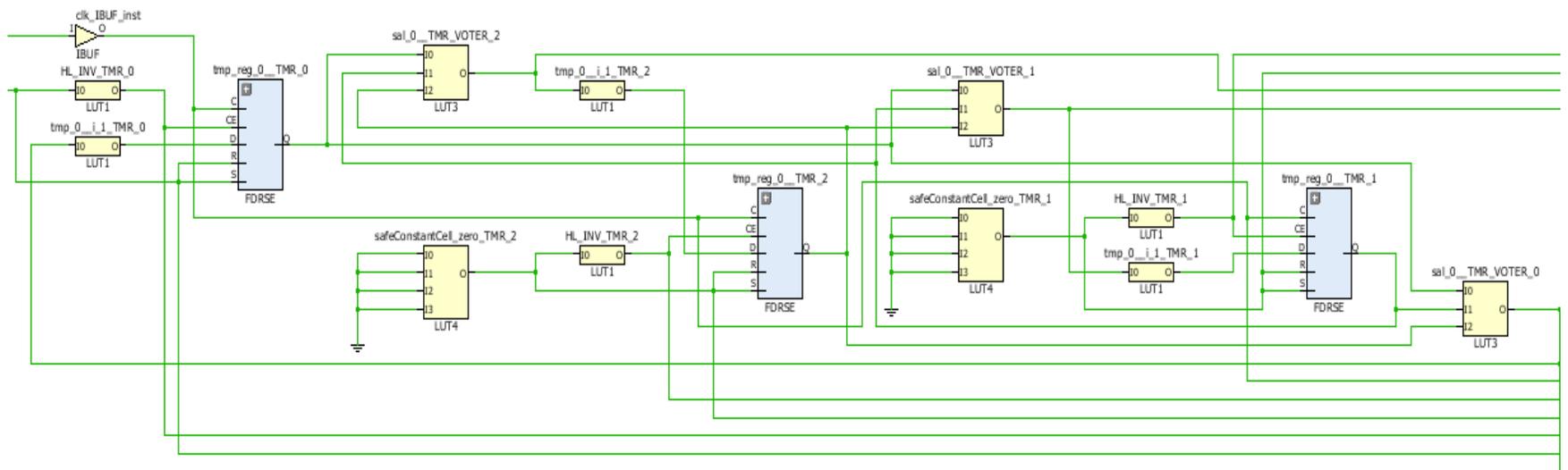


Figura 44 Resultado de la síntesis del diseño replicado

En la Figura 65 se muestra una sección del diseño triplicado por la herramienta. Si comparamos esta imagen con la figura 64, podemos apreciar que el bloque tmp_reg_0 fue triplicado, obteniéndose los bloques tmp_reg_0_TMR_0, tmp_reg_0_TMR_1 y tmp_reg_0_TMR_2. También podemos apreciar la inserción de votadores a las salidas de los flip-flops, como es el caso de los bloques sal_0_TMR_VOTER_0, sal_0_TMR_VOTER_1 y sal_0_TMR_VOTER_2. El mismo resultado se obtiene para todos los contadores.

4.7 Aplicación de TMR al modulador adaptativo usando la herramienta ByuEdifTool

Un elemento importante a tener en cuenta es que la aplicación de redundancia a un sistema incrementa su confiabilidad a costa de una cantidad superior de recursos comparado con un sistema sin redundancia. Este factor puede implicar la necesidad de cambiar de tecnología o dispositivo, por lo que es muy importante dimensionar la cantidad de recursos extras que se requerirán luego de aplicada la redundancia.

Hierarchy					
Name	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	PLLE2_ADV (5)	
Test_bench_GMSKmodulator	200	30	22	1	
UA (Generador_frecuencias)	1	0	0	1	
UC (QPSKmodulator)	31	5	0	0	
UD (PSK_8_Modulator)	35	9	0	0	
UE (NCO1)	81	5	0	0	
UF (QAM16modulator)	21	11	0	0	
UI (multiplier)	29	0	0	0	
UJ (multiplier_0)	14	0	0	0	
UK (adder)	0	0	0	0	

Figura 45 Reporte de utilización de Vivado por cantidad de elementos ocupados. Diseño sin redundancia

Hierarchy				
Name	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	PLLE2_ADV (5)
Test_bench_GMSKmodulator	0.96%	0.07%	20.75%	20.00%
UA (Generador_frecuencias)	<0.01%	0.00%	0.00%	20.00%
UC (QPSKmodulator)	0.15%	0.01%	0.00%	0.00%
UD (PSK_8_Modulator)	0.17%	0.02%	0.00%	0.00%
UE (NCO1)	0.39%	0.01%	0.00%	0.00%
UF (QAM16modulator)	0.10%	0.03%	0.00%	0.00%
UI (multiplier)	0.14%	0.00%	0.00%	0.00%
UJ (multiplier_0)	0.07%	0.00%	0.00%	0.00%
UK (adder)	0.00%	0.00%	0.00%	0.00%

Figura 46 Reporte de utilización de Vivado dado en por ciento. Diseño sin redundancia

Hierarchy			
Name	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)
Test_bench_GMSKmodulator_flat	676	90	22

Figura 47 Reporte de utilización de Vivado por cantidad de elementos ocupados. Diseño con TMR

Hierarchy			
Name	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)
Test_bench_GMSKmodulator_flat	3.25%	0.22%	20.75%

Figura 48 Reporte de utilización de Vivado dado en por ciento. Diseño con TMR

Luego del análisis mostrado en las Figuras 66 – 69, podemos concluir que, a pesar de que se incrementa la ocupación de elementos dentro del FPGA, principalmente en las LUTs (donde va de 0,96% a 3.5%) este incremento no impacta significativamente en el uso total de los recursos del FPGA, por lo que la adición de tolerancia a fallas usando TMR no constituye un riesgo en cuanto a la ocupación de recursos del FPGA.

4.8 Conclusiones del capítulo

En este capítulo se realizó un análisis del efecto que tienen las partículas de alta energía presentes en el ambiente espacial sobre los bloques sensibles del FPGA SRAM, complementado con un estudio documental acerca de las técnicas de tolerancia a fallas más utilizadas en sistemas espaciales, con el objetivo de brindar protección contra efectos indeseables como los SEU. Se realizó un análisis comparativo entre varias herramientas que permiten la implementación de sistemas tolerantes a fallas de forma automática. Se seleccionó ByuEdifTool para la realización de un ejemplo demostrativo del funcionamiento de la misma, el cual se llevó a cabo al implementar TMR sobre un contador de 0 a 9. Posteriormente se dimensionó la cantidad de recursos ocupados por el diseño correspondiente al modulador adaptativo sin redundancia, para compararla con los recursos ocupados por el mismo modulador una vez aplicada la TMR usando la herramienta ByuEdifTool, donde se mostró que en este caso no se compromete la ocupación de recursos totales del FPGA.

Capítulo 5. Pruebas de validación y resultados

En este capítulo se muestran las pruebas realizadas para validar el correcto funcionamiento del sistema.

Para realizar la validación de nuestro sistema se propuso la implementación en el mismo FPGA de un circuito digital que realizara la demodulación de la información modulada. Para comprobar el correcto funcionamiento del sistema se compara la entrada de datos al modulador con la salida respectiva en el demodulador. Para fines de validación, y por cuestiones de alcance del trabajo, no se realizó el proceso de recuperación de la portadora a partir de la señal modulada. Por cuestiones de simplicidad, se tomaron las señales portadoras y de reloj generadas en el modulador para realizar la demodulación.

5.1 Validación del modulador QPSK

El esquema del demodulador QPSK se muestra en la Figura 5.1

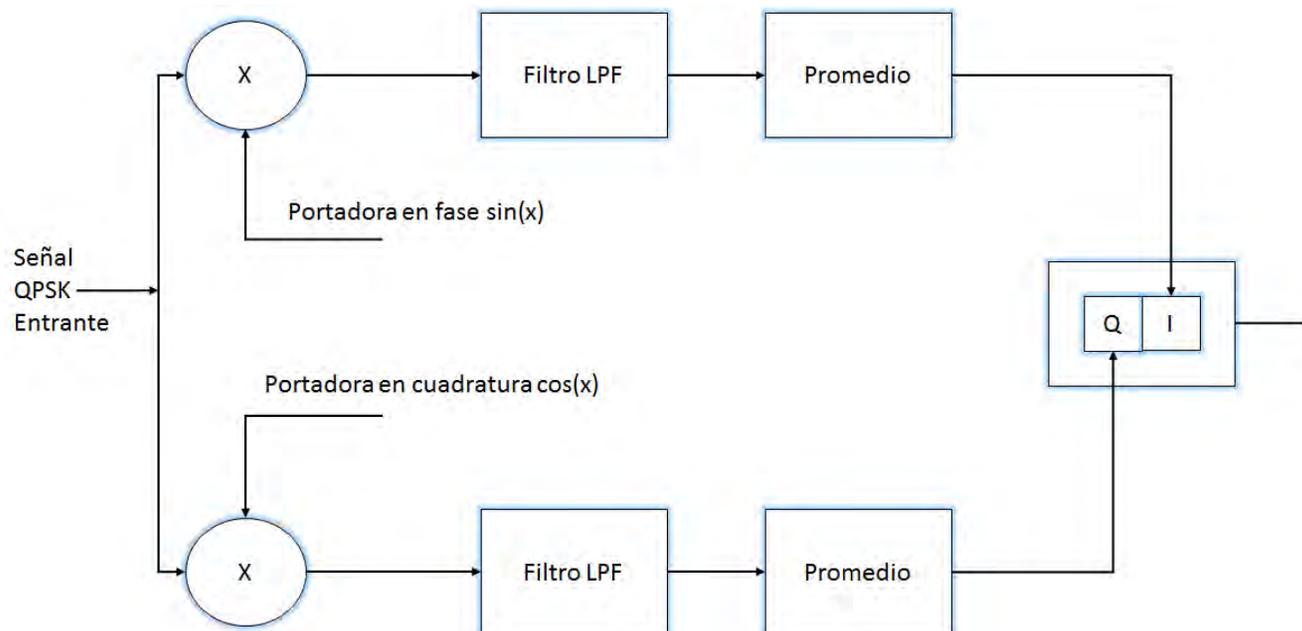


Figura 49 Esquema a bloques del demodulador QPSK

La señal QPSK entrante es multiplicada por la señal portadora, en fase para el canal I y en cuadratura para el canal Q. La señal obtenida a la salida del multiplicador es filtrada con el objetivo de eliminar las componentes de alta frecuencia. A la salida del filtro se obtiene el promedio de la señal durante un período de bit T_b , entregando a la salida del bloque promedio el valor transmitido, 0 ó 1. Los dos canales son combinados para generar el flujo de bits de salida.

Como ejemplo demostrativo del funcionamiento del demodulador, supongamos que se transmitió la combinación 01. Esto generaría la señal QPSK $-\sin w_c t + \cos w_c t$. En el caso del canal I se multiplica esta señal por la portadora $\sin w_c t$

$$\sin w_c t * (-\sin w_c t + \cos w_c t) = -\sin^2 w_c t + \sin w_c t * \cos w_c t$$

$$= -\frac{1}{2}[1 - \cos 2w_c t] + \frac{1}{2}\sin[(w_c + w_c)t] + \frac{1}{2}\sin[(w_c - w_c)t]$$

El filtro elimina todas las componentes de alta frecuencia, quedando la constante -0.5. Para el canal Q obtenemos 0.5.

$$\begin{aligned} \cos w_c t * (-\sin w_c t + \cos w_c t) &= \cos^2 w_c t - \sin w_c t * \cos w_c t \\ &= \frac{1}{2}[1 + \cos 2w_c t] - \frac{1}{2}\sin[(w_c + w_c)t] - \frac{1}{2}\sin[(w_c - w_c)t] \end{aligned}$$

De esta forma podemos decir que se recibió la combinación de bits 01, la misma que fue transmitida. Este circuito digital fue implementado en el FPGA, como se muestra en la Figura 71

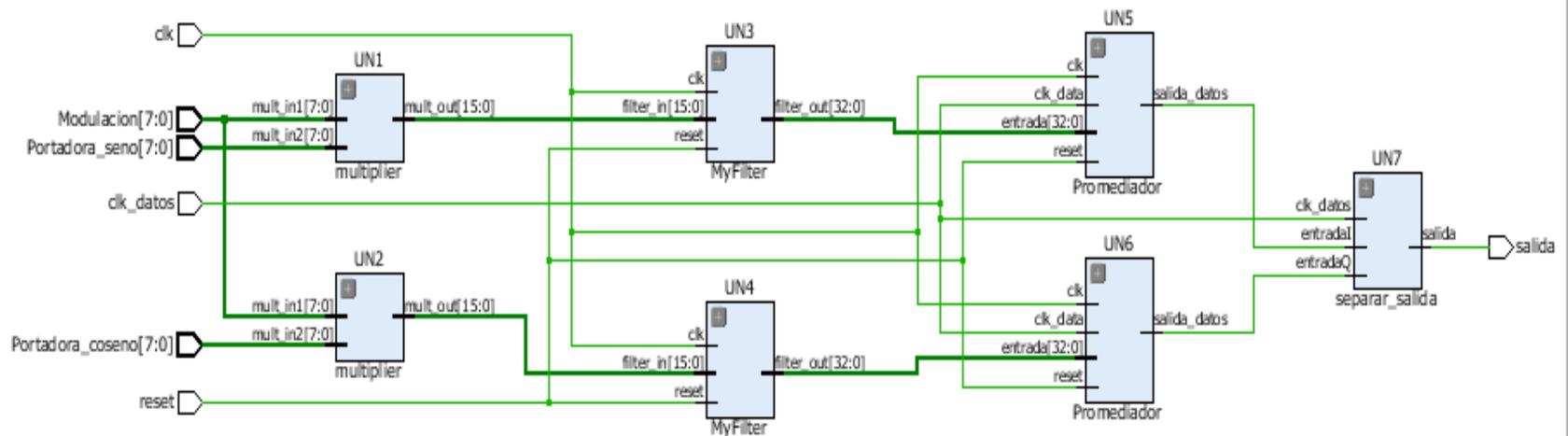


Figura 50 Circuito digital implementado en Vivado del demodulador QPSK

Para validar el funcionamiento del modulador QPSK se realiza la simulación usando el software ModelsimPE Student donde podemos visualizar las señales en diferentes puntos del diseño. El resultado de la simulación se muestra en la Figura 72



Figura 51 Validación del modulador QPSK

En la Figura 72 podemos ver 6 señales, por orden de aparición la señal de datos, el reloj de datos, los canales I y Q, la señal portadora original y la señal modulada QPSK. Para probar el modulador se usó la secuencia binaria 00011110. Podemos apreciar que la fase de la señal cambia cada 2 bits, y que las señales I y Q tiene cambios de fases de 180 grados, mientras que la señal combinada, al ser una combinación de ambas, solo tiene cambios de 90 grados en la fase. También podemos apreciar que los cambios de fase obtenidos para las diferentes combinaciones son: *para 11* $\varphi_1 = \frac{\pi}{4}$; *para 01* $\varphi_2 = 3\frac{\pi}{4}$; *para 00* $\varphi_3 = 5\frac{\pi}{4}$ y *para 10* $\varphi_4 = 7\frac{\pi}{4}$ como nos planteamos inicialmente, por lo que el modulador funciona correctamente. Posteriormente alimentamos al demodulador con nuestra señal QPSK modulada, y verificamos que a la salida se obtenga el mismo flujo de datos que a la entrada. Usaremos igualmente la secuencia 00011110. En la Figura 5.4 se muestran las siguientes señales:

- datos de entrada
- reloj de datos
- reset
- control: esta señal controla la modulación de salida con la siguiente lógica: 00 QPSK, 01 8PSK y 10 16QAM
- RDY: señal generada por el PLL indicando que la frecuencia se ha estabilizado. 0 estabilizado 1 no estabilizado
- Datos a la salida del demodulador QPSK

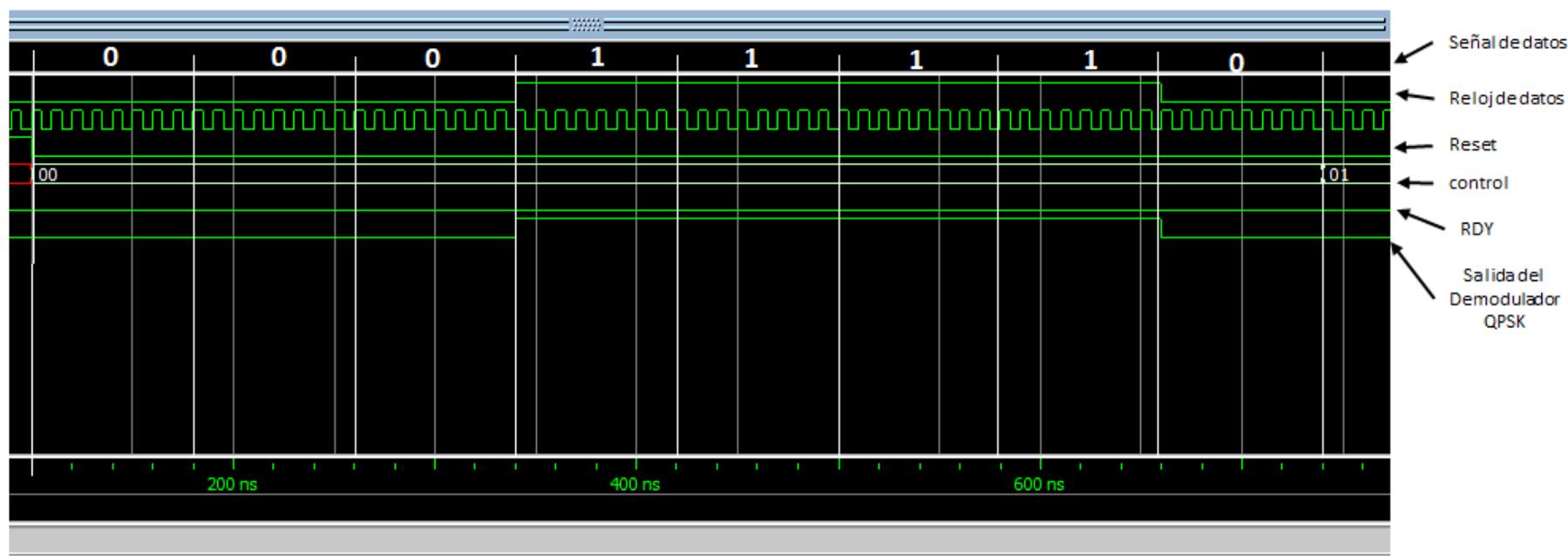


Figura 52 Validación del demodulador QPSK

Podemos apreciar que existe correspondencia entre la señal de datos que entra al modulador QPSK y la que sale del demodulador QPSK.

5.2 Validación del modulador 8PSK

Para la validación del modulador 8PSK, se implementó un circuito digital que fuese capaz de demodular dicha señal. El esquema del mismo se muestra en la Figura 74

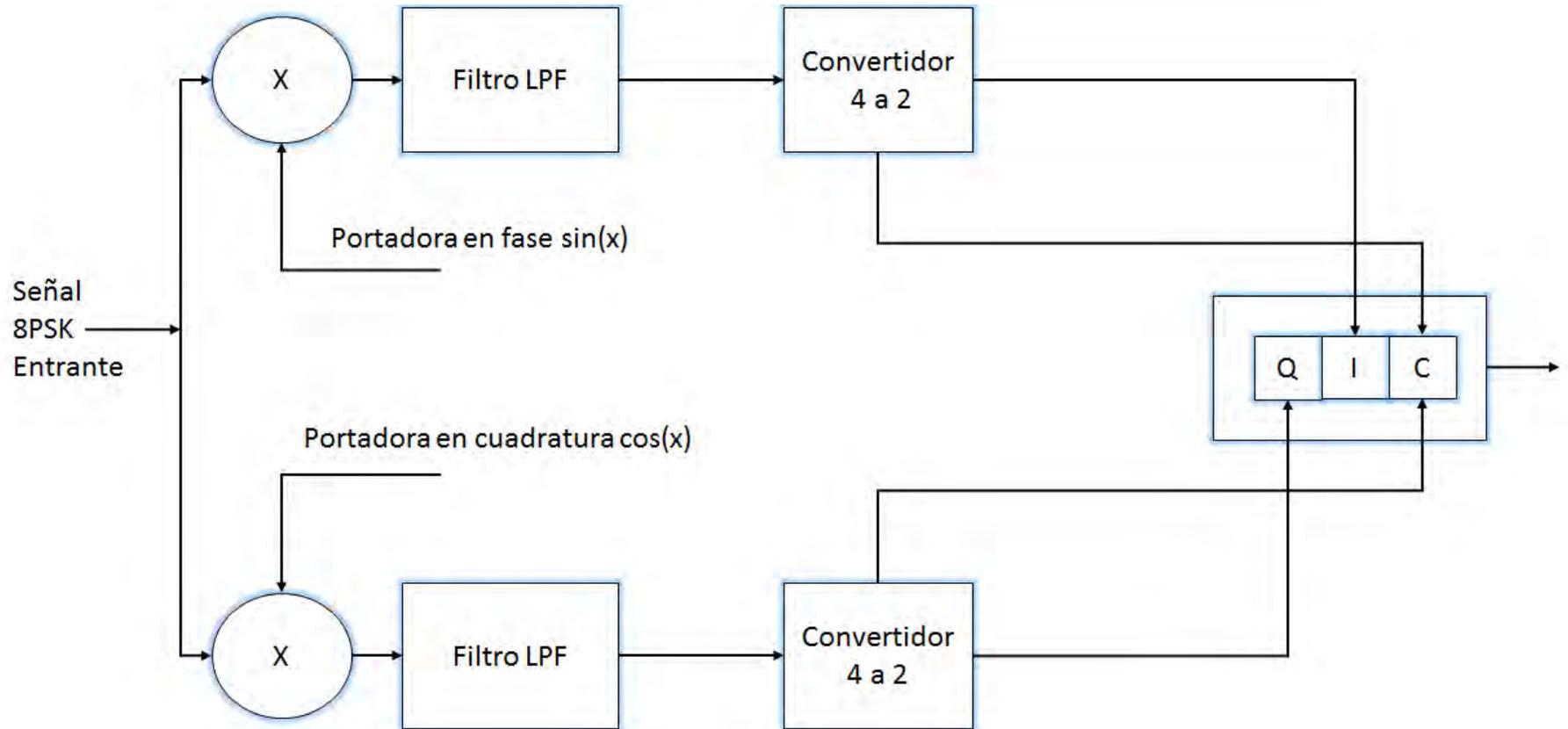


Figura 53 Esquema a bloques del demodulador 8PSK

La señal 8PSK entrante es multiplicada por la portadora en fase para el canal I y por la portadora en cuadratura para el canal Q. A la salida de los detectores de producto se obtiene una señal que contiene varias componentes de alta frecuencia y una constante la cual puede tomar 4 posibles valores; A, B, -A y -B. El filtro pasa bajas elimina las componentes de alta frecuencia, y su salida es procesada por un conversor de 4 a 2 niveles, el cual sustituye al promediador del demodulador QPSK. El conversor genera la salida binaria correspondiente a los niveles obtenidos. La salida de los convertidores es procesada por un bloque paralelo-serie que genera el flujo de salida binario correspondiente. El conversor de 4 a 2 niveles funciona con una lógica inversa al conversor de 2 a 4 niveles. En este caso, el conversor recibe un valor que puede ser positivo o negativo, el cual determina si el bit transmitido es 0 ó 1, mientras que la magnitud del valor determina si el bit C es 0 ó 1. Posterior al conversor el bloque paralelo-serie acomoda los bits recibidos en paralelo en un flujo serial con el orden correcto. El esquema del circuito digital en Vivado se muestra en la Figura 75

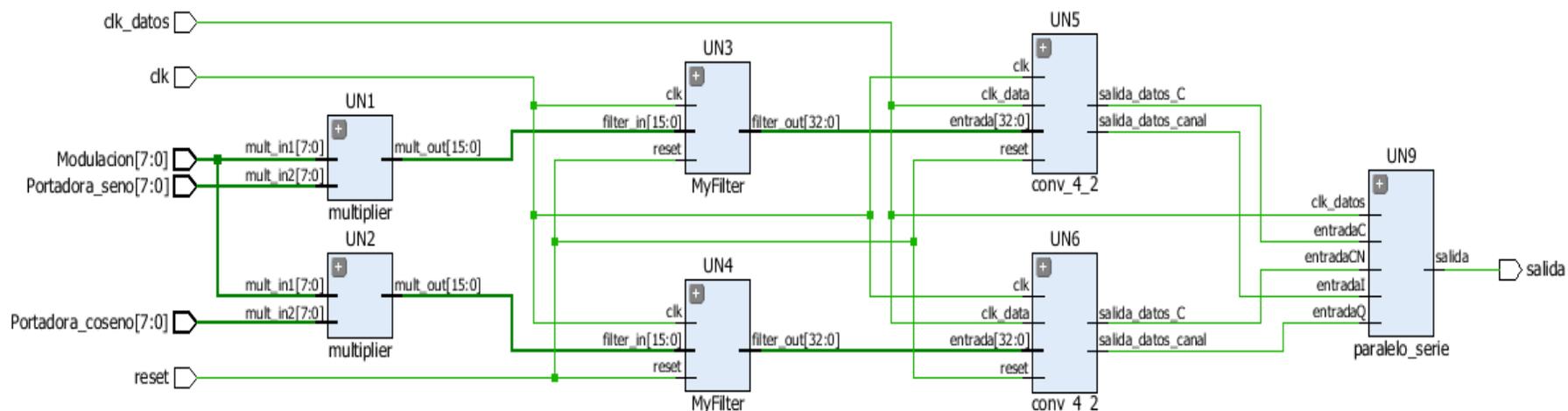


Figura 54 Circuito digital implementado en Vivado del demodulador 8PSK

Para validar el funcionamiento del modulador 8PSK se realiza la simulación usando el software ModelsimPE Student donde podemos visualizar las señales en diferentes puntos del diseño. El resultado de la simulación se muestra en la figura 76.

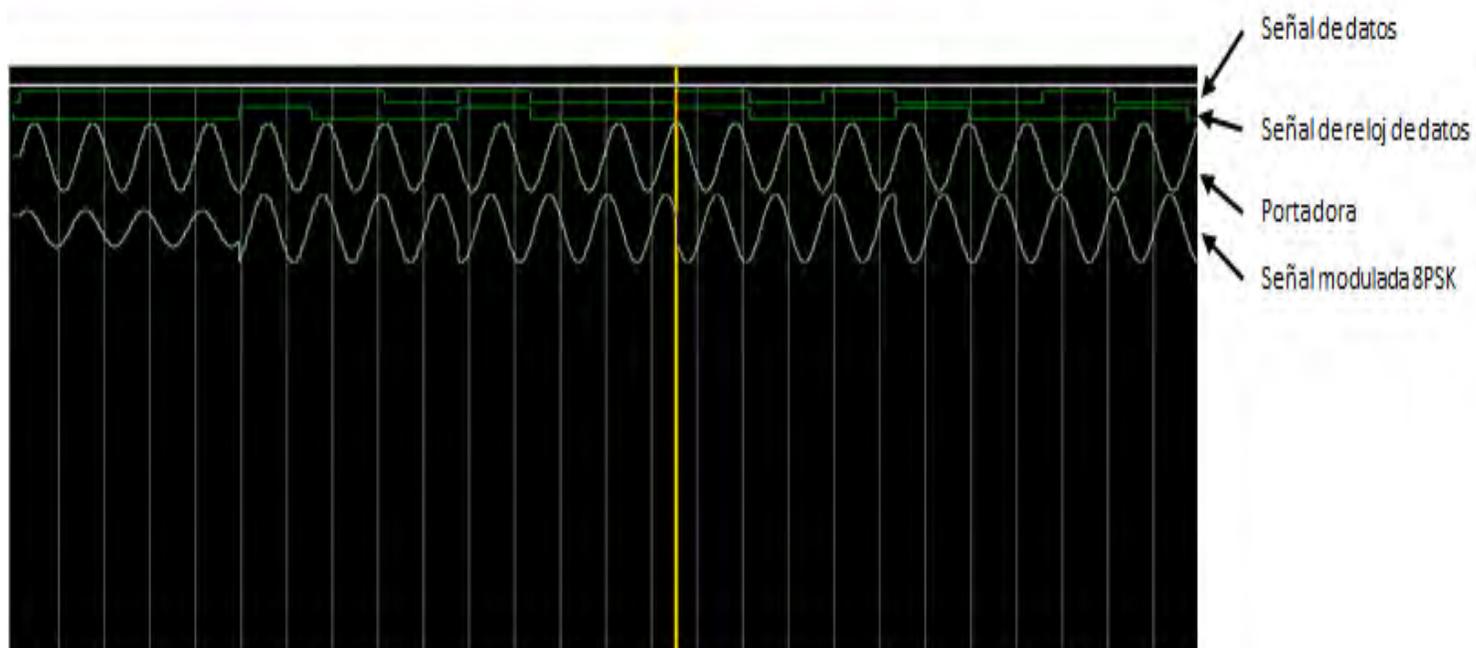


Figura 55 Validación del modulador 8PSK

En la Figura 76 podemos apreciar las características principales de la modulación 8PSK. La primera señal mostrada son los bits de entrada al modulador, la segunda el reloj de datos, la tercera es es la portadora de referencia, y la última la señal modulada. La señal modulada cambia su parámetro de fase con respecto a la portadora, en este caso cada tres bits.

El siguiente paso es realizar la comparación entre los bits de entrada al modulador y los bits de salida del demodulador, como se muestra en la Figura 77.

Las señales presentes en la Figura 77 son:

- La señal de datos entrante al modulador 8PSK
- La señal de reloj de datos
- La señal de selección de la modulación (01 para 8PSK)
- La señal de datos a la salida del demodulador 8PSK.

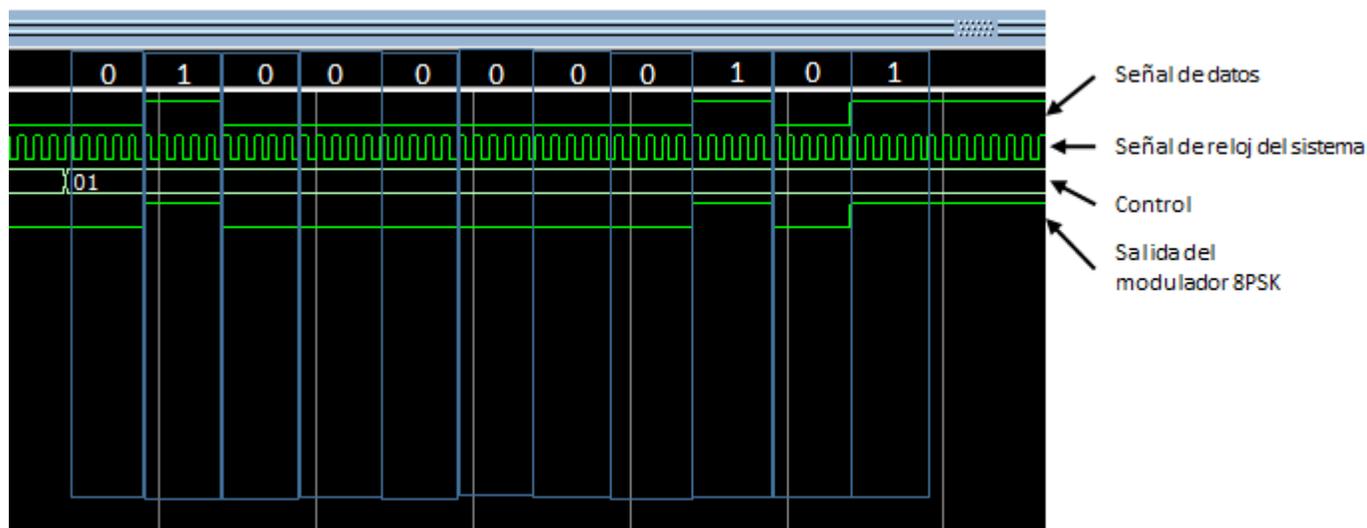


Figura 56 Validación del demodulador 8PSK

Por ser muy larga la secuencia completa de tribits solo se probó con una secuencia de 11 bits, específicamente 01000000101. Podemos apreciar que existe correspondencia entre el flujo de bits que entra al modulador y el generado por el demodulador.

5.3 Validación del modulador 16QAM

Para la validación del modulador 16QAM, se implementó un circuito digital que fuese capaz de demodular dicha señal. El esquema del mismo se muestra en la Figura 5.9. Este demodulador es muy similar al demodulador para 8PSK, la diferencia es que en este caso los convertidores de 4 a 2 niveles tienen valores de conversión distintos. La señal modulada recibida es multiplicada por la portadora correspondiente dependiendo del canal (en fase para el canal I y en cuadratura para el canal Q). El resultado de la multiplicación es filtrado para eliminar las componentes de alta frecuencia para luego tomar la decisión en el bloque conversor de que combinación de bits fue transmitida. Finalmente, un bloque de conversión paralelo serie recibe los datos generados por los conversores y conforma el flujo de datos de salida. El circuito digital implementado en Vivado para estos fines se muestra en la Figura 78.

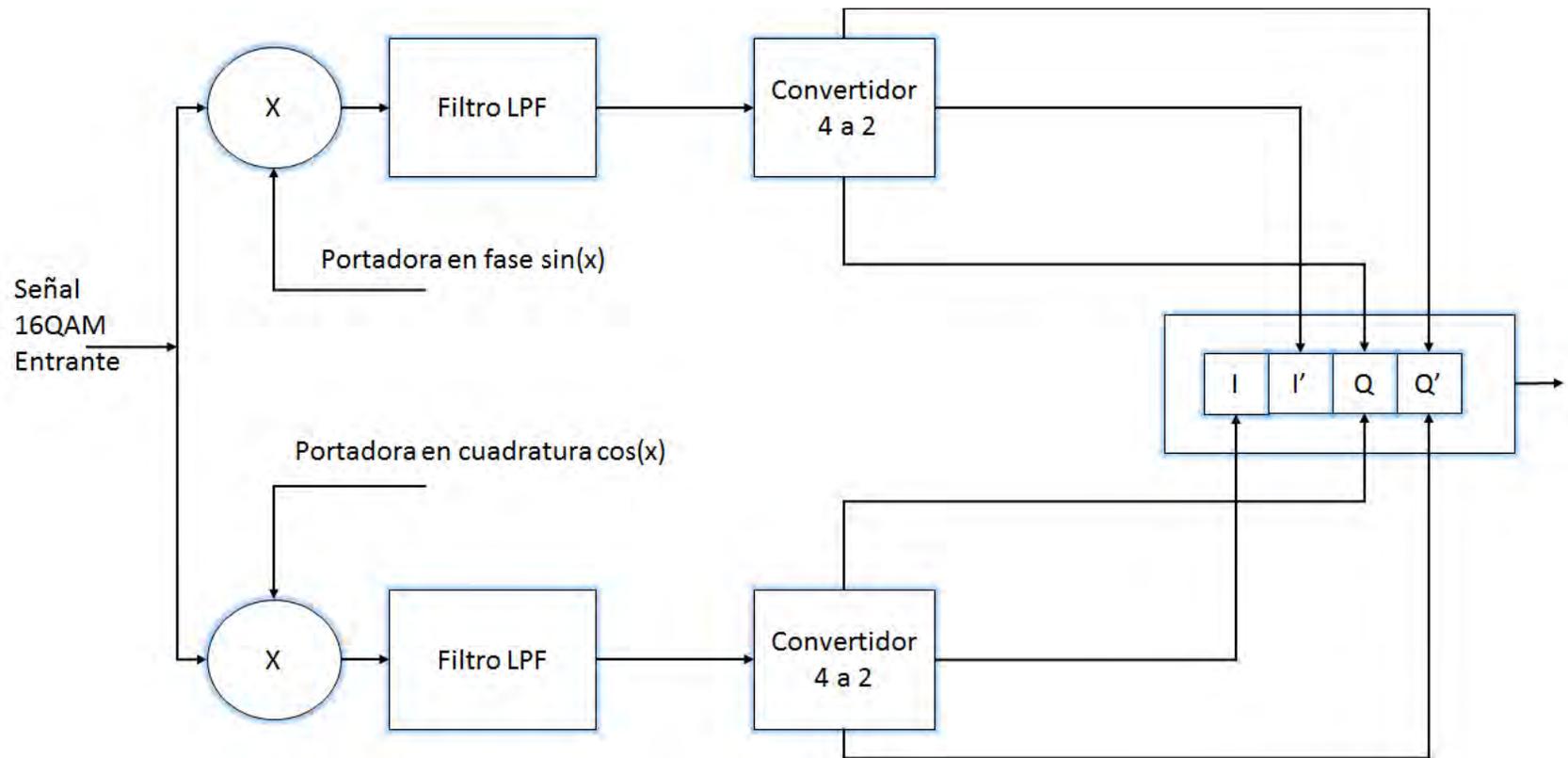


Figura 57 Esquema a bloques del demodulador 16QAM

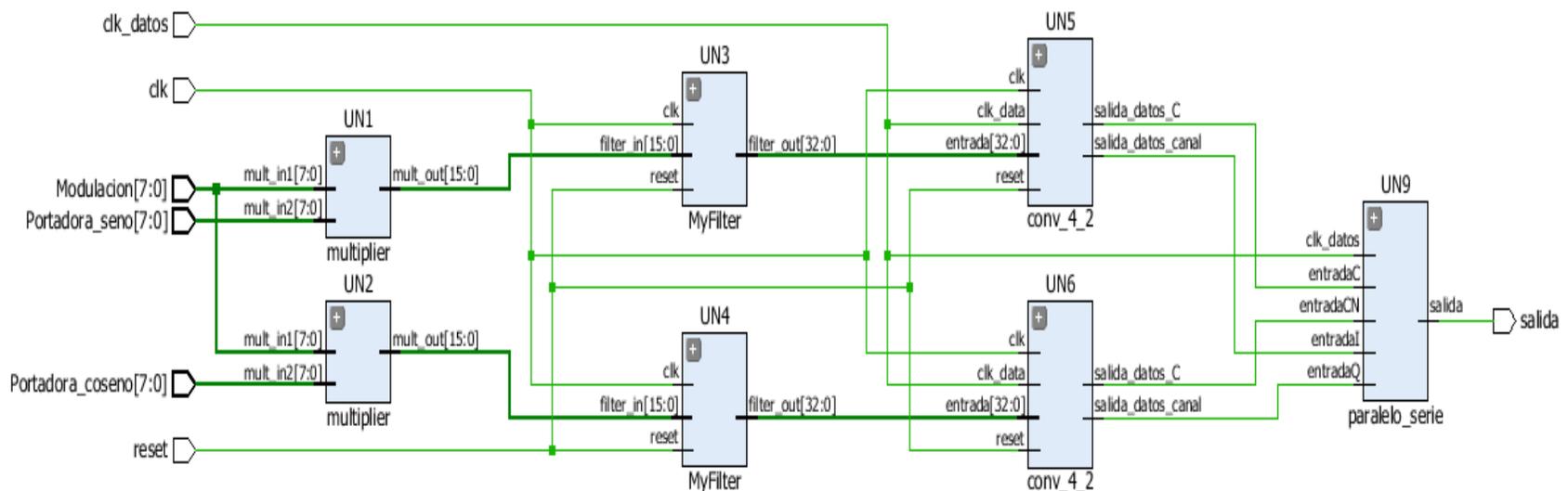


Figura 58 Circuito digital implementado en Vivado del demodulador 16QAM

Para validar el funcionamiento del modulador 16QAM se realiza la simulación usando el software ModelsimPE Student donde podemos visualizar las señales en diferentes puntos del diseño. El resultado de la simulación se muestra en la Figura 5.11.



Figura 59 Validación del modulador 16QAM

En la Figura 80 podemos apreciar las características principales de la modulación 16QAM. La primera señal mostrada son los bits de entrada al modulador, la segunda es la portadora de referencia, y la tercera la señal modulada. La señal modulada cambia sus parámetros de fase y amplitud con respecto a la portadora, en este caso cada cuatro bits.

El siguiente paso es la comparación entre el flujo de bits de entrada del modulador 16QAM y la salida de bits del demodulador 16QAM. Este análisis se muestra en la Figura 81



Figura 60 Validación del demodulador 16QAM

Las señales que se muestran en la Figura 5.12 son:

- La señal de datos de entrada al modulador 16QAM
- La señal de reloj del sistema
- La señal de control para seleccionar la modulación (10 para 16QAM).
- La señal de datos a la salida del demodulador 16QAM

Podemos apreciar que existe correspondencia entre la señal de datos de entrada y la de salida.

5.3 Conclusiones del capítulo

En el capítulo anterior se mostraron las pruebas realizadas al sistema para validar su correcto funcionamiento. Para poder visualizar las señales de entrada y salida de los moduladores nos auxiliamos del software de simulación ModelSim. Se inyectó una cadena de bits a la entrada de los moduladores, y se verificó que a la salida del demodulador correspondiente se obtuviera la cadena de bits correspondiente, lo que nos permitió comprobar que la señal obtenida a la salida del modulador es demodulable. También se realizaron pruebas en cuanto a la forma de onda de la señal modulada, comprobando que cumpliera con las características de la modulación seleccionada.

Capítulo 6 Conclusiones y trabajo a futuro

Conclusiones

Por medio del presente trabajo de investigación se logró el desarrollo de un modulador adaptativo para un sistema de telecomunicación de a bordo de un microsatélite, considerando que los objetivos planteados al inicio del trabajo fueron cumplidos.

Mediante la metodología de análisis del canal de comunicación presentada en el capítulo 2 para estimar el desempeño de las modulaciones seleccionadas para el sistema de a bordo (QPSK, 8PSK y 16QAM), se pudo determinar que mediante la aplicación de la modulación adaptativa se puede maximizar la capacidad de transmisión durante las sesiones de enlaces satelitales en órbita baja.

Se comprobó la posibilidad de aplicar técnicas de tolerancia a fallas, específicamente tolerancia modular triple, mediante la utilización de la herramienta “ByuEdifTool”, lo que permite reducir el tiempo de diseño de las estructuras de redundancia en arquitecturas complejas implementadas en FPGA.

El diseño del modulador adaptativo en un FPGA permite que este pueda ser reconfigurado tanto en su diseño operacional como en su tolerancia a fallas, aun en un aparato espacial en vuelo, esto gracias a la capacidad de reconfiguración de las arquitecturas implementadas en FPGA SRAM, lo que convierte a este proyecto en una propuesta que podría mejorar su desempeño a partir de los resultados de su operación en vuelo y en consecuencia retrasar su obsolescencia.

Trabajo a futuro

- Desarrollar el sistema de control de la modulación a usar basándose en la técnica PSAM propuesta y descrita en el presente trabajo.
- Realizar un estudio comparativo a profundidad entre las diferentes técnicas de tolerancia a fallas mostradas en este trabajo para determinar cuál se ajusta mejor a los requerimientos de nuestro sistema, basándose en elementos como la radiación a la que se encontrará expuesto el sistema, la cantidad de elementos extras necesarios para aplicar redundancia y el nivel de confiabilidad obtenido.
- Implementar el sistema propuesto usando la técnica de tolerancia a fallas determinada luego de realizar el estudio descrito en el punto anterior, para lo cual en el presente trabajo se seleccionó la herramienta a usar, y se muestra un ejemplo del uso de la misma.
- Implementar la detección de portadora en el demodulador

Bibliografía

- 1- Butchart, K. and Dr Braun R. M., “An Adaptive Modulation Scheme for Low Earth Orbit Satellites.”
- 2- Shervin Shambayati, “GMSK Modulation for Deep Space Applications”. California Institute of Technology.
- 3- Murat Bicer S, “A Software Communications Architecture Compliant Software Defined Radio Implementation, in Department of Electrical and Computer Engineering”. Northeastern University. Boston, Massachusetts. 2002.
- 4- Lima Kastensmidt Fernanda, “Fault-Tolerance techniques for SRAM-Based FPGAs”
- 5- Digital Communications
- 6- Simon, Marvin. “Bandwidth-Efficient Digital Modulation with application to deep space communications”. Deep-Space Communications and Navigation Systems, Center of Excellence Jet Propulsion Laboratory, California Institute of Technology.
- 7- Yu, Cao. “Exact symbol rate and total degradation performance of non-linear M-QAM fading channels” Electrical and Computer Engineering Department, Queen's University, Ontario, Canada
- 8- Wei Gao. “Predistorter design for nonlinear high power amplifiers with multikernel adaptive filtering” School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China.
- 9- Maha Cherif Dakhli, “Nonlinear Distortion Cancellation of High Power Amplifier nonlinearity on MIMO MC-CDMA systems over Rayleigh fading channel”. Innov'Com Lab, Sup'Com, University of Carthage, Tunis, Tunisia
- 10- S.A. Juárez, Cáceres. “FPGA-Based Modeling and Design Methodology of a Digital Pre-distortion System for Power Amplifier Linearization” Inst. Tecnol. de Tijuana, Tijuana, Mexico
- 11- Miller Richard, “Adaptive Coding and Modulation (ACM) in the CDM-625 Advanced Satellite Modem ”
- 12- L. Hadj Abderrahmane, “Design of an Adaptive Communication System for Implementation on Board a Future Algerian LEO Satellite.” Centre of Space Techniques, Division of Space Instrumentation, Algeria
- 13- Grayver Eugene, “Software Defined Radio for Small Satellites”.
- 14- Castañeda, Alcalá Daniel Omar. “Implementación de un modulador-demodulador digital QPSK en base a un FPGA como prototipo para un microsatélite” Facultad de Ingeniería Eléctrica UNAM
- 15- System Toolkit Online Help
- 16- James, K. “An Analysis of pilot Symbol Assisted Modulation for Rayleigh Fading Channels”
- 17- Xiaodong, Cai. “Adaptive Modulation with Adaptive Pilot Symbol Assisted Estimation and Prediction of Rapidly Fading Channels”, Dept. of Electrical and Computer Engineering University of Minnesota.
- 18- Martínez, David. “*VHDL, el arte de programar sistemas digitales*”
- 19- Xilinx Online Information <https://www.xilinx.com/products/design-tools/vivado/vivado-webpack.html>
- 20- Siegle Felix, “Mitigation of Radiation Effects in SRAM-Based FPGAs for Space Applications”. University of Leicester
- 21- Sisterna, Cristian, “Fiel Programmable Gate Arrays (FPGAs).”

- 22- H. M. Quinn, “A Test Methodology for Determining Space Readiness of Xilinx SRAM-Based FPGA Devices and Designs” ; ISR-3 Space Data Syst., Los Alamos Nat. Lab., Los Alamos, NM, USA.
- 23- L. Sterpone, “A Novel Design Flow for the Performance Optimization of Fault Tolerant Circuits on SRAM-based FPGA's” Dipt. di Autom. e Inf., Politec. di Torino, Turin.
- 24- Saponara, Sergio “Hardware/Software FPGA-based Network Emulator for High-speed On-board Communications” Dept. Information Engineering, University of Pisa.
- 25- Kranitis, N. “ A single chip dependable and adaptable payload Data Processing Unit”
- 26- Bujalance Aguiar Eduardo José, “ μ Procesadores ESA.Una visión de los μ Procesadores en aplicaciones espaciales.”
- 27- E. Dodd Paul, “Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics”. Senior Member, IEEE.
- 28- NASA Fault Management HandBook, 20012
- 29- http://www.xilinx.com/publications/prod_mktg/CS11XX_TRMTool_Product_Brief_FINAL0806.pdf, Nota Técnica Xilinx TMR Tool
- 30- <https://www.synopsys.com/implementation-and-signoff/fpga-based-design/advanced-application-solutions/high-reliability.html>, Nota Técnica de la herramienta Synopsys Synplify Premier
- 31- Jonathan, J “Synchronization Voter Insertion Algorithms for FPGA Designs Using Triple Modular Redundancy” NSF Center for High-Performance Reconfigurable Computing (CHREC) Dept. of Electrical and Computer Engineering Brigham Young University
- 32- Rashmi, Suthar. “Performance Analysis of Different M-ARY Modulation Techniques in Cellular Mobile Communication” Department of Electronics & Communication, CTAE, MPUAT, Udaipur (Raj.), India.