

PROYECTO FINAL INTEGRADOR

**DESARROLLO DE UN MODULADOR PSK PARA  
TRANSMISIÓN DE TELEMETRÍA SATELITAL**

**Ivette Llanes Reyes**

**Ing. Francisco Denk**

Director

**Ing. Nicolás Tempone**

Co-director

Abril de 2023



Escuela de Producción, Tecnología y Medio Ambiente  
Universidad Nacional de Río Negro  
Argentina



A mis abuelos:

Adelaida, Angel, Atito, Lena y Zoila



# Índice de símbolos

AMBA Advanced Microcontroller Bus Architecture

AOS Advanced Orbiting Systems

CCSDS Consultative Committee for Space Data Systems

DUT Device under test

FIFO First In First Out

FPGA Field Programmable Gate Array

FSM Finite State Machine

GPS Global Positioning System

HDL Hardware Description Language

LFSR Linear-feedback shift register

OSI Open Systems Interconnection

RF Radiofrequency

SWaP-C Size, Weight, Power and Cost

TT&C Tracking, Telemetry and Control



# Índice de contenidos

<b>Índice de contenidos</b>	<b>iii</b>
<b>Índice de figuras</b>	<b>vii</b>
<b>Índice de tablas</b>	<b>xi</b>
<b>Resumen</b>	<b>xiii</b>
<b>Abstract</b>	<b>xv</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Tecnología satelital . . . . .	1
1.2. Aplicaciones de los satélites . . . . .	2
1.3. Pequeños satélites . . . . .	4
1.3.1. Ventajas y limitaciones de los pequeños satélites . . . . .	5
1.4. Estructura de un satélite . . . . .	7
1.5. Objetivo del trabajo . . . . .	10
<b>2. Modulación</b>	<b>13</b>
2.1. Modulación digital . . . . .	14
2.2. Elección de una técnica de modulación digital adecuada . . . . .	16
2.3. Modulación PSK . . . . .	17
<b>3. Dispositivos lógicos programables</b>	<b>21</b>
3.1. Introducción . . . . .	21
3.2. Ventajas de utilizar FPGAs . . . . .	23
3.3. Lenguajes de descripción de hardware . . . . .	23
<b>4. Implementación</b>	<b>25</b>
4.1. Implementación del modulador digital PSK en la FPGA . . . . .	25
4.1.1. Definición de la arquitectura . . . . .	26
4.1.2. Randomizado y agregado de la marca de sincronismo . . . . .	28
4.1.2.1. Randomizador . . . . .	28

4.1.2.2.	Agregado de la marca de sincronismo . . . . .	29
4.1.3.	Codificador . . . . .	30
4.1.4.	Controlador FIFO . . . . .	31
4.1.5.	Moduladores . . . . .	34
4.1.6.	Interfaces AMBA . . . . .	38
4.1.6.1.	APB . . . . .	40
4.1.6.2.	AHB . . . . .	41
4.1.7.	Interconexión entre los bloques del modulador . . . . .	42
4.2.	Obtención de los diagramas de constelaciones . . . . .	43
4.2.1.	Simulación de circuitos para generar señales I/Q . . . . .	43
4.2.1.1.	Descripción del procedimiento . . . . .	44
4.2.2.	Procesamiento numérico para obtener los cambios de fase . . . . .	46
<b>5.</b>	<b>Resultados</b>	<b>49</b>
5.1.	Verificación por simulación del funcionamiento de los bloques del modulador . . . . .	49
5.1.1.	Funcionamiento de los esclavos APB y AHB . . . . .	51
5.1.2.	Funcionamiento del Randomizador . . . . .	53
5.1.3.	Funcionamiento del Codificador . . . . .	54
5.1.4.	Funcionamiento de la FIFO . . . . .	56
5.1.5.	Funcionamiento de los moduladores . . . . .	56
5.1.5.1.	Simulaciones generales . . . . .	57
5.1.5.2.	Verificación de las señales portadoras . . . . .	57
5.1.5.3.	Lectura de la FIFO . . . . .	58
5.1.5.4.	Verificación de los cambios de fase . . . . .	58
5.1.5.5.	Verificación del período de símbolo . . . . .	60
5.1.5.6.	Proceso de elección de portadora . . . . .	60
5.2.	Utilización de recursos . . . . .	62
5.3.	Obtención de los diagramas de constelaciones . . . . .	64
5.3.1.	Proceso de simulación en el LTSpice . . . . .	65
5.3.2.	Obtención de las constelaciones con métodos numéricos . . . . .	71
	<b>Conclusiones y perspectivas</b>	<b>73</b>
<b>6.</b>	<b>Apéndices</b>	<b>75</b>
6.1.	Simulación en el programa ModelSim . . . . .	75
6.1.1.	Simulaciones generales . . . . .	75
6.1.2.	Verificación de las señales portadoras . . . . .	76
6.1.3.	Lectura de la FIFO . . . . .	78
6.1.4.	Verificación de los cambios de fase . . . . .	80

---

6.1.5.	Verificación del período de símbolo . . . . .	81
6.1.6.	Proceso de elección de portadora . . . . .	82
6.2.	Simulación en el programa LTSpice . . . . .	84
6.2.1.	Espectro de las señales moduladas PSK . . . . .	84
6.2.2.	Espectro de las señales filtradas . . . . .	86
6.2.3.	Espectro de las señales I/Q . . . . .	87
6.2.4.	Espectro de las señales I/Q filtradas . . . . .	90
6.2.5.	Visualización de la simulación completa . . . . .	93
6.2.6.	Visualización de los primeros 15 ms de simulación . . . . .	95
6.2.7.	Visualización de los cambios de fase . . . . .	96
	<b>Bibliografía</b>	<b>99</b>
	<b>Agradecimientos</b>	<b>103</b>



# Índice de figuras

1.1. Formación de CubSats en proporción a 1U [1]. . . . .	4
1.2. Esquema del manejo de las señales de TT&C [2]. . . . .	10
2.1. Degradación de pulso en la transmisión [3]. . . . .	14
2.2. Diferentes tipos de modulación digital [4]. . . . .	15
2.3. Rendimiento de diferentes modulaciones PSK, $E_b/N_0$ vs. $BER$ [5]. . . . .	17
2.4. Transición de símbolos OQPSK. . . . .	19
3.1. Arquitectura de una FPGA[6]. . . . .	22
4.1. Esquema general de un sistema de comunicación. . . . .	26
4.2. Relación entre las capas del modelo OSI y las capas CCSDS propuesto por [7]. . . . .	26
4.3. Etapas de procesamiento de la señal dentro de la FPGA. . . . .	27
4.4. Diagrama general de los bloques involucrados en el modulador PSK. . . . .	27
4.5. Orden designado para el envío de la ASM [7]. . . . .	29
4.6. Diagrama de la FSM del randomizador. . . . .	30
4.7. Codificador convolucional utilizado propuesto por [7]. . . . .	31
4.8. Diagrama de la FSM del codificador. . . . .	32
4.9. Configuración de la FIFO. . . . .	33
4.10. Diagrama de la FSM del bloque Llena_FIFO. . . . .	33
4.11. Bloque <i>Moduladores</i> que contiene a los moduladores seleccionables. . . . .	35
4.12. Señales QPSK obtenidas con el bloque generador de portadoras. . . . .	35
4.13. Diagrama de la FSM del bloque generador de portadoras BPSK. . . . .	36
4.14. Diagrama de la FSM del bloque generador de portadoras QPSKs. . . . .	36
4.15. Diagrama de la FSM del bloque generador de portadoras 8PSK. . . . .	37
4.16. Bloques genéricos que conforman los moduladores PSK desarrollados. . . . .	37
4.17. Mapa de constelaciones propuestos por el estándar 401.0-B-27 a) BPSK, b) QPSK y OQPSK, c) 8PSK. . . . .	38
4.18. Diagrama de la FSM del multiplexor BPSK. . . . .	38
4.19. Diagrama de la FSM del multiplexor QPSK. . . . .	39

4.20. Diagrama de la FSM del multiplexor OQPSK. . . . .	39
4.21. Diagrama de la FSM del multiplexor 8PSK. . . . .	40
4.22. Comunicación Maestro-Eslavo en protocolos AMBA. . . . .	40
4.23. Diagrama de la FSM del esclavo APB. . . . .	41
4.24. Diagrama de la FSM del esclavo AHB. . . . .	43
4.25. Obtención de las señales I/Q. . . . .	44
4.26. Ejemplo de señales VCD exportadas por el ModelSim. . . . .	45
5.1. Valores de los <i>enables</i> durante la simulación. . . . .	52
5.2. Funcionamiento del esclavo APB. . . . .	52
5.3. Funcionamiento del esclavo AHB. . . . .	53
5.4. Funcionamiento del randomizador y agregado de la ASM. . . . .	54
5.5. Funcionamiento del codificador. . . . .	55
5.6. Funcionamiento de la FIFO. . . . .	56
5.7. Vista general de la simulación OQPSK. . . . .	57
5.8. Señales portadoras OQPSK. . . . .	58
5.9. Ejemplo de selección de bits en la modulación OQPSK. . . . .	59
5.10. Visualización de cambios de fase en modulación OQPSK. . . . .	59
5.11. Medición del período de símbolo en modulación OQPSK. . . . .	60
5.12. Proceso de selección de la señal portadora OQPSK (a). . . . .	61
5.13. Proceso de selección de la señal portadora OQPSK (b). . . . .	61
5.14. Recursos disponibles en las FPGAs RT ProASIC3 de acuerdo a la hoja de datos [8]. . . . .	63
5.15. Recursos de las FPGAs Zynq UltraScale+ de acuerdo a la hoja de datos [9]. . . . .	64
5.16. Resultado de la síntesis desarrollada en el Synplify. . . . .	64
5.17. Espectro de la señal modulada QPSK. . . . .	65
5.18. Filtro pasabajos Bessel, orden 5, $f_c = 110$ kHz [10]. . . . .	66
5.19. Características del filtro pasabajos Bessel, orden 5, $f_c = 110$ kHz. . . . .	66
5.20. Espectro de las señales QPSK filtradas. . . . .	66
5.21. Espectro de las señales I QPSK. . . . .	67
5.22. Espectro de las señales Q QPSK. . . . .	67
5.23. Filtro pasabajos Bessel, orden 6, $f_c = 20$ kHz[10]. . . . .	68
5.24. Características del filtro pasabajos Bessel, orden 6, $f_c = 20$ kHz [10]. . . . .	68
5.25. Espectro de la señal I QPSK filtrada. . . . .	68
5.26. Espectro de la señal Q QPSK filtrada. . . . .	69
5.27. Señales simuladas QPSK en LTSpice. . . . .	69
5.28. Primeros 15 ms de simulación QPSK en LTSpice. . . . .	70
5.29. Visualización de un cambio de fase QPSK en LTSpice. . . . .	70

5.30. Diagrama de constelaciones a) BPSK y b) 8PSK. . . . .	71
5.31. Diagrama de constelaciones a) QPSK y b) OQPSK. . . . .	71
5.32. Diagrama de vectores a) BPSK y b) 8PSK. . . . .	71
5.33. Diagrama de vectores a) QPSK y b) OQPSK. . . . .	72
5.34. Diagrama de constelaciones a) BPSK y b) 8PSK. . . . .	72
5.35. Diagrama de constelaciones a) BPSK y b) 8PSK. . . . .	72
6.1. Vista general de la simulación BPSK. . . . .	75
6.2. Vista general de la simulación QPSK. . . . .	76
6.3. Vista general de la simulación 8PSK. . . . .	76
6.4. Señales portadoras BPSK. . . . .	77
6.5. Señales portadoras QPSK. . . . .	77
6.6. Señales portadoras 8PSK. . . . .	78
6.7. Ejemplo de selección de bits en la modulación BPSK. . . . .	78
6.8. Ejemplo de selección de bits en la modulación QPSK. . . . .	79
6.9. Ejemplo de selección de bits en la modulación 8PSK. . . . .	79
6.10. Visualización de cambios de fase en modulación BPSK. . . . .	80
6.11. Visualización de cambios de fase en modulación QPSK. . . . .	80
6.12. Visualización de cambios de fase en modulación 8PSK. . . . .	81
6.13. Medición del período de símbolo en modulación BPSK. . . . .	81
6.14. Medición del período de símbolo en modulación QPSK. . . . .	82
6.15. Medición del período de símbolo en modulación 8PSK. . . . .	82
6.16. Proceso de selección de la señal portadora BPSK. . . . .	83
6.17. Proceso de selección de la señal portadora QPSK. . . . .	83
6.18. Proceso de selección de la señal portadora 8PSK. . . . .	84
6.19. Espectro de la señal modulada BPSK. . . . .	85
6.20. Espectro de la señal modulada OQPSK. . . . .	85
6.21. Espectro de la señal modulada 8PSK. . . . .	86
6.22. Espectro de la señal BPSK filtrada. . . . .	86
6.23. Espectro de la señal OQPSK filtrada. . . . .	87
6.24. Espectro de la señal 8PSK filtrada. . . . .	87
6.25. Espectro de la señal I BPSK. . . . .	88
6.26. Espectro de la señal Q BPSK. . . . .	88
6.27. Espectro de la señal I OQPSK. . . . .	89
6.28. Espectro de la señal Q OQPSK. . . . .	89
6.29. Espectro de la señal I 8PSK. . . . .	90
6.30. Espectro de la señal Q 8PSK. . . . .	90
6.31. Espectro de la señal I BPSK filtrada. . . . .	91
6.32. Espectro de la señal Q BPSK filtrada. . . . .	91

---

6.33. Espectro de la señal I OQPSK filtrada. . . . .	92
6.34. Espectro de la señal Q OQPSK filtrada. . . . .	92
6.35. Espectro de la señal I 8PSK filtrada. . . . .	93
6.36. Espectro de la señal Q OQPSK filtrada. . . . .	93
6.37. Señales simuladas BPSK en LTSpice. . . . .	94
6.38. Señales simuladas OQPSK en LTSpice. . . . .	94
6.39. Señales simuladas 8QPSK en LTSpice. . . . .	95
6.40. Primeros 15 ms de simulación BPSK en LTSpice. . . . .	95
6.41. Primeros 15 ms de simulación OQPSK en LTSpice. . . . .	96
6.42. Primeros 15 ms de simulación 8PSK en LTSpice. . . . .	96
6.43. Visualización de un cambio de fase BPSK en LTSpice. . . . .	97
6.44. Visualización de un cambio de fase OQPSK en LTSpice. . . . .	97
6.45. Visualización de un cambio de fase 8PSK en LTSpice. . . . .	98

# Índice de tablas

1.1. Propósitos de los satélites en funcionamiento orbitando la Tierra. . . . .	3
4.1. Descripción de las señales del APB implementado. . . . .	41
4.2. Descripción de las señales del AHB implementado. . . . .	42
5.1. Tiempos configurados para cada una de las simulaciones. . . . .	50
5.2. Correspondencia entre señal de datos APB y modulación seleccionada. . . . .	51
5.3. Tabla de verdad del codificador convolucional. . . . .	55
5.4. Porcentaje de ocupación de los recursos de las FPGAs. . . . .	63



# Resumen

Las modulaciones digitales de bajo orden son la primera opción en aplicaciones asociadas a la transmisión de telemetría satelital. Esta preferencia radica en la capacidad que tienen estos esquemas de proveer robustez al enlace, lo cual es requerido para garantizar la capacidad de control del segmento de vuelo desde la estación terrena. Así mismo, dada la naturaleza de la aplicación, los criterios de SWaP-C también juegan un rol de gran relevancia a la hora del diseño de los sistemas que deberán implementar estas funcionalidades, lo cual se potencia aún más en el segmento de satélites pequeños. El uso de FPGAs se presenta como una alternativa a los sistemas digitales tradicionales, yendo en pos de garantizar el cumplimiento de estos criterios al dar como resultado soluciones menos costosas, más livianas, más pequeñas y que consumen menor cantidad de energía. Además, esta tecnología permite realizar cambios en la implementación sin necesidad de modificar el hardware, lo que representa una gran ventaja en lo que respecta a la flexibilidad y adaptabilidad en los satélites una vez que han sido puestos en órbita. En este trabajo se desarrolla un modulador digital PSK configurable, basado en FPGA para aplicaciones de TT&C en comunicaciones satelitales. Además de los moduladores, en el diseño se incluyen una serie de bloques auxiliares, típicamente utilizados en esta clase de aplicaciones, dedicados al acondicionamiento y formateo de los datos, los cuales incluyen: randomización, codificación convolucional, introducción de marcas de sincronismo y estructuras de almacenamiento. Los datos a modular, así como algunos parámetros de configuración son introducidos al diseño a través de interfaces estándar AMBA AHB y APB, lo que permite la fácil integración del modulador en diseños más complejos. Se implementan testbenches para las diferentes configuraciones del sistema de modulación, los que permiten ejecutar simulaciones por medio de las cuales se realiza la verificación del correcto funcionamiento de los mismos. En el campo analógico, se utiliza el programa LTSpice para realizar simulaciones de circuitos que permiten generar las señales en fase y cuadratura de las señales moduladas. Las mismas se utilizan para obtener los diagramas de constelaciones y de vectores de cada una de las modulaciones, proveyendo así otra alternativa de comprobación del funcionamiento de los moduladores.

**Palabras clave:** DIAGRAMA DE CONSTELACIONES, FPGA, MODULACION DIGITAL, SATELITE, SWaP-C, PSK, TT&C



# Abstract

Low-order digital modulations are the first choice in applications associated with satellite telemetry transmission. This preference lies in the ability of these schemes to provide robustness to the link, which is required to guarantee the control capability of the flight segment from the ground station. Likewise, given the nature of the application, SWaP-C criteria also play a very important role in the design of the systems that must implement these functionalities, which is even more important in the small satellite segment. The use of FPGAs is presented as an alternative to traditional digital systems, going towards ensuring compliance with these criteria by resulting in less expensive, lighter, smaller and less power consuming solutions. In addition, this technology allows changes in the implementation without the need to modify the hardware, which represents a great advantage in terms of flexibility and adaptability in the satellites once they have been placed in orbit. In this work, a configurable, FPGA-based PSK digital modulator is developed for TT&C applications in satellite communications. Additionally to the modulators, the design includes a series of auxiliary blocks, typically used in this kind of applications, dedicated to data conditioning and formatting, which include: randomization, convolutional coding, insertion of synchronism marks and storage structures. The data to be modulated, as well as some configuration parameters are introduced to the design through standard AMBA AHB and APB interfaces, which allows easy integration of the modulator in more complex designs. Test benches are implemented for the different configurations of the modulation system, which allow to run simulations to verify the correct operation of the modulator. In the analog field, the LTSpice program is used to perform circuit simulations to generate the signals in phase and quadrature of the modulated signals. These are used to obtain the constellation and vector diagrams of each modulation, thus providing another alternative to verify the operation of the modulators.

**Keywords:** CONSTELLATION DIAGRAM, DIGITAL MODULATION, FPGA, SATELLITE, SWaP-C, PSK, TT&C



# Capítulo 1

## Introducción general

### 1.1. Tecnología satelital

Debido a la curvatura de la tierra, no es posible establecer comunicaciones a distancias muy largas utilizando antenas ubicadas en dos puntos distantes del globo terrestre. Es por esto que se buscaron formas alternativas para lograr que la señal llegara de uno de estos puntos al otro. En los años 50 se intentó establecer sistemas de comunicación mediante el rebote de señales sobre globos meteorológicos, pero las señales recibidas eran muy débiles por lo que la idea no funcionó. Luego, en 1954, la marina de Estados Unidos construyó un sistema operacional para la comunicación de barcos con la costa mediante señales que rebotaban en la Luna. Este constituyó el primer mensaje transmitido con éxito en un sistema de comunicaciones entre la Tierra y su satélite artificial y, luego se continuó tratando de establecer comunicaciones por esta vía. Ambos casos, la Luna y los globos, representan satélites pasivos, es decir, funcionan como reflectores en los que la señal rebota para dirigirse a otro punto. Sin embargo, la idea de usar la Luna como satélite pasivo no tuvo trascendencia ya que ésta se mantiene por encima del horizonte la mitad del día y su posición con respecto a la Tierra cambia constantemente, lo que hace ineficiente la comunicación.

Esta limitación de la Luna recién mencionada fue superada con la propuesta hecha en 1945 por el escritor de ciencia ficción Arthur C. Clarke, que había descrito un sistema completo de comunicaciones que utilizaba lo que hoy conocemos por satélites geoestacionarios. Clarke más adelante concluyó que estos satélites no eran prácticos debido a la imposibilidad de poner en órbita amplificadores basados en tubos de vacío frágiles y que consumían una gran cantidad de energía.

La invención del transistor cambió todo eso y el 4 de octubre de 1957 se lanzó Sputnik 1 el primer satélite artificial de la historia. Sputnik 1 constituyó además un satélite activo, ya que era capaz de recibir, amplificar, regenerar y retransmitir información. A partir de esto fue aumentando cada vez más la cantidad de satélites

en el cielo, ya no sólo de aquellos que se encuentran en órbitas geoestacionarias, sino también de los que están en órbitas más bajas constantemente dándole vuelta a la Tierra. Por otro lado, todos estos satélites no se utilizan solamente para comunicaciones, sino también para observación de la Tierra, ya sea para investigación meteorológica o de los suelos, entre otras.

Ya en los últimos años, el advenimiento del denominado “NewSpace” y la democratización del espacio ha generado un proceso de apertura y oportunidades para la participación de nuevos actores dentro de este segmento. El cambio de filosofía desde el modelo basado en soluciones de gran complejidad y altos costos, hacia aquel en donde dominan los sistemas pequeños, baratos y de bajo time-to-market, hizo que el espacio dejara de estar reservado a un pequeño grupo de empresas, y que se produzca así la aparición de centenares de startups orientadas al negocio de los micro y nanosatélites.

Producto de esta coyuntura, en el año 2017 se observó un quiebre con un crecimiento superior al 200 % en la cantidad de lanzamientos de micro y nanosatélites con respecto al año anterior, y el comienzo de una tendencia creciente que de acuerdo a las estimaciones pronostica un potencial de mercado superior a 700 lanzamientos para el año 2023.

## 1.2. Aplicaciones de los satélites

Los satélites desarrollan un papel importante en la vida cotidiana de las personas, aunque a veces sus roles no sean fácilmente relacionados con las ventajas que nos proveen. Muchas son las aplicaciones a las que pueden estar enfocados, siendo una de las fundamentales su uso con fines de comunicación. El propósito de estos es retransmitir la señal alrededor de la curva de la Tierra permitiendo la comunicación entre puntos geográficos muy separados. Uno de sus principales usos es la radio, televisión y más recientemente la comunicación vía teléfonos celulares, sobre todo la utilización de datos móviles. Además, los satélites posibilitan la comunicación entre aviones y barcos, entre ellos mismos o con sus respectivas bases de control. Hoy en día, hay varios cientos de sistemas de comunicaciones satelitales que operan en prácticamente todos los rincones del mundo. Estas empresas proporcionan circuitos fijos de telefonía y datos en todo el mundo, difusión de televisión punto a punto, distribución de televisión en red, difusión de música, servicio de telefonía móvil, servicio de navegación, y redes de comunicaciones privadas para grandes empresas, organismos gubernamentales y aplicaciones militares.

Los satélites también son utilizados con el propósito de observar la tierra, tomar ciertas mediciones de esta y utilizarlas con fines de cuidado y aprovechamiento de

APLICACIÓN	CANTIDAD
Comunicaciones	1832
Observación de la Tierra	906
Ciencia de la Tierra	20
Desarrollo/demostración de tecnología	350
Posicionamiento y Navegación	150
Ciencia y observación espacial	104
Otras aplicaciones	10

**Tabla 1.1:** Propósitos de los satélites en funcionamiento orbitando la Tierra.

la misma, como por ejemplo en la agricultura, monitoreo icebergs, pronóstico del clima, monitorear los fenómenos térmicos volcánicos, entre otros. Otra aplicación es su uso en el desarrollo y demostración tecnológica, los cuales tienen como objetivo servir de base para próxima generación de fabricación o mejoramiento de las actuales. Su utilización para posicionamiento y navegación es muy conocida, casi todos los aviones y barcos utilizan el Sistema de Posicionamiento Global (GPS) como complemento de los sistemas de navegación tradicionales. También se puede mencionar el uso para observaciones del universo, dado que la atmósfera interfiere en las observaciones desde la tierra, los telescopios acoplados en aeronaves son capaces de captar mucha más información.

Los ejemplos anteriores muestran algunas de las tantas aplicaciones de la tecnología satelital. Su importancia para la vida del hombre y el futuro de la humanidad ha hecho que la cantidad en el espacio haya aumentado, y lo siga haciendo, como recién se mencionó. Según datos recogidos por United Nations Office for Outer Space Affairs (UNOOSA) en su reporte de 2020, se han lanzado históricamente un total de 11139 satélites. De estos, existen 6542 orbitando la tierra, de los cuales 3372 están en funcionamiento [11]. La distribución de los propósitos de los satélites operativos se puede observar en la Tabla 1.1.

La cantidad de aeronaves lanzadas históricamente representan un gran número dado lo complejo de diseñarlos, desarrollarlos y ponerlos en órbita. Antes del año 2010 se lanzaban entre 60 y 100 satélites anualmente, pero en años recientes ese ritmo creció muchísimo. En el año 2020, 114 lanzamientos pusieron en órbita 1400 aeronaves. Este drástico aumento se debe a la miniaturización, lo que hace posible poner en órbita más de uno al mismo tiempo, reduciendo de manera importante los costos de lanzamiento. Esta variable es significativa a la hora de evaluar el presupuesto total de confección y puesta en funcionamiento. Es por ello que desarrollar plataformas pequeñas es imperante en estos tiempos. La sección siguiente describe todo lo relacionado a los pequeños satélites.

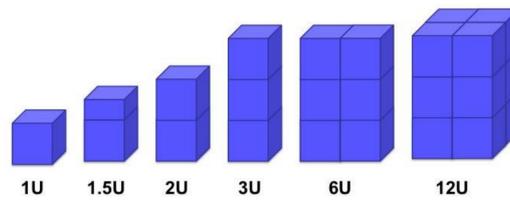


Figura 1.1: Formación de CubSats en proporción a 1U [1].

### 1.3. Pequeños satélites

Si bien el tamaño de los satélites va a depender de la aplicación a la que estén destinados, los avances tecnológicos en el campo satelital están dirigidos a reducir el tamaño lo más posible. De esta manera, como ya se ha mencionado, el costo de lanzamiento es menor. A partir de este criterio surgen los llamados pequeños satélites, que son naves que pesan menos de  $180\text{ kg}$  y tienen un tamaño menor a una heladera [1]. Los pequeños satélites se clasifican según su peso en varios grupos: Minisatélites ( $100\text{-}180\text{ kg}$ ), Microsatélites ( $10\text{--}100\text{ kg}$ ), Nanosatélites, ( $1\text{--}10\text{ kg}$ ), Picosatélites, ( $0,01\text{--}1\text{ kg}$ ) y Femtosatélites, ( $0,001\text{--}0,01\text{ kg}$ ).

Dentro de los nanosatélites existe una clase de satélites llamados CubSats, caracterizados por tener un tamaño y forma establecido, ampliable en escala. El estándar establece que un CubSat base sea un cubo de  $10\times 10\times 10\text{ cm}$ , lo que se conoce por una unidad ( $1U$ ). A partir de esta unidad base se pueden desarrollar CubSat de tamaños mayores ampliados a  $1,5$ ,  $2$ ,  $3$ ,  $6$  e incluso  $12U$ . La Figura 1.1 muestra gráficamente la conformación de CubSats a escala. Los CubSats fueron desarrollados en 1999 con el objetivo de brindar una plataforma para la educación y la exploración espacial. Hoy en día su desarrollo ha avanzado y comprende una industria propia proporcionando una plataforma rentable para investigaciones científicas, demostraciones de nuevas tecnologías y conceptos de misiones avanzadas.

En la Argentina actualmente existen varios proyectos que contemplan el diseño y desarrollo de pequeños satélites. Uno de ellos es la compañía Innova Space, quien puso en órbita el primer picosatélite latinoamericano, surgido a partir de un trabajo práctico en la Escuela Técnica número 5 de Mar del Plata. La empresa provee conexión a través de pequeños satélites para aplicaciones marítimas, ambientales y para la agricultura [12]. Además, el *Ministerio de Ciencia y Tecnología* (MINCyT) y la *Comisión Nacional de Actividades Espaciales* (CONAE) llevan a cabo una iniciativa llamada CANSAT que tiene como propósito desarrollar satélites del tamaño de una gaseosa en ámbitos escolares [13]. También en la Universidad de La Plata se están desarrollando CubSats bajo el proyecto "Satélite Universitario", que tiene como objetivo construir cinco pequeños satélites de un máximo de  $20\text{ kg}$  [14], los cuales se convertirán en los primeros satélites argentinos diseñados y desarrollados en su

totalidad por una universidad.

### 1.3.1. Ventajas y limitaciones de los pequeños satélites

Los satélites pequeños presentan muchas ventajas con respecto a los satélites clásicos. Una de las principales, ya mencionada con anterioridad, es el costo de lanzamiento. Al contar con objetos de menor tamaño, y por consiguiente menor peso, el costo de ponerlos en órbita se reduce significativamente. Además, el costo de producción también es menor. Esto se debe no sólo al costo de los componentes, sino también porque a partir de un modelo inicial se pueden fabricar muchos satélites muy similares, paralelizando la producción. La creación de satélites pequeños también posibilita la inserción de estudiantes en la fabricación de los mismos, desde la concepción inicial hasta casi su lanzamiento; lo que crea recursos humanos bien formados. También cabe resaltar la importancia de los satélites pequeños para aplicaciones como la observación de la tierra o los pronósticos del tiempo, pues al ubicar estos en las órbitas bajas, las mediciones son más precisas. Asimismo, la posibilidad de “independencia” en el espacio hace que los pequeños satélites faciliten una forma asequible para que muchos países consigan capacidad de observación de la Tierra, de defensa y comunicaciones, sin depender de las principales naciones con histórica capacidad espacial.

El desarrollo de los satélites pequeños ha sido posible debido a varias tendencias contemporáneas en el mundo de la tecnología:

- Las últimas décadas han logrado avances en la miniaturización electrónica y la capacidad de rendimiento asociada a estos dispositivos, lo que hace posible que todos los componentes de un satélite se concentren en un espacio mucho menor que el requerido por los satélites tradicionales.
- Lo mencionado en el punto anterior influye en la reducción continua de la complejidad de las misiones, dado que la carga útil de los satélites también está diseñada con componentes electrónicos. Además, también los costos asociados a la gestión, al cumplimiento de las normas de seguridad, etc, se han visto significativamente reducidos.
- La reciente aparición en el mercado de nuevos lanzadores pequeños (por ejemplo, mediante el uso de misiles militares modificados para lanzar pequeños satélites) apoyan la puesta en órbita de los satélites pequeños.
- El desarrollo de pequeñas redes de estaciones terrestres conectadas con métodos de distribución de datos rápidos y rentables.

- Mejoras en el procesamiento de señales y el aumento de la velocidad de cómputo también han representado un papel importante. Los satélites pequeños que están en órbitas bajas tienen que enviar la información a la tierra en el momento justo que pasen por encima de la antena terrestre. Esto hace que la recepción de la información deba ser de manera muy rápida.
- El tiempo de vida que se le exige a este tipo de misiones es mucho menor al de las convencionales. Esto se debe a que al ser más barato el costo de lanzamiento y más rápido el proceso de fabricación, es aceptable que las misiones sean más cortas. Cuando el satélite deje de funcionar, se lanza uno nuevo, inclusive una versión con mejoras o cambios aprendidos de la anterior. Todo esto abre una posibilidad a la innovación, pudiendo utilizarse nuevas tecnologías que para el caso de misiones más largas, requerirían un tiempo de prueba mucho mayor. De esta manera se generan soluciones con una relación costo/rendimiento mucho más alto.

A pesar de los grandes avances tecnológicos que hacen posible el desarrollo de pequeños satélites, el tamaño en sí también presenta algunas limitaciones a la hora del diseño de la estructura física de los mismos. Las partes que conforman un satélite son descritas en la sección siguiente y su clasificación es independiente a las dimensiones de la aeronave, pero es necesario cumplir con los requerimientos de tamaño a la hora de su confección. Un ejemplo de limitación es el poco espacio disponible para ubicar los propulsores y su propelente, encargados de la corrección de la órbita del satélite. Esto hace que la vida útil de los mismos se vea disminuida con respecto a los satélites convencionales, aunque hay algunos CubeSats que llevan casi 10 años en funcionamiento [15]. Otro de los problemas es el volumen de las baterías que conforman los suministros de potencia del satélite. Con ellas pasa lo mismo que con los propulsores, los satélites tienen menos espacio para ubicar estas baterías. Tener menos baterías trae consigo que la plataforma tenga una capacidad limitada para utilizar instrumentos de alta velocidad de datos y/o con alto consumo de energía. Las limitadas capacidades de tamaño y potencia de los satélites pequeños provoca que se tengan opciones restringidas para las combinaciones de instrumentos en una sola plataforma. Además, acotan la posibilidad de incorporar más de una funcionalidad a la carga útil. También las limitaciones de tamaño y de estabilidad de la plataforma no permiten el uso de grandes radares y telescopios. Por lo tanto, la resolución de las mediciones se ven algo comprometidas.

## 1.4. Estructura de un satélite

La estructura de un satélite se divide en dos partes fundamentales: el segmento terrestre y el segmento de vuelo. El primero es donde se encuentran las estaciones encargadas de la Seguimiento, Telemetría y Control (TT&C) necesarias para mantener los satélites operativos, y el segundo está compuesto por la aeronave en sí.

El equipamiento acoplado en una aeronave también se divide en dos grupos según su función. Por un lado está la carga útil, quién está destinada a brindar el servicio determinado para el cual fue lanzado. Ésta es quien lleva a cabo la misión satelital, por lo tanto va a variar en dependencia de los propósitos de cada uno. Por ejemplo, para los casos de observación de la tierra, la carga útil estará equipada con cámaras especiales que tomen fotos de la misma. Para comunicaciones, la misión es actuar como un repetidor, donde la carga útil está compuesta por el *subsistema Comunicaciones*, también conocido como transponder.

El otro grupo de subsistemas es la llamada plataforma de servicios, que engloba todos los componentes que hacen posible el funcionamiento del satélite en sí, sin importar el propósito del mismo, desde la administración de la energía hasta las funciones TT&C. Cada uno de esos subsistemas se describen brevemente a continuación.

El *subsistema Estructural mecánico* posibilita el escenario para el acoplamiento del resto de los subsistemas. Cumple la función de soporte de todos los equipos electrónicos que se llevan a bordo. Como el costo de lanzamiento de un satélite está relacionado con su peso, se quiere que este sea lo más ligero posible. Es por esto, que la estructura no debe ser pesada, pero a su vez debe tener la fortaleza y resistencia necesaria para soportar al resto de los subsistemas y mantener el correcto funcionamiento durante todo el tiempo de vida planificado. La estructura sirve además de pantalla protectora contra las radiaciones energéticas, mezclando así esta función con la esencia del subsistema control térmico. También hace de escudo para el polvo y los micrometeoritos en el espacio, por lo que el material utilizado para cubrir el exterior de la aeronave debe ser resistente a la perforación por estas partículas que se mueven a velocidades muy altas. Además, este subsistema, provee una interfaz entre el satélite y el vehículo de lanzamiento. Desempeña un papel importante al garantizar el funcionamiento fiable en el espacio de ciertos procesos como separarse del lanzador, el despliegue y la orientación de los paneles solares, el apuntamiento preciso de las antenas, el funcionamiento de las piezas giratorias etc.

El *subsistema Suministro de potencia* como el nombre lo indica es el encargado de proporcionar la energía necesaria para el funcionamiento de los equipos electrónicos. Esta energía se obtiene principalmente de la luz proveniente del sol a través de celdas solares extendidas a lo largo de satélite. Existe también un suministro de

energía secundario conformado por baterías que se ponen en funcionamiento cuando la luz del sol no llega a las celdas. Estas baterías son cargadas mientras las celdas solares están trabajando.

Se cuenta además con el *subsistema Control de la actitud*, que se encarga de controlar la orientación de la nave respecto a un sistema de referencia, generalmente un punto determinado en la Tierra. Realizar esta tarea es muy necesario para poder garantizar que las antenas apunten en las direcciones adecuadas. El mantenimiento de la actitud se realiza mediante la medición de la orientación de un satélite en el espacio, si hay algún cambio de los valores de referencia se debe proceder a su corrección.

El *subsistema Propulsión* se encarga de proporcionar los empujes necesarios para imprimir cambios de velocidad indispensables para ejecutar todas las maniobras durante la vida útil. Estas maniobras pueden estar enfocadas en realizar pequeños ajustes de mantenimiento de la estación o el mantenimiento en órbita. Este subsistema apoya las tareas del *subsistema Control de la actitud*, usando propulsores para la estabilización de las aeronaves.

Cuando los satélites están en órbita reciben mucha radiación del sol por uno de sus lados, mientras que no recibe nada por otro que está orientado al espacio. Esto trae consigo una diferencia de temperatura muy grande que hay que manejar para que el funcionamiento de cada dispositivo en el interior no se vea afectado. Además, en el caso de satélites de órbitas bajas, el albedo terrestre puede ser muy significativo. También es necesario extraer la radiación que disipan los equipos internos mientras están operando. Es por esto que es muy importante el *subsistema Control térmico*, que se encarga de manejar las temperaturas en el interior de la nave.

El *subsistema Antenas* es uno de los más sensibles de un satélite cuando nos referimos al diseño. La elección de las antenas responde a varios criterios limitantes: no pueden ser muy grandes, ya que son difíciles de montar y pueden causar problemas estructurales al plegarse. Sin embargo, se quiere tener una antena grande, pues esto implicaría alcanzar una mayor densidad de potencia sin necesidad de aumentar la potencia de transmisión. Este subsistema va a estar compuesto por antenas de diferentes características, contando con antenas omnidireccionales, antenas de cobertura global o terrestre, antenas de cobertura zonal y antenas que producen haces puntuales. Estas últimas pueden tener una orientación fija con respecto a la Tierra, pero además, pueden estar diseñadas para ser dirigidas por comandos remotos. Por lo general, las antenas omnidireccionales se utilizan para las operaciones de TT&C durante la fase en que el satélite ha sido inyectado en su órbita de estacionamiento hasta que alcanza su posición final.

El *subsistema Seguimiento, telemetría y control* es el encargado de comunicarse con las estaciones terrenas para recibir y enviar información necesaria para mantener el

correcto funcionamiento de la aeronave. El seguimiento se considera como la acción de localizar y seguir a los satélites para que el segmento de mando sepa dónde está y hacia dónde se dirige en todo momento. El seguimiento se realiza haciendo que la nave transmita señales de baliza que se reciben en las estaciones terrestres de TT&C. Esta función es muy importante durante las fases de transferencia y deriva orbital del lanzamiento. Para el caso de satélites geoestacionarios, una vez ubicados en su posición puede haber fuerzas perturbadoras que lo muevan de su posición. Es por esto que es imprescindible poder rastrear el movimiento y enviar señales de corrección según sea necesario.

La telemetría es la recogida de información sobre el estado del satélite y sus subsistemas, para luego transmitir estos datos al segmento de mando en tierra. Esto requiere no sólo un sistema de telemetría en la aeronave, sino también una red global de estaciones terrestres en todo el mundo para recoger los datos, a menos que, la red de satélites de aplicación incluya enlaces intersatelitales que sean capaces de retransmitir los datos a un punto central de recogida. Los datos que se transmiten como señales de telemetría incluyen información de actitud, como la obtenida de los sensores solares y terrestres, información medioambiental, como la intensidad y dirección del campo magnético, la frecuencia de impacto de meteoritos, información relacionada a las condiciones específicas de la nave como temperatura de los dispositivos, tensiones de alimentación, presión del combustible almacenado, entre otras similares.

El control se realiza mediante el envío de señales de mando al satélite para que se lleven a cabo acciones específicas como realizar cambios de actitud, conectar y desconectar los transpondedores de comunicación, redirigir las antenas y realizar maniobras de mantenimiento de la estación, etc. La recepción y el procesamiento de los comandos permiten el funcionamiento continuo de la aeronave con el fin de proporcionar el servicio de interés. Mientras que las señales de telemetría son enviadas hacia la Tierra, las señales de control realizan el recorrido inverso. En la Figura 1.2 se muestra un esquema representativo del intercambio de señales de TT&C.

Los enlaces de TT&C son una de las partes más importantes de un satélite, ya que son el único vínculo con tierra una vez que este sea lanzado. A través de esta comunicación se conoce el estado de los mismo, por lo que es posible diagnosticar y resolver posibles situaciones anormales durante el tiempo de operación. Además, como se encargan de los ajustes de ubicación espacial, si se pierde el enlace, es posible que quede orbitando la tierra potencialmente sin visibilidad con las estaciones de control. Esto implicaría perder el dominio total sobre la nave, lo que conllevaría a que la misión fallara con todo lo que eso implica (pérdida de la inversión, necesidad de desarrollar otro satélite, retrasar los objetivos que se perseguían con la misión, etc). También se pierde toda la posibilidad de sacarlo de la órbita en la que quedó,

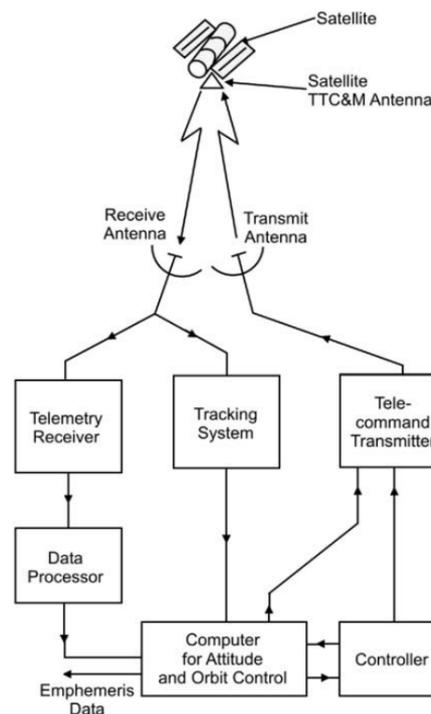


Figura 1.2: Esquema del manejo de las señales de TT&C [2].

lo que supone, en el mejor de los casos, que se transforme en basura ocupando un lugar preciado. En el peor de los casos, puede terminar colisionando contra otro satélite, lo que provocaría la pérdida de un tercero y generando aún más basura.

## 1.5. Objetivo del trabajo

La expansión experimentada por la industria satelital en los últimos años, principalmente en lo que respecta al segmento de los pequeños satélites, ha dado lugar a la creación de un espacio de nuevas oportunidades, así como también ha planteado la necesidad de resolver un conjunto de nuevos desafíos. A diferencia de las fórmulas tradicionalmente establecidas en la industria, las características particulares de este tipo de misiones hace que las soluciones deban ser concebidas a partir de requerimientos muy diferentes. Las grandes limitaciones asociadas a las dimensiones, el peso y el costo, junto a la apertura al uso de tecnologías típicamente ajenas al segmento espacial, constituye un gran empuje hacia el nivel de integración que deben contemplar los diseños que componen a estos sistemas.

Por otro lado, independientemente de las características u objetivos de la misión, siempre será indispensable contar con un sistema de TT&C por medio del cual establecer un vínculo entre los segmentos de vuelo y terrestres. En este contexto es que se enmarca el presente Proyecto Final Integrador, teniendo como objetivo principal del mismo el desarrollo de un modulador digital configurable basado en FPGAs

---

(Field Programmable Gate Arrays), orientado a aplicaciones de transmisión de telemetría satelital. El trabajo propone un enfoque innovador en relación a su entorno de aplicación, buscando mediante el aprovechamiento de nuevas tecnologías brindar una alternativa que pueda contribuir a una mejora en lo que respecta al SWaP-C (Size, Weight, Power and Cost), los cuales son factores claves en este espacio.

Cabe destacar que si bien el desarrollo se encuentra orientado a su utilización en el contexto de las comunicaciones satelitales, también se considera relevante en otros potenciales escenarios que presenten requerimientos similares.



# Capítulo 2

## Modulación

Las señales que llevan información, para ser enviadas a través de algún canal de comunicación, generalmente pasan por un proceso de modulación con el objetivo de acomodar mejor dicha señal a las características del canal. La modulación es definida como el proceso mediante el cual algunas características de una señal, llamada portadora, son variadas en concordancia con la señal moduladora, señal de entrada que lleva la información a transmitir. En el proceso de modulación, se suele llevar la señal moduladora a una nueva ubicación espectral dependiendo de la frecuencia prevista para la transmisión.

Existen varias razones que hacen importante modular la señal antes de transmitirla. Una de ellas es que posibilita la multiplexación, es decir la capacidad de acomodar de manera conveniente varias señales en un mismo canal de comunicaciones. Esta práctica permite enviar más información al tiempo que se administra mejor el espectro radioeléctrico, haciendo un uso más eficiente del mismo. Además, la modulación permite manejar el tamaño deseado para una antena. El tamaño de la antena es directamente proporcional a la longitud de onda de la señal, por lo tanto es inversamente proporcional a su frecuencia. Por ejemplo, si se emiten señales de amplitud modulada (AM),  $535 - 1705 \text{ kHz}$  según la ENACOM en Argentina [16], se necesitan antenas muy grandes, dado que las longitudes de onda se encuentran entre 560 y 176 m. En el caso de las comunicaciones satelitales, donde la información será transmitida en el espacio libre, el traslado a una frecuencia más alta es muy necesario para que la señal pueda ser radiada eficientemente con antenas de tamaño razonable. Esto se hace mucho más evidente en el caso de satélites pequeños. También es importante el comportamiento de la señal a su paso por el canal de comunicaciones. Algunas modulaciones son más robustas ante interferencias y desvanecimiento, lo que hace importante su uso en aplicaciones particulares.

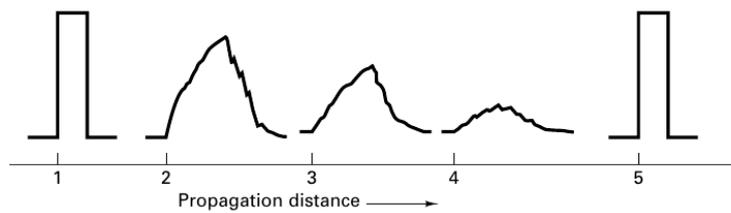


Figura 2.1: Degradación de pulso en la transmisión [3].

## 2.1. Modulación digital

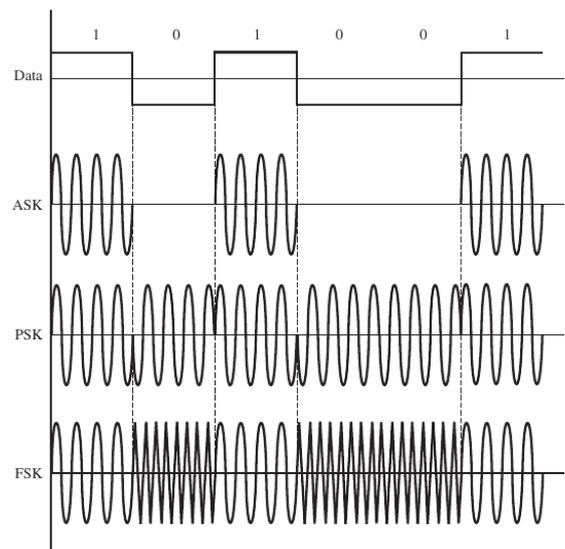
Las técnicas de modulación se dividen en dos grandes grupos, las analógicas y las digitales. En el caso de la modulación analógica, la señal portadora cambia sus parámetros a partir de una señal de información continua. Por otro lado, en la modulación digital los cambios se producen a partir de una señal que toma valores discretos.

Actualmente el mundo ha tendido a la digitalización, aunque todavía se continúan utilizando modulaciones analógicas. Esta preferencia por la modulación digital se debe a que es más rentable que los sistemas de transmisión analógicos [17], lo cual es posible gracias a grandes avances en la tecnología de integración a muy gran escala (VLSI) y en el procesamiento digital de señales (DSP). Los circuitos digitales, además de ser más baratos, son más confiables y el hardware utilizado para resolverlos ofrece una implementación más flexible que para el caso de los analógicos.

La modulación suele utilizarse para trasladar la señal a frecuencias específicas que sean requisitos de un diseño, como puede ser el caso de restricciones en el uso de filtros y amplificadores. Este es el motivo por el cual las señales de radiofrecuencia (RF) son trasladadas en el receptor a una frecuencia intermedia.

También cabe destacar que las señales digitales se reconstruyen mucho más fácil que las señales analógicas [3]. Si se observa la Figura 2.1 se puede ver que la forma del pulso original ha sido afectada durante la transmisión, sin embargo en la etapa 5 se obtiene de nuevo el pulso original. Esto se debe a la forma de las señales digitales, que al tomar valores binarios permiten que se aplique mejor el proceso de regeneración, el cual consiste en demodular, detectar los símbolos y volver a modular. Esta posibilidad de regeneración, llevada a cabo por dispositivos llamados repetidores, hace que las modulaciones digitales presenten mayor inmunidad al ruido y sean más robustas a las deficiencias del canal de comunicaciones.

Las transmisiones digitales, además, pueden incorporar códigos de control de errores que permitan detectar errores en la transmisión, corregirlos, o ambas, lo que las hace mucho más confiables. Asimismo, admiten complejas técnicas de procesamiento de las señales, tales como la codificación y la ecualización, que influyen en mejorar el rendimiento del enlace de comunicación. Por otro lado, también da lugar



**Figura 2.2:** Diferentes tipos de modulación digital [4].

a la aplicación con mayor facilidad de procesos de multiplexación, lo cuales permiten acomodar más canales en un mismo ancho de banda y, por lo tanto, transmitir mucha más información.

Las modulaciones digitales se engloban en tres categorías: modulación por desplazamiento de amplitud (ASK), modulación por desplazamiento de frecuencia (FSK) y modulación por desplazamiento de fase (PSK). Como se puede observar en la Figura 2.2, las señales moduladas ASK representan el cambio de señal binaria (de un cero a un uno o viceversa) con el cambio de la amplitud de la señal. En el caso del ejemplo, un '1' digital se representa con la señal sinusoidal y el '0' con la ausencia de la misma. Cuando el cambio de bit está representado por el cambio de la frecuencia de la señal, se está en presencia de una modulación FSK. Asimismo, los cambios de fase que indican cambios de bits están asociados a la modulación PSK.

Existen casos en los que estas variables (amplitud, frecuencia y fase) toman más de dos valores, lo que permite transmitir más de un bit de información por símbolo. Este tipo de modulaciones se les conoce como M-arias, donde M representa el número de valores que puede tomar la magnitud que está cambiando, es decir, el número de símbolos. La ecuación siguiente muestra la relación entre la cantidad de bits (N) y la cantidad de símbolos de información que se transmiten:

$$N = \log_2 M. \quad (2.1)$$

De la ecuación anterior se ve que contar con mayor número de símbolos posibilita transmitir más bits de información. Es por ello que al utilizar modulaciones M-arias se puede enviar más información en un mismo ancho de banda.

## 2.2. Elección de una técnica de modulación digital adecuada

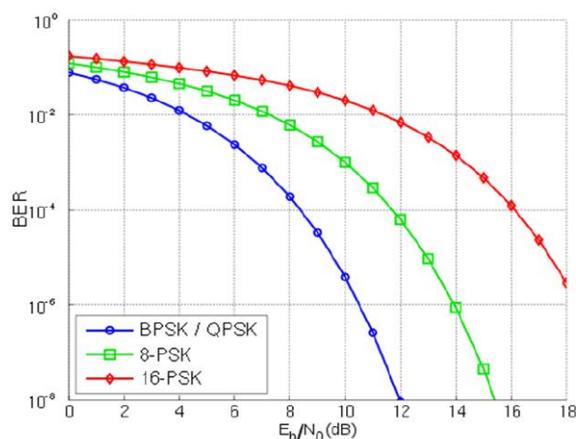
La elección lógica de una técnica de modulación digital para una aplicación específica está influenciada por las características de la señal de mensaje, las características del canal, el rendimiento deseado del sistema de comunicación global, el uso que se hará de los datos transmitidos y los factores económicos que siempre son importantes en las aplicaciones prácticas.

Para analizar cómo una señal puede acomodarse mejor a una modulación u otra, debemos saber las características de las técnicas de modulación. Una de las clasificaciones de las mismas es la linealidad, la cual se da si la amplitud o la fase de la señal transmitida varía linealmente con la señal digital de modulación. Estas técnicas tienen un ancho de banda eficiente, por lo que son muy atractivas para su uso en sistemas de comunicación inalámbricos donde se quiere acomodar gran cantidad de usuarios dentro de un espectro limitado. Además, permiten el uso de amplificadores de potencia lineales, como los de Clase C, que son eficientes en potencia [18], lo que los convierte en deseables para aplicaciones en pequeños satélites. Consecuente con lo que expresa la definición, dentro de las modulaciones digitales lineales podemos encontrar las ASK (Modulación por desplazamiento de amplitud) y las PSK (Modulación por desplazamiento de fase).

En comunicaciones satelitales, los enlaces se ven afectados por el desvanecimiento por trayectos múltiples, ya que las señales se propagan en espacios abiertos. El rendimiento del esquema de modulación bajo varios tipos de degradaciones de canal, como el desvanecimiento de Rayleigh y Rician y la dispersión de tiempo de trayectos múltiples, es un factor clave en la selección de una modulación. Estos efectos producen distorsiones en la amplitud de la onda mientras se traslada entre el emisor y el receptor, por lo que en estos casos es conveniente utilizar modulaciones de frecuencia o fase para no ver afectado el mensaje. A estas modulaciones, como la amplitud de la portadora no varía con la señal moduladora, se les llama modulación de envolvente constante.

Entonces, si se prefiere el uso de modulaciones lineales y con envolvente constante, las técnicas PSK son las más adecuadas. Este es el motivo por el que se decidió para este trabajo el desarrollo de un modulador de este tipo. Ahora queda analizar cuál de todas las variantes de esta modulación serían las más adecuadas para implementar en los enlaces de TT&C.

Teniendo en cuenta lo anterior, otro parámetro que también ayuda a decidir que modulación resulta más conveniente, en dependencia de las demandas de una aplicación en particular, es la tasa de error de bit (BER), característica que representa



**Figura 2.3:** Rendimiento de diferentes modulaciones PSK,  $E_b/N_0$  vs. BER [5].

la fidelidad del mensaje digital. La BER está relacionada con la eficiencia energética, pues ésta describe la capacidad de una técnica de modulación para preservar la fidelidad del mensaje digital a bajos niveles de potencia. En un sistema de comunicación digital, para aumentar la inmunidad al ruido, es necesario aumentar la potencia de la señal. Sin embargo, la cantidad por la cual la potencia de la señal debe aumentarse para obtener un cierto nivel de fidelidad (es decir, una probabilidad de error de bit aceptable) depende del tipo particular de modulación empleada. La eficiencia energética de un esquema de modulación digital es una medida de qué tan favorablemente se realiza esta compensación entre la fidelidad y la potencia de la señal, y a menudo se expresa como la relación entre la energía de la señal por bit y la densidad espectral de potencia de ruido requerida en la entrada del receptor para una cierta probabilidad de error ( $E_b/N_0$ ).

Un esquema de modulación deseable proporciona bajas tasas de error de bit. En la Figura 2.3 podemos comprobar el resultado de un análisis hecho por N. Akcam et al. en [5], donde se comprueba que las modulaciones PSK con menor valor  $M$  presentan un menor BER ante un mismo valor de  $E_b/N_0$  que la modulación con  $M$  inmediato superior. Así se ve que BPSK, QPSK y 8PSK tendrá menor BER que 16PSK en ese orden. Con esta información se puede concluir que para comunicaciones satelitales de TT&C, donde se envía poca cantidad de información y se necesita que el enlace sea robusto, técnicas con bajo  $M$ , como las recién mencionadas, son las más adecuadas.

## 2.3. Modulación PSK

Las señales PSK se representan de forma analítica como se muestra en la ecuación siguiente:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + \phi_i(t)] \quad \begin{array}{l} 0 \leq t \leq T \\ i = 1, \dots, M \end{array}, \quad (2.2)$$

donde la fase, representada por el término  $\phi_i(t)$  tomará distintos valores discretos dependiendo de  $M$  según:

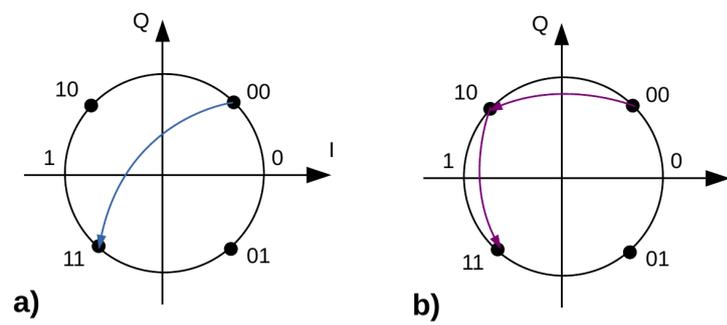
$$\phi_i(t) = \frac{2\pi}{M} \quad i = 1, \dots, M \quad (2.3)$$

Los parámetros  $E$  y  $T$  representan la energía y el período de símbolo, respectivamente.

Entre las diferentes variantes de modulación digital de fase se encuentra la modulación por desplazamiento de fase binaria (BPSK) en la cual  $M = 2$ , lo que significa que la señal de datos moduladora hará que la fase de la onda  $s_i$  tome uno de dos valores separados  $\pi$  entre sí. Otra opción se da cuando la fase puede tomar un valor entre cuatro disponibles, separados cada uno de ellos en  $\pi/2$ . A esta modulación se le conoce como desplazamiento de fase cuádruple (QPSK) y de ella se desprenden otras variantes como la modulación por desplazamiento de fase cuádruple con offset (OQPSK). Esta versión surge de la necesidad de evitar cruces por cero en las transiciones entre símbolos, que se da cuando se cambia a un símbolo que está opuesto en el diagrama de constelaciones. Estos pasos por el origen pueden ser problemáticos, primeramente porque la amplitud de la portadora va temporalmente a cero, lo que causa que la relación entre la potencia media y la potencia pico sea muy alta, complicando así la selección y el diseño de transmisores y amplificadores. También pueden causar problemas de sincronización e incrementar la ocupación espectral. La técnica que se implementa en OQPSK para evitar el paso por cero es retrasar una de las componentes del símbolo por la mitad de un período de símbolo. Eso significa no cambiar más de  $90^\circ$  en una transición. Eso se traduce en que si es necesario cambiar entre dos símbolos opuestos, tal como se representa en la Figura 2.4, en la primera mitad del tiempo de símbolo se transaccionará a un símbolo adyacente y en la segunda mitad, se pasa al símbolo real que se quiere transmitir.

De manera sucesiva se encuentran otras variantes M-PSK, con  $M$  tomando valores 8, 16, 32, 64, ..., etc, donde cada señal portadora puede tener fases que estén separadas  $\pi/4$ ,  $\pi/8$ ,  $\pi/16$ ,  $\pi/32$ , etc, respectivamente, en función de los datos instantáneos a modular.

Como se mencionó anteriormente, los enlaces de TT&C son de vital importancia para mantener al satélite operativo. Debido a todo esto, se necesitan que estos enlaces sean lo más robustos posible, garantizando así la correcta puesta en operación del satélite y su control durante toda la misión. Bajo esta condición, las variantes M-



**Figura 2.4:** Transición de símbolos OQPSK.

arias de más de 8 símbolos no serían las preferidas. Además, por las restricciones de potencia, también conviene hacer uso de las técnicas de modulación más sencillas.



# Capítulo 3

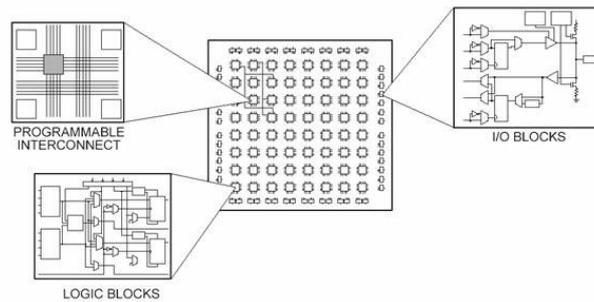
## Dispositivos lógicos programables

### 3.1. Introducción

La tecnología digital ha ido evolucionado a lo largo de los años, desde la lógica discreta hasta los circuitos altamente integrados. Los dispositivos de lógica discreta son diseños que están hechos casi en su totalidad de puertas lógicas, como por ejemplo AND y OR (o sus complementos NAND y NOR). Aunque se puede decir que cualquier sistema digital está hecho de puertas lógicas básicas, este término está reservado para sistemas que están hechos principalmente de componentes discretos individuales como resistencias, condensadores, chips lógicos discretos y chips de funcionalidad única. Con el paso del tiempo ha ido disminuyendo el uso de estos dispositivos ya que son lentos, caros, voluminosos y consumen mucha energía, por lo que no cumplen con los criterios de SWaP-C necesarios. Se debe tener en cuenta que cuando se desarrolla un nuevo dispositivo electrónico, sobre todo para uso aeroespacial, las decisiones de diseño suelen implicar que éste sea más pequeño, más ligero, más potente y/o de menor coste. Lograr incluso uno o dos de esos criterios puede ser motivo suficiente para solicitar un nuevo diseño, aunque lo ideal es lograr un compromiso favorable entre las cuatro áreas.

Por otra parte, los circuitos integrados son los más utilizados en la actualidad. En sus inicios los diseñadores usaban filas de compuertas básicas que eran completamente personalizadas o se hacían con filas estándar de puertas, pero dejando éstas desconectadas. Para un diseño específico, las puertas se conectan a través de líneas metálicas en los canales de cableado. Por lo tanto, la personalización está sólo en las capas de metal del cableado y no en las capas con transistores. Esto se conoce como circuitos integrados específicos de la aplicación (ASIC). El diseño de circuitos como éstos es muy caro y no se puede cambiar fácilmente, lo que conlleva al aumento del uso de la tecnología FPGA.

La FPGA es un circuito integrado compuesto por bloques de hardware internos



**Figura 3.1:** Arquitectura de una FPGA[6].

que tienen interconexiones programables por el usuario. De esta manera, se logra personalizar la operación para una aplicación específica. Se pueden reprogramar fácilmente las interconexiones, permitiendo así que una FPGA se adapte a modificaciones en los diseños o incluso admita nuevas aplicaciones durante la vida útil de la pieza.

Una arquitectura FPGA básica consta de miles de elementos fundamentales que son programados para lograr la funcionalidad requerida. La empresa Xilinx los llama bloques lógicos configurables (CLB), mientras que según otros fabricantes puede denominarse bloques lógicos (LB), elementos lógicos (LE) o celdas lógicas (LC). Los CLBs están rodeados por un sistema de interconexiones programables, llamado tejido, que enruta las señales entre todos ellos. Además, cuenta con una interfaz de bloques de entrada/salida (E/S) entre la FPGA y los dispositivos externos. Esta arquitectura se observa en la Figura 3.1.

Las FPGAs ofrecen múltiples ventajas, por lo que se han convertido en una tecnología con gran nivel de inserción en lo que respecta a electrónica digital. El uso de ellas brinda la posibilidad de independizarse de los fabricantes, ya que las interfaces y los componentes del sistema se pueden adaptar exactamente a nuestros requerimientos. Además, el ciclo de desarrollo de sistemas complejos se acorta por la disponibilidad temprana de prototipos de productos y el hardware desarrollado se puede actualizar en el campo, lo que resulta en un tiempo de comercialización más corto. Como son completamente programables, las actualizaciones y adaptaciones se pueden realizar incluso después de la entrega de un producto.

Para programar la FPGA se configuran los bloques lógicos que conforman la misma como una función específica, que pudieran ser por ejemplo multiplexores o memorias. Para el diseño, los programadores pueden hacer uso de lenguajes de programación específico para esto, los cuales son conocidos como “lenguajes de descripción de hardware (HDL)”. Ejemplos de algunos de estos lenguajes son Verilog, ABEL y VHDL.

## 3.2. Ventajas de utilizar FPGAs

Las FPGAs presentan muchas ventajas frente a otros sistemas de tecnología digital. Una de ellas es la posibilidad de desarrollar a medida el hardware de acuerdo a lo requerido por el proyecto. De esta manera, el diseñador no tiene que buscar todos los dispositivos electrónicos que se adapten al proyecto y armar el equipo final, sino que los diseña desde el inicio a su medida según los requerimientos del mismo.

Además, el desarrollo de un dispositivo específico a partir del diseño en FPGAs, acorta los ciclos de diseño. El circuito es ahora un fichero de texto, que se puede editar, simular, modificar y finalmente sintetizar. Se pueden crear repositorios de hardware, con colecciones de diseños ya probados: UARTs, temporizadores, CPUs, entre otros. El diseñador puede crear prototipos muy rápidamente, probarlos, medirlos y modificarlos.

La flexibilidad en el desarrollo es otra ventaja. Con el mismo hardware físico, conseguimos tener hardware con comportamientos diferentes. Esto es muy importante en tecnología satelital, ya que una vez puesta en órbita una aeronave, no se le pueden realizar cambios de hardware. Sin embargo, con el uso de lógica programable, el mismo hardware que fue enviado al espacio puede rediseñarse desde tierra para cumplir otra función o la misma con mejores prestaciones.

También cabe resaltar la posibilidad de contar con diseños de hardware libres. Con esto se pueden realizar diseños no propietarios, que se compartan dentro de una comunidad y que cualquier diseñador pueda utilizarlos, modificarlos y distribuir las modificaciones. Esto es especialmente útil en el campo de la docencia y la investigación, donde los satélites pequeños encuentran un lugar importante. Esto se potencia aún más si el hardware físico en el que se prueban los diseños es también libre, desapareciendo la dependencia con el fabricante de la placa y cada universidad o diseñador puede fabricarse las placas que considere necesarias.

## 3.3. Lenguajes de descripción de hardware

Los HDL son lenguajes de programación usados para describir la estructura o el comportamiento de circuitos digitales. Algunos HDL surgen en los años 50, pero la creciente complejidad de los circuitos electrónicos digitales en la década de los 70 hizo que los diseñadores de circuitos necesitaran descripciones lógicas digitales que se realizaran a un alto nivel sin estar atados a una tecnología electrónica específica.

En estos años también aparecen lenguajes orientados al uso de aprendizaje universitario como es el caso de AHPL (A Hardware Programming Language) o CDL (Computer Design Language)[19], pero por lo general tenían escaso mantenimiento. Por otro lado se encontraban los lenguajes TI-HDL e IDL, pertenecientes a Texas Ins-

truments e IBM, respectivamente, pero al estar orientados a las industrias no estaban disponibles para terceros [20]. Por los años 1980 surgen los lenguajes de descripción de hardware más utilizados en la actualidad, como son VHDL y Verilog. Los HDL son usados tanto para la implementación de diseños como para simulaciones. Es importante tener en cuenta que hay ciertas estructuras que pueden ser utilizadas para simular un diseño, pero pueden no ser capaces de ser traducidas al hardware físico. Cuando una estructura HDL se puede implementar en un hardware, se dice que es sintetizable.

El modulador digital basado en FPGA de este trabajo fue desarrollado en el programa *Liberio SoC* [21], el cual es una plataforma ofrecida por *Microchip Company*. El software puede ser descargado en la página web de la empresa y la licencia para el uso del mismo puede ser solicitada gratuitamente por un año. El software brinda la posibilidad de hacer la descripción del hardware Verilog o VHDL, siendo este último el escogido para este trabajo.

# Capítulo 4

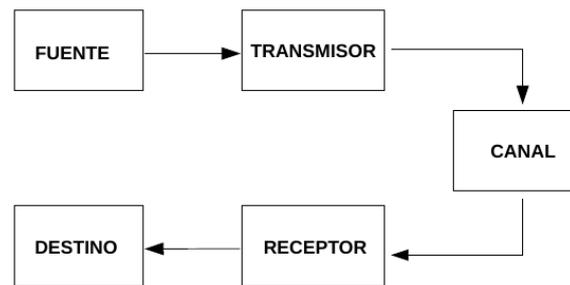
## Implementación

### 4.1. Implementación del modulador digital PSK en la FPGA

De manera general un sistema de comunicaciones está compuesto por tres partes fundamentales: un transmisor, un canal de comunicación y un receptor. En la Figura 4.1 se puede observar un diagrama en bloques del camino secuencial que recorre la información desde inicio a final. La primera parte mencionada, el transmisor, es el encargado de acoplar el mensaje, proveniente de la fuente de información, al canal de comunicación. Para ello, debe realizar diferentes procesos entre los que se encuentran la codificación, modulación, filtrado y amplificación. El canal es el medio donde se transmiten las señales que puede ser un cable, una fibra óptica, el espacio libre, entre otros. Por su parte, el receptor se encarga de convertir las señales que llegan en mensaje de información y entregarlas a su destino. En él también hay etapas de filtrado y amplificación, además de realizar procesos inversos que se llevaron a cabo en el emisor, tales como la decodificación y/o la demodulación. En comunicaciones inalámbricas los transmisores cuentan con una etapa de conversión ascendente (up-converter), un amplificador de alta potencia y una antena. La parte del receptor consta de una antena y un amplificador de bajo ruido (LNA), mientras que la conversión a baja frecuencia se realiza en el demodulador [3].

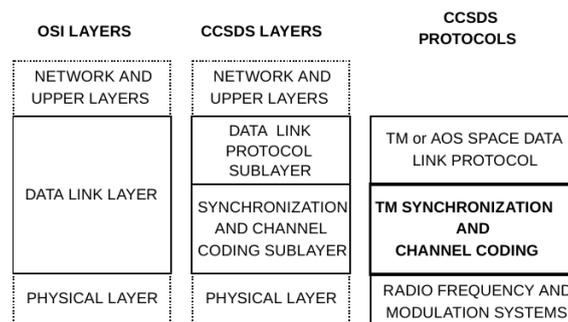
En este trabajo se implementó una de las partes que conforman al transmisor: el modulador. Dado que el uso de éste, aunque no excluyente, está enfocado en la transmisión de telemetría satelital se tomaron como documentos de apoyo los estándares recomendados CCSDS 131.0-B-3 *para sincronización y codificación de canal* [7] y CCSDS 401.0-B-27 *para radiofrecuencia y sistemas de modulación en estaciones terrestres y vehículos espaciales* [22], propuestos por el *Comité Consultivo de Sistemas de Datos Espaciales* (CCSDS).

El CCSDS propone un esquema de capas análogo al modelo de Interconexión de



**Figura 4.1:** Esquema general de un sistema de comunicación.

Sistemas Abiertos (OSI), donde el equivalente a la capa de enlace de datos será particionada en dos subcapas llamadas *Subcapa de Protocolo de Enlace de Datos* y *Subcapa de Sincronización y Codificación del Canal*. A esta última subcapa es que está dedicado el estándar 131.0-B-3, mientras que la análoga a la capa física es tratada en el estándar 401.0-B-27. La relación entre el modelo OSI y la propuesta del CCSDS se puede observar en la Figura 4.2. Si bien este trabajo estará enfocado en la capa física, que es donde se implementa el modulador, también se tienen en cuenta algunas de las sugerencias propuestas en CCSDS 131.0-B-3 como son la codificación o randomizado de la señal.

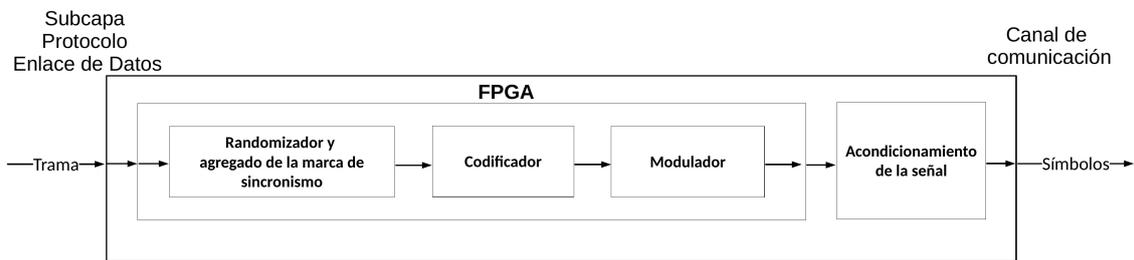


**Figura 4.2:** Relación entre las capas del modelo OSI y las capas CCSDS propuesto por [7].

### 4.1.1. Definición de la arquitectura

La Figura 4.3 muestra un diagrama conceptual de los bloques que conforman el modulador digital desarrollado en este trabajo. Las señales luego de ser procesadas en la *Subcapa Protocolo Enlace de Datos*, llegan a esta sección de la FPGA en forma de tramas. La *Subcapa Sincronización y codificación de canal* será la encargada de realizar tres funciones: codificación de control de errores, sincronización y randomizado. El estándar propone primeramente realizar una codificación Reed-Solomon, Turbo o Comprobación de paridad de baja densidad (LDPC), luego el proceso de randomizado seguido del agregado de una marca de sincronismo y por último, un proceso

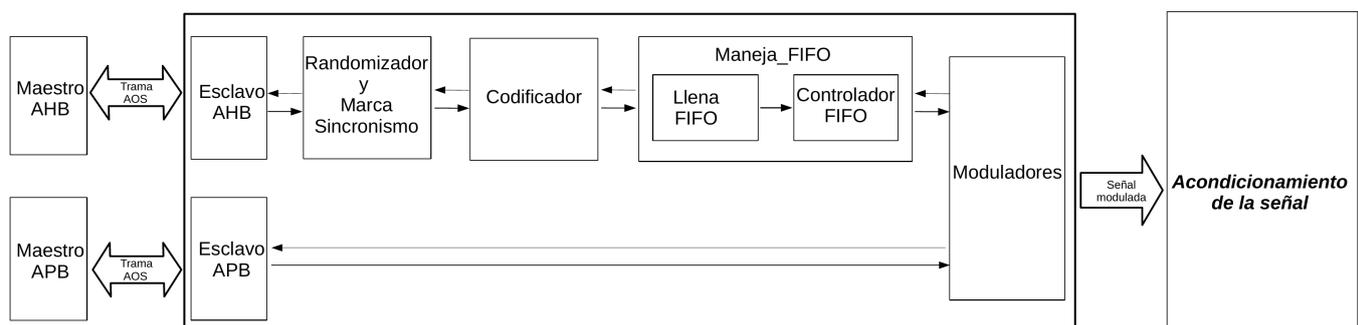
de codificación convolucional o LDPC. Todos estos procedimientos, con excepción del sincronismo, son opcionales para la transmisión.



**Figura 4.3:** Etapas de procesamiento de la señal dentro de la FPGA.

En este trabajo se realiza el randomizado y seguidamente se agrega una marca de sincronismo. Luego, se realiza una codificación convolucional y finalmente se lleva a cabo el proceso de modulación. En la Figura 4.4 se muestra el diagrama conceptual de cada uno de los bloques que conforman el modulador digital PSK desarrollado en este trabajo, los cuales están encerrados en el bloque mayor en el centro de la imagen.

La señal de datos a modular, proveniente de capas superiores, entra al modulador a través de una interfaz AMBA AHB en forma de tramas AOS ( Sistemas avanzados en órbita) y luego pasa por todas las etapas recién mencionadas para obtener a la salida de la FPGA las señales digitales moduladas PSK. Estas señales serán recogidas en una etapa siguiente (independiente a la implementación desarrollada en este trabajo) para su procesamiento analógico y debido acondicionamiento al canal de comunicaciones.



**Figura 4.4:** Diagrama general de los bloques involucrados en el modulador PSK.

El modulador digital PSK desarrollado es configurable en relación al esquema de modulación a aplicar sobre los datos de entrada, es decir la señal puede ser modulada en alguna de cuatro opciones disponibles BPSK, QPSK, OQPSK u 8PSK. La modulación a aplicar, al igual que otros parámetros funcionales del modulador, podrán ser configurados por el usuario por medio de una interfaz APB estándar, donde

estos datos también provienen de capas superiores en forma de trama AOS. A continuación se explicarán en mayor detalle las características y funcionalidades de cada uno de los bloques que conforman el modulador.

## 4.1.2. Randomizado y agregado de la marca de sincronismo

### 4.1.2.1. Randomizador

En un proceso de comunicación las señales que llegan al receptor deben tener una densidad de transición de bits suficiente que garantice el correcto funcionamiento del mismo. La densidad de transición puede definirse como la relación entre los cambios de '1' a '0' (y viceversa) y el número total de bits en una cadena[23]; por tanto cuantas más transiciones se tengan, mayor será la densidad. Un valor grande de este parámetro hace que no se tengan largas cadenas de '1' y '0' , y esto es importante porque el receptor utiliza las transiciones en procesos de sincronización. Además, estas cadenas crean líneas espectrales que pueden interferir en otras transmisiones. El estándar 401.0-B-27 [22] recomienda que la máxima longitud de una cadena de ceros o unos sea 64 bits.

Una manera de aumentar el número de transiciones es haciendo que la señal de datos sea aleatoria. El método propuesto por la CCSDS para asegurar esta aleatoriedad de todas las combinaciones de esquemas de modulación y codificación es el uso de un Randomizador. El estándar CCSDS 131.0-B-3 [7] plantea que se han encontrado problemas en enlaces de telemetría satelital por no haber utilizado randomizadores ni otros métodos que asegurasen la aleatoriedad suficiente de los datos. Por lo tanto, para este trabajo se decidió incluir un bloque randomizador que cumpla con las sugerencias del estándar.

La manera propuesta para garantizar una suficiente transición de bits es realizar una operación XOR a cada bit del bloque de datos o trama de transferencia con una secuencia pseudoaleatoria estándar. La creación de la señal randomizada se hará haciendo la operación XOR entre el primer bit de datos (o primera trama de datos, según corresponda) y el primer bit de la secuencia pseudoaleatoria (o grupo de bits equivalente), luego el segundo bit de datos y segundo bit de la secuencia y así sucesivamente. La secuencia pseudoaleatoria tiene un tamaño finito (256 bits es lo propuesto), por lo que una vez se hayan utilizado todos los bits de la misma, se comenzará otra vez por el primer bit.

Por su parte, en el receptor la señal recibida será reconstruida haciendo el mismo proceso con el uso de la misma secuencia pseudoaleatoria que se utilizó en el proceso de transmisión.

### Cálculo de la secuencia pseudoaleatoria

Una manera de calcular la secuencia pseudoaleatoria es usando un *Linear-feedback shift register* (LFSR), donde la entrada de retroalimentación del registro estará dada por la adición en módulo dos de determinados bits del registro, definidos por un polinomio generador. El estándar CCSDS 131.0-B-3 [7] propone como polinomio generador el mostrado en la siguiente ecuación:

$$h(x) = x^8 + x^7 + x^5 + x^3 + 1. \quad (4.1)$$

Como condición inicial (semilla) se propone asignar '1' a todas las etapas del registro. Una vez implementado esto, obtenemos como salida del registro de desplazamiento la secuencia pseudoaleatoria que se necesita para el randomizador. En este trabajo se implementó el LFSR en Python y la secuencia obtenida, que se muestra a continuación en su forma hexadecimal, se importó en el software Libero SoC.

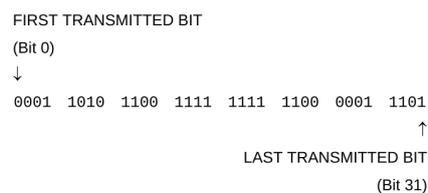
$$FF480EC09A0D70BC8E2C93ADA7B746CE5A977DCC32A2BF3E0A10F18894CDEAB1 \quad (4.2)$$

#### 4.1.2.2. Agregado de la marca de sincronismo

Además de la aleatoriedad en los datos, para un buen funcionamiento del sistema de comunicaciones, debe existir un mecanismo que garantice la sincronización entre transmisor y receptor a nivel trama. El estándar propone el uso de una marca de sincronismo adjunta (ASM) que debe ser colocada entre cada trama si se utiliza un codificador convolucional, que es el caso escogido en este trabajo. La ASM será la marca fija de 32 bits siguiente:

$$1ACFFC1D, \quad (4.3)$$

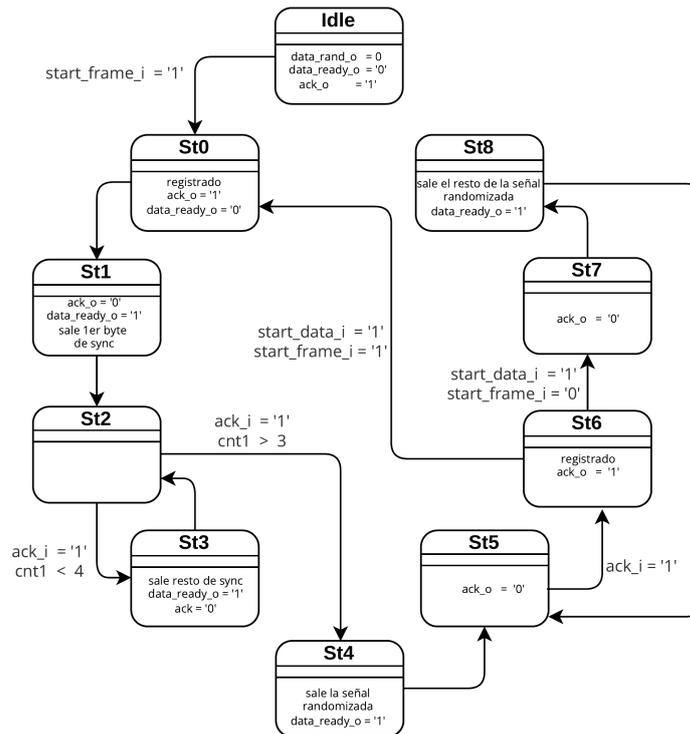
en su representación hexadecimal. El envío de la marca debe comenzar por el bit más significativo de la misma como sugiere la Figura 4.5.



**Figura 4.5:** Orden designado para el envío de la ASM [7].

La marca de sincronismo no debe ser randomizada, por lo tanto este bloque garantiza el envío de toda la ASM, seguido por los datos randomizados. La Figura 4.6

muestra la máquina de estados finitos (FSM) que describe la funcionalidad de este bloque, en la que se observa que primero se realiza el proceso de randomizado y luego se agrega la marca de sincronismo. Las flechas muestran bajo qué condiciones se cambia al estado indicado y la información dentro de cada bloque de estado revela las salidas que ocurren en tal estado. Esta convención es la misma para todas las FSM que aparecerán en este trabajo.



**Figura 4.6:** Diagrama de la FSM del randomizador.

El randomizador está diseñado para recibir ocho bits de datos de señal en forma paralela, por lo que la trama que envían las capas superiores deben ser particionadas en paquetes de un byte. Un señal de entrada al randomizador indica que un nuevo byte está disponible, al mismo tiempo que otra indica el inicio de la trama completa, garantizando así que independientemente del tamaño que tenga la trama se sabrá en qué momento debe ser adjuntada la marca de sincronismo. Este bloque coloca en su salida la marca de sincronismo y los datos randomizados también en grupos de un byte, con su respectiva señal de aviso de que los datos están listos. De esta forma, los datos pueden ser tomados y procesados por el siguiente bloque de la cadena.

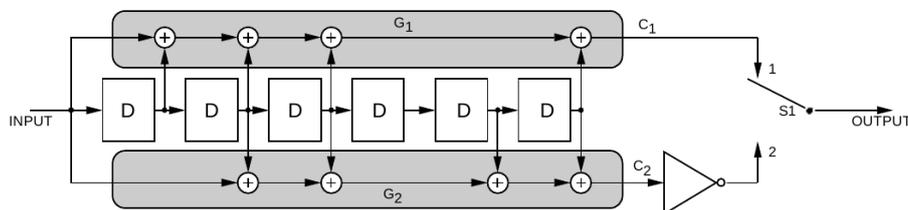
### 4.1.3. Codificador

El codificador utilizado en este trabajo es un codificador convolucional con tasa de código igual a  $1/2$  propuesto por [7]. Los códigos convolucionales son códigos de corrección de errores no sistemáticos, lo que significa que el mensaje de información

no forma parte de la señal codificada, sino que esta será el resultado de operaciones sobre los bits del mensaje [3]. La Figura 4.7 muestra las propiedades del registro de desplazamiento que caracteriza al codificador, en la cual se observa que la longitud de conexiones es igual a siete. Además se pueden identificar los siguientes polinomios

$$\begin{aligned} g_1(X) &= X^6 + X^5 + X^4 + 1 \\ g_2(X) &= X^6 + X^4 + X^3 + X + 1 \end{aligned} \quad (4.4)$$

que representan las conexiones que darán lugar a los bits de la señal codificada.



**Figura 4.7:** Codificador convolucional utilizado propuesto por [7].

La secuencia de símbolos de salida del codificador tendrá la forma siguiente:

$$C_1(1), \overline{C_2(1)}, C_1(2), \overline{C_2(2)}, \dots,$$

La Figura 4.8 muestra el diagrama de la máquina de estados finita que describe el codificador implementado. Este codificador recibe una señal de ocho bits proveniente del randomizador y devuelve una señal codificada de dos bits por cada bit de señal randomizada. Para realizar la codificación, cada bit randomizado entra al registro de desplazamiento mostrado en 4.7 y se calcula la señal codificada a partir de los polinomios descritos en la ecuación 4.4. Una vez que el randomizador avisa que hay datos disponibles, el bloque codificador calcula la primera señal codificada y le entrega estos datos al bloque siguiente, notificando también que hay datos disponibles.

#### 4.1.4. Controlador FIFO

Una vez la señal está codificada llega el momento de modularla y para ello se necesitan agrupar los bits que definirán la portadora. Como la señal de datos es binaria y la modulación BPSK escoge entre sólo dos señales portadoras, necesita un sólo bit para seleccionar la señal portadora. Por otro lado, las modulaciones QPSK y 8PSK necesitan dos y tres bits para poder seleccionar entre cuatro u ocho posibles

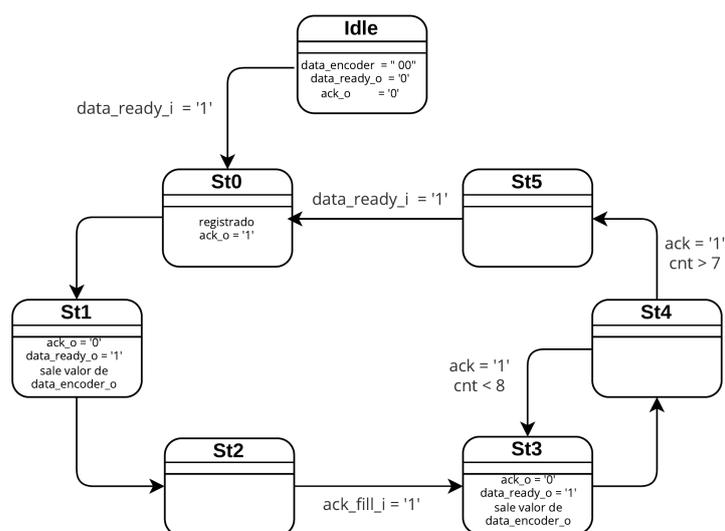


Figura 4.8: Diagrama de la FSM del codificador.

portadoras, respectivamente. El codificador convolucional entrega dos bits en cada proceso de codificación, lo cual funciona perfecto para QPSK ya que son exactamente los bits necesarios para realizar la selección, pero plantea una restricción para las otras dos modulaciones. Una manera de enfrentar la situación es utilizar una estructura de almacenamiento *First In First Out* (FIFO) que permita a cada bloque modulador extraer la cantidad de bits que necesite en el orden correcto.

La FIFO utilizada en este trabajo es un estructura (IP Core) desarrollada por *Microsemi Company*. La compañía proporciona un amplio conjunto de IP Cores que están probados y optimizados haciendo más productivos los diseños de quienes usan sus herramientas. Estos IP Cores son muy fáciles de usar, están disponibles para su implementación en el programa *Libero SoC* utilizado para este trabajo y son incluidos en la licencia. Todas las instrucciones de uso de los IP Cores de Libero son recogidos en el documento *SmartGen Cores Reference Guide* [24] propuesto por *Microsemi*.

El IP Core utilizado es llamado *Controlador Soft FIFO con función de memoria* que puede ser configurado hasta una capacidad máxima de 4096 bits. Posee señales de bandera que notifican si está lleno (FULL), vacío (EMPTY) o si hay datos disponibles para leer (DVLD). La FIFO se configuró para que se reinicie con una señal de reset en bajo y que tenga una capacidad de 1024 bits. El IP Core posee además una señal de escritura y por separado, una señal de lectura de datos llamadas *WE* y *RE*, respectivamente. Los accesos de lectura y escritura pueden realizarse con diferentes ciclos de reloj o usar el mismo para ambas. En la configuración de esta FIFO se estableció que tanto la lectura como la escritura se realicen con el mismo ciclo de reloj. La señal de datos de escritura es llamada *DATA* y la de lectura *Q*. La Figura 4.9 muestra los detalles de configuración de la FIFO utilizada.

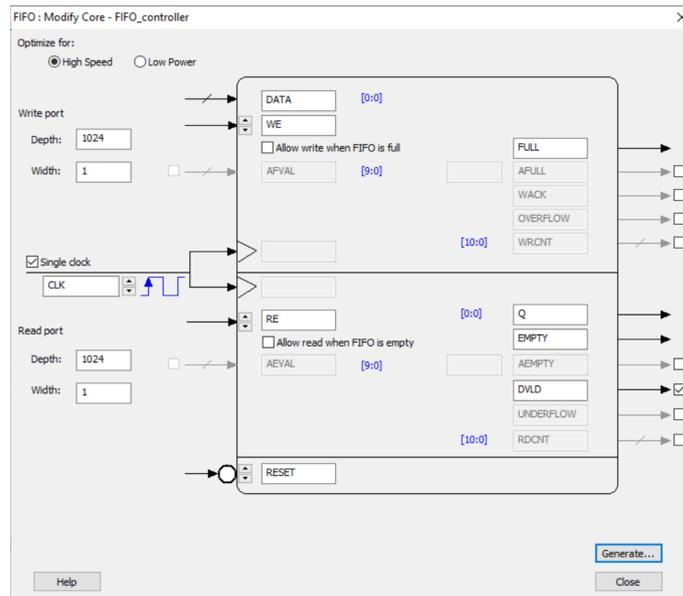


Figura 4.9: Configuración de la FIFO.

Como regla básica de funcionamiento de la FIFO se pide que se cumpla la igualdad representada en la ecuación:

$$W_w \cdot W_d = R_w \cdot R_d , \tag{4.5}$$

donde  $W_w$  y  $W_d$  representan el ancho y profundidad de escritura, respectivamente; y  $R_w$  y  $R_d$  representan el ancho y profundidad de lectura. Es por esto que la FIFO escogida para este trabajo fue configurada para que tanto la lectura como la escritura sea de la misma cantidad de bits, en este caso un bit por acceso.

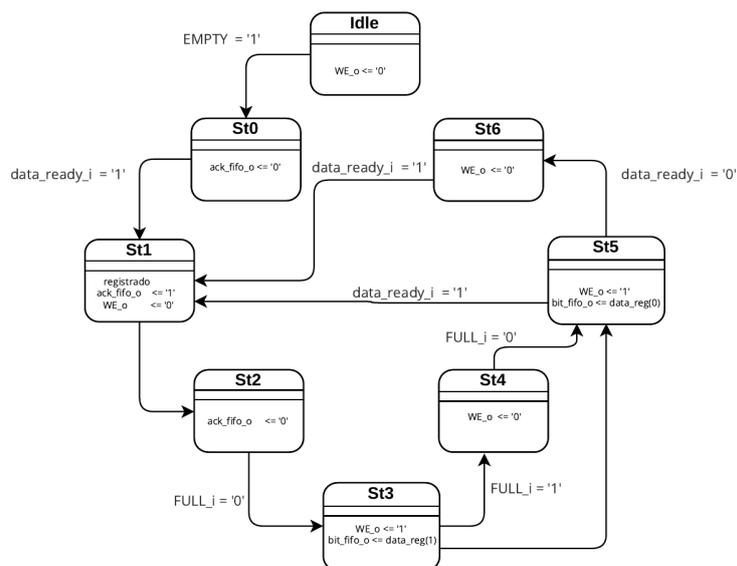


Figura 4.10: Diagrama de la FSM del bloque Llena\_FIFO.

Para este trabajo se implementó un bloque encargado de llenar la FIFO (Lle-

na\_FIFO en Figura 4.4), el cual recibe los dos bits provenientes del bloque codificador y los almacena en la FIFO. El controlador soft está configurado para que no se sobrescriba si está lleno, por lo que el bloque Llena\_FIFO tiene en cuenta no mandar a escribir si la bandera de FULL está activa. El diagrama de la máquina de estados finitos que describe el comportamiento de Llena\_FIFO se muestra en la Figura 4.10.

Un bloque general llamado Maneja\_FIFO (Figura 4.4) conecta al bloque Llena\_FIFO y al IP Core, quedando así este módulo disponible para usar en otras aplicaciones. El bloque se puede usar como caja negra que tiene como entrada una señal de dos bits que es almacenada en una estructura FIFO. La señal de salida de este módulo es una señal de un bit, que es entregada luego de la activación de señal de lectura (RE) de la FIFO.

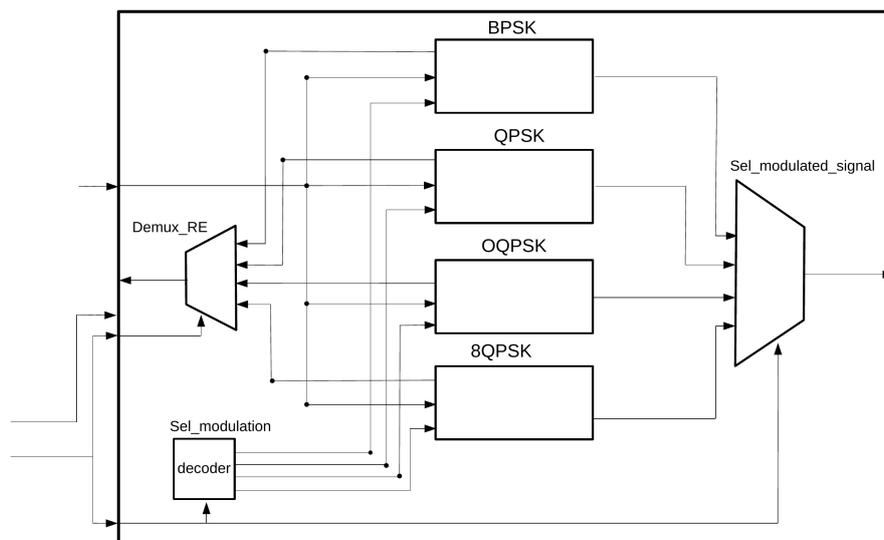
En el diseño de este trabajo la lectura de la FIFO se hace por parte de cada uno de los moduladores, utilizando bloques específicos (*Selección de bits* en Figura 4.16) encargados de esta tarea que se verán más adelante en las sección 4.1.5.

#### 4.1.5. Moduladores

El bloque general que se muestra en la Figura 4.11 se encarga de la modulación de la señal a transmitir. El bloque maneja las cuatro opciones de moduladores desarrollados en este trabajo: BPSK, QPSK, OQPSK y 8PSK. Una señal de entrada de este bloque define cuál de las cuatro opciones de modulador es la que se utiliza. Esta señal de selección entra a un decodificador (*Sel\_modulation* en Figura 4.11) y este la utiliza para poner en alto un señal de *enable* que va hacia el bloque que realizará la modulación deseada, al mismo tiempo que pone en bajo las señales *enable* del resto de los bloques moduladores. Esta señal también es la señal de selección de un multiplexor (*Sel\_modulation\_signal*) que deriva la señal modulada hacia la salida de la FPGA, según cuál sea la modulación seleccionada. Además, habilita a través del bloque *Demux\_RE* qué modulador leerá de la FIFO.

Una de las ventajas de los moduladores desarrollados en este trabajo es la simplicidad con la que se generan las señales portadoras. Mientras que en otras arquitecturas se utilizan algoritmos Cordic y técnicas DDS [25–29] para obtener las portadoras sinusoidales, combinados con moduladores IQ para el proceso de modulación [30], en este trabajo se obtienen señales portadoras cuadradas, con los desfases correspondientes, a partir de la señal de clock de la FPGA. Según los grupos de bits que definen el mensaje, las señal de salida del modulador será una de las portadoras generadas. Una vez que las señales moduladas salen de la FPGA, son filtradas para eliminar las componentes espectrales no deseadas.

Tomando como ejemplo la modulación QPSK la obtención de las portadoras se realiza de la siguiente forma: se definió una máquina de estados finitos que genera

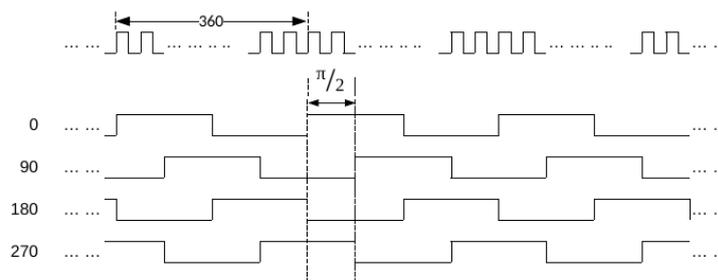


**Figura 4.11:** Bloque *Moduladores* que contiene a los moduladores seleccionables.

cuatro señales cuadradas desfasadas cada una de ellas  $45^\circ$  con respecto a la anterior. Por cada 360 períodos de reloj del sistema ( $T_{clk}$ ) se generará un período de señal portadora ( $T_c$ ). Esta relación, que se expresa en la ecuación siguiente:

$$T_c = 360 T_{clk}, \quad (4.6)$$

se escogió por simplicidad a la hora de generar los desfasajes, ya que 360 es múltiplo de 2, 4 y 8 que son las cantidades de señales portadoras que se necesitan para las modulaciones desarrolladas.

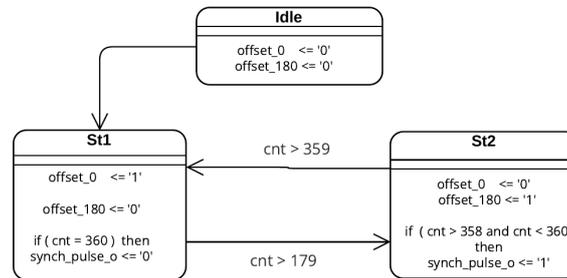


**Figura 4.12:** Señales QPSK obtenidas con el bloque generador de portadoras.

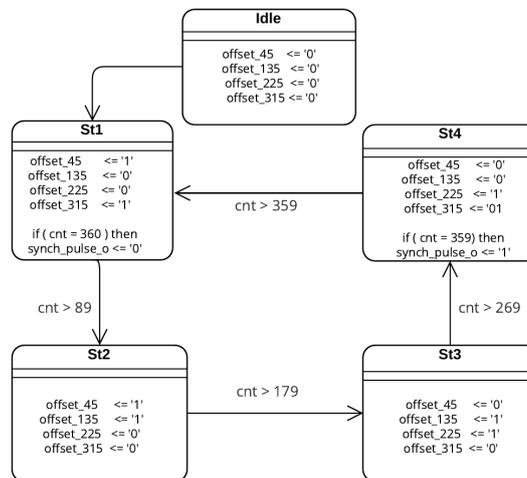
La Figura 4.12 muestra las cuatro señales portadoras QPSK generadas a partir del clock del sistema. Tomando como referencia la señal identificada como '0' en la figura, las otras tres señales estarán desfasadas  $90^\circ$ ,  $180^\circ$  y  $270^\circ$  con respecto a ella. Del mismo modo, para los casos de modulaciones BPSK y 8PSK se generarán señales desfasadas en  $180^\circ$  y  $45^\circ$ , respectivamente. En todos los casos estos bloques generadores de portadora crean un pulso de salida en cada inicio de período que permite la sincronización con el bloque que selecciona la portadora.

Los diagramas de máquinas de estado de cada uno de los bloques generadores

de portadoras BPSK, QPSKs y 8PSK se muestran en las Figuras 4.13-4.15, respectivamente.



**Figura 4.13:** Diagrama de la FSM del bloque generador de portadoras BPSK.



**Figura 4.14:** Diagrama de la FSM del bloque generador de portadoras QPSKs.

Para realizar la modulación, es decir generar una señal portadora modulada en función de la señal mensaje, se necesitan realizar otros dos procesos. Por un lado se necesitan agrupar la cantidad de bits necesarios para escoger la portadora y por otro lado, transmitir por el puerto de salida de la FPGA la señal modulada.

Para cada modulación la selección de los bits se realiza en un bloque dedicado a esta tarea (*Selección de bits* en 4.16) que lee de a un bit por vez de la FIFO y una vez reunida la cantidad necesaria, los pasa agrupados a la etapa siguiente. Tal como se mencionó en la sección 4.1.4, dado que la señal de datos es binaria se va a necesitar un bit para seleccionar la portadora BPSK, dos bits para QPSK y tres bits para 8PSK. La señal modulada será la salida de un multiplexor (*Selección de portadora* en 4.16) que tiene como señal de selección los grupos de bits y que escoge de entre el conjunto de señales portadoras generadas. La conformación por estos tres bloques recién mencionados de cada uno de los moduladores PSK que se ven en la Figura 4.11, puede observarse en el Figura 4.16.

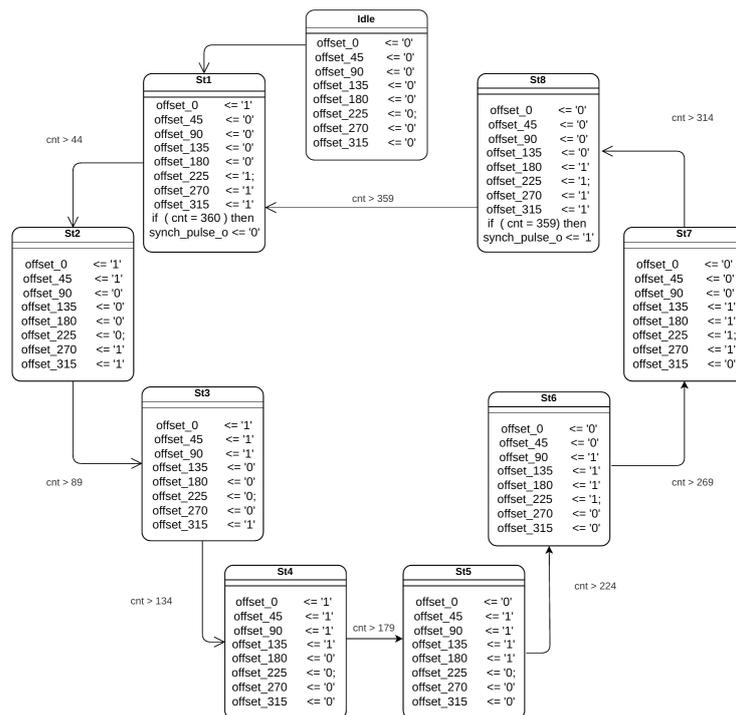


Figura 4.15: Diagrama de la FSM del bloque generador de portadoras 8PSK.

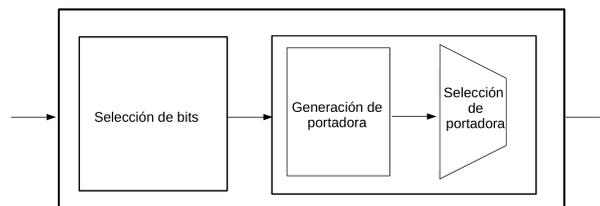
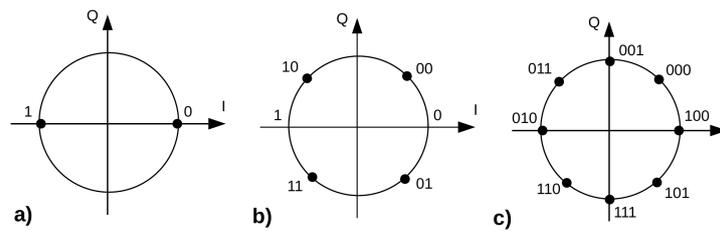


Figura 4.16: Bloques genéricos que conforman los moduladores PSK desarrollados.

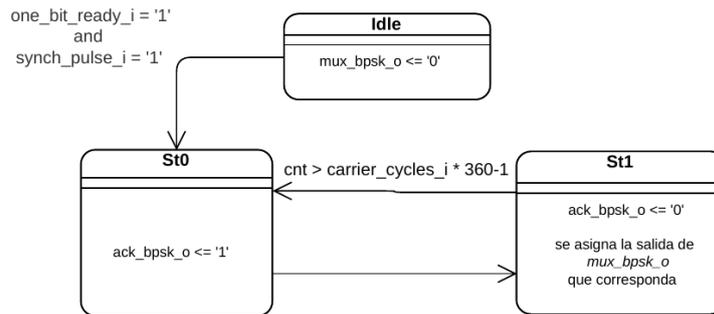
El bloque de selección de bits considera no leer la FIFO hasta que esta tenga datos disponibles. Esto lo hace chequeando la bandera EMPTY, que mientras tenga su valor en '0' significa que no está vacía. Además, este bloque tiene en cuenta, tal como se especifica en el manual de la FIFO [24], que el dato que se indicó leer estará disponible dos ciclos de clock más tarde que cuando se hizo la petición de lectura. La correspondencia entre los bits agrupados y la señal portadora a escoger definida en este trabajo responde a los diagramas de constelaciones sugeridos en el estándar [22], los cuales se muestran en la Figura 4.17.

La selección de la portadora también se realiza a través de máquinas de estado finitos, las cuales se describen en las Figuras 4.18-4.21.

El tiempo de símbolo, es decir el intervalo entre cambios de fase de la portadora, está determinado en las capas superiores y es definido en el bloque *Modulators* (Figura 4.11) a través de una señal de entrada del bloque. Esta información, como también qué modulación se utilizará, llegan a través de una interfaz APB, donde un esclavo APB se encarga de distribuirla a los bloques correspondientes. En la Sección



**Figura 4.17:** Mapa de constelaciones propuestos por el estándar 401.0-B-27 a) BPSK, b) QPSK y OQPSK, c) 8PSK.



**Figura 4.18:** Diagrama de la FSM del multiplexor BPSK.

4.1.6 se explicará con más detalles este procedimiento.

#### 4.1.6. Interfaces AMBA

El modulador digital desarrollado en este trabajo se diseñó para comunicarse con las capas superiores a través de interfaces estándar, en particular las definidas por la *Arquitectura de Bus del Microcontrolador Avanzado* (AMBA). La especificación de AMBA [31] define un estándar de comunicaciones para el diseño de microcontroladores de alto rendimiento que es independiente de la tecnología de los dispositivos que la utilicen. Tiene como uno de sus objetivos fomentar el diseño de sistemas modulares para mejorar la independencia del procesador, y eso es precisamente lo que se desarrolló en este trabajo, un modulador independiente que puede ser incorporado fácilmente como un bloque cerrado en el contexto de otro diseño. El uso de este protocolo estandarizado es lo que precisamente posibilita la inserción del modulador en diseños más complejos de manera sencilla.

El protocolo AMBA engloba tres interfaces de comunicación: Sistema de Bus Avanzado, (Advanced System Bus, ASB), Bus Avanzado de Alto rendimiento (Advanced High-performance Bus, AHB) y Bus Periférico Avanzado (Advanced Peripheral Bus, APB). Todas las arquitecturas AMBA funcionan bajo la arquitectura de Maestro-Eslavo. En la Figura 4.22 se pueden ver las señales que comunican un maestro con un esclavo. Los maestros envían señales de instrucciones y control hacia el esclavo, además de una señal de datos y otra de dirección. Las direcciones

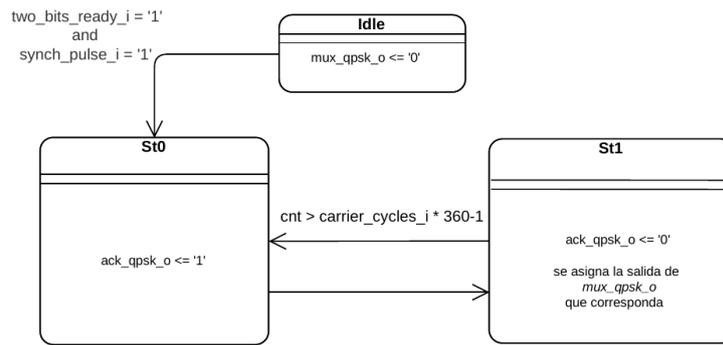


Figura 4.19: Diagrama de la FSM del multiplexor QPSK.

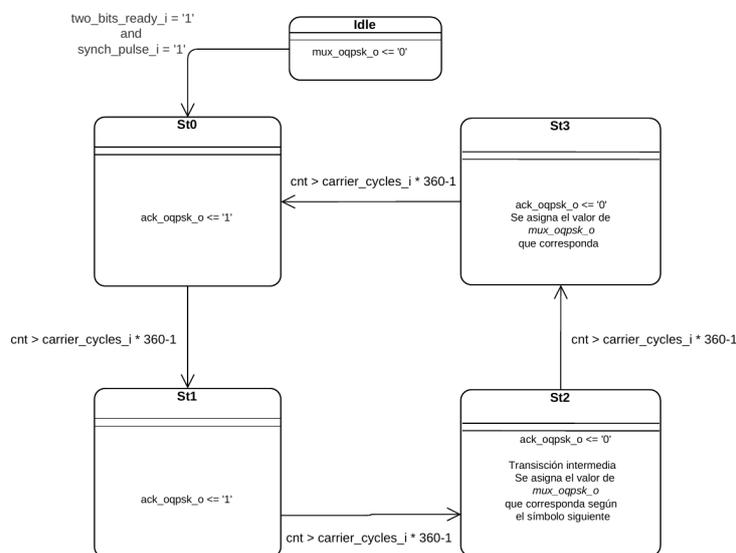


Figura 4.20: Diagrama de la FSM del multiplexor OQPSK.

son decodificadas en el esclavo para que los datos lleguen a su destino dentro del dispositivo o se realice una acción específica en el mismo.

Los buses ASB representan la primera generación de sistemas AMBA. Están pensados para comunicaciones de alto rendimiento, pero no soportan transferencias en ráfagas. Por su parte, los buses AHB son los más complejos, también diseñados para comunicaciones de alto rendimiento. Aceptan transferencias en ráfagas y pueden convivir varios buses maestro en una misma arquitectura. Este bus proporciona una interfaz de gran ancho de banda entre los elementos que intervienen en la mayoría de las transferencias. Los buses APB tienen una complejidad de diseño mucho menor, donde un solo maestro maneja todos los esclavos. Están optimizados para un consumo de energía mínimo, lo que los hace adecuados para el manejo de muchos periféricos.

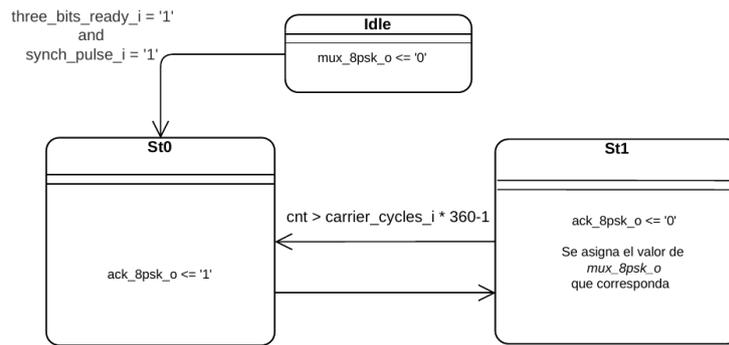


Figura 4.21: Diagrama de la FSM del multiplexor 8PSK.

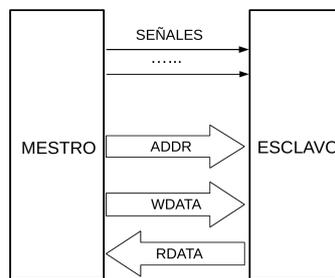


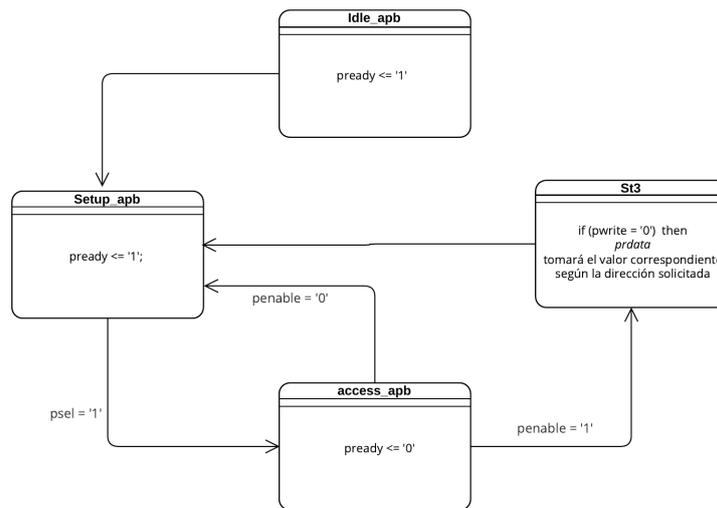
Figura 4.22: Comunicación Maestro-Esclavo en protocolos AMBA.

#### 4.1.6.1. APB

Como se ha mencionado con anterioridad, la modulación a ser aplicada a los datos por transmitir es configurable, así como también la cantidad de ciclos de portadora antes de cambiar de símbolo. Dada la naturaleza de configuración asociada a estos parámetros, sumado a la baja cantidad de intercambio de datos que implican, y a lo esporádico de su transmisión, resulta apropiado el uso de un bus APB para resolver esta interfaz. La máquina de estados del esclavo APB que maneja estas configuraciones se muestra en la Figura 4.23, la cual responde a las características de comunicación propuestas por el estándar *AMBA APB Protocol Specification* [32].

El protocolo propone un conjunto de señales para la comunicación entre el maestro APB y el esclavo APB las cuales se definen en el *Capítulo 2* de [32]. No todas las señales propuestas son de uso obligatorio, la selección de las señales dependerá de la complejidad de la aplicación en la que se usa el protocolo, pero siempre deben cumplir los requerimientos de comunicación propuestos. En este trabajo se establecieron las señales que se muestran en la Tabla 4.1. El ancho de la señal de dirección se configuró a 32 bits, así como el tamaño de las palabras de lectura y escritura.

Las transferencias entre el maestro y el esclavo en un bus APB pueden ser de lectura o escritura y pueden incluir o no *wait states*. Esto quiere decir que el esclavo APB puede realizar la acción solicitada por el maestro en el ciclo de reloj siguiente



**Figura 4.23:** Diagrama de la FSM del esclavo APB.

Nombre	Descripción	Destinatario
PCLK	Señal de reloj	Esclavo
PRESTn	Señal de reinicio, activo bajo	Esclavo
PSEL	Indica que el esclavo a sido seleccionado	Esclavo
PENABLE	Indica que una transferencia está ocurriendo	Esclavo
PWRITE	Indica si es un acceso de escritura (alto) o lectura (bajo)	Esclavo
PADDR	Dirección que debe ser decodificada en el esclavo	Esclavo
PWDATA	Información que debe ser usada en el dispositivo	Esclavo
PREADY	Indica que el esclavo está listo para recibir una nueva tarea	Maestro
PRDATA	Datos solicitados por el maestro	Maestro

**Tabla 4.1:** Descripción de las señales del APB implementado.

a la recepción de la instrucción, o puede tardar varios ciclos en completar la tarea. En el último caso, el esclavo debe avisar que estará trabajando en la asignación y notificar que está listo para recibir una nueva instrucción una vez que haya terminado la actual. En este trabajo el APB está diseñado para trabajar con *wait states*, tardando dos ciclos de reloj en completar la tarea. Las tareas definidas para este esclavo APB son las de establecer la modulación a utilizar y la cantidad de ciclos de portadora por símbolo. Por otro lado, también permite la lectura del valor de estas configuraciones si el maestro las solicita.

#### 4.1.6.2. AHB

Un bus AHB fue implementado como interfaz de entrada para comunicar la señal de datos al modulador digital desarrollado en este trabajo. De forma análoga al bus APB, el estándar *AMBA AHB Protocol Specification* [33] plantea las señales a utilizar y el modo en el que se realizan las transferencias entre el maestro AHB y

el esclavo AHB. Igualmente al caso anterior, se desarrolló un bus AHB simplificado con algunas de las señales propuestas, las cuales se muestran en la Tabla 4.2. El estándar plantea que las señales de datos pueden tener un tamaño de 8, 16, 32, 64, 128, 256, 512 ó 1024 bits, sin embargo los anchos de menos de 32 o mayores de 256 no son recomendados. Es por esto que, si bien el randomizador tiene 8 bits de señal de entrada, la señal de datos del AHB fue configurada para 32 bits. Los ocho bits menos significativos del bus de datos AHB contendrán el mensaje y el bit 8 indicará si hay un nuevo inicio de trama de datos. El ancho de la señal de direcciones está fijado a 32 bits tal como sugiere el estándar.

Nombre	Descripción	Destinatario
HCLK	Señal de reloj	Esclavo
HRESTn	Señal de reinicio, activo bajo	Esclavo
HSEL	Indica que el esclavo a sido seleccionado	Esclavo
HADDR	Dirección que debe ser decodificada en el esclavo	Esclavo
HWRITE	Indica si es un acceso de escritura (alto) o lectura (bajo)	Esclavo
HSIZE	Indica el tamaño de la transferencia	Esclavo
HBRUST	Indica cuantas transferencia se hará en una ráfaga	Esclavo
HTRANS	Indica el tipo de transferencia	Esclavo
HWDATA	Información que debe ser usada en el dispositivo	Esclavo
HRDATA	Datos solicitados por el maestro	Maestro
HREADYOUT	Indica que el esclavo está listo para recibir una nueva tarea	Maestro
HRESP	Notifica si la transferencia se realizó con éxito o si hubo error	Maestro

**Tabla 4.2:** Descripción de las señales del AHB implementado.

El esclavo AHB también está diseñado para trabajar con transferencias *con wait states*, esto se puede ver la Figura 4.24 que muestra la máquina de estados del funcionamiento del esclavo. Este bus AHB no acepta datos en ráfagas, por lo que la transferencia SEQ (ver Sección 3.2 de [33]) no está habilitada.

#### 4.1.7. Interconexión entre los bloques del modulador

Todos los bloques que conforman el modulador digital PSK configurable de este trabajo utilizan procesos de handshaking con sus bloques adyacentes. Como se puede observar en la Figura 4.4 el flujo de información se mueve de izquierda a derecha, por lo que los bloques de la derecha les avisan a sus inmediatos anteriores, a través de una señal de *acknowledgment (ACK)*, que ya tomaron el dato y están listos para recibir uno nuevo. De esta manera los bloques anteriores se liberan y pueden comenzar a preparar la siguiente información. Del mismo modo, una vez que los bloques de la izquierda tienen una información lista, la colocan en la línea de datos y ponen en alto una señal de *ready*. A partir de este acuerdo de comunicaciones entre los bloques, cada uno de ellos espera a que el bloque siguiente le avise que

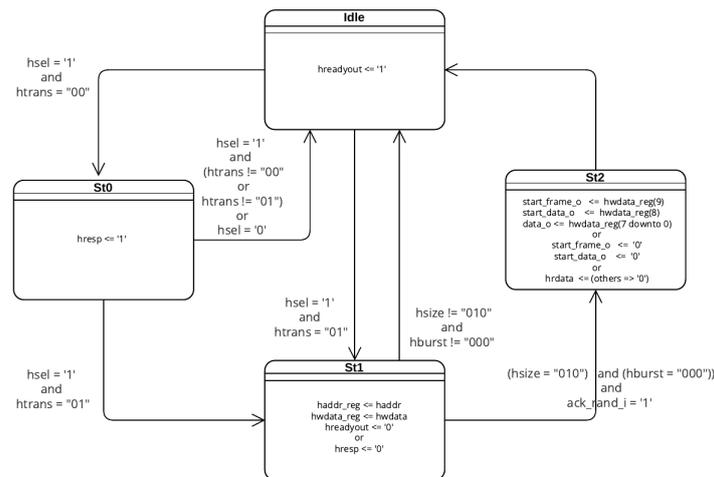


Figura 4.24: Diagrama de la FSM del esclavo AHB.

leyó el dato para solicitar uno nuevo al bloque inmediato anterior. Mientras realiza este proceso de solicitud y preparación del nuevo dato, la línea de *ready* está en bajo. Para este diseño, estos momentos de espera no comprometen la velocidad de salida de los datos modulados porque el proceso de modulación es mucho más lento que el randomizado, codificación y acceso a la FIFO.

## 4.2. Obtención de los diagramas de constelaciones

### 4.2.1. Simulación de circuitos para generar señales I/Q

Los diagramas de constelaciones muestran la magnitud y fase del vector de una señal modulada en cada instante de muestreo. Estos son útiles para visualizar los distintos símbolos para un esquema de modulación digital dado, analizar posibles problemas en la modulación y determinar sus causas. Los errores en la modulación pueden ser causados por el transmisor, el receptor y/o el canal de comunicaciones. Este último puede introducir ruido a la señal causando que los puntos se dispersen alrededor de los símbolos de referencia. Los errores en el transmisor/receptor pueden introducir problemas de amplitud y fase. Cuando se presentan problemas de amplitud los puntos de la constelación se alejan o acercan con respecto al origen de coordenadas. En el caso de errores de fase, los puntos presentarán rotaciones respecto al origen. También se pueden presentar imperfecciones introducidas por los moduladores I/Q que alteran la geometría de la constelación.

A pesar de que los diagramas de constelaciones son obtenidos luego que la señal llega al receptor, una vez que ésta ya haya pasado por todas las etapas de transmisión/recepción y atravesado el canal, son consideradas una buena herramienta

de visualización de las modulaciones. Es por ello que fueron obtenidos para cada una las señales moduladas PSK de este trabajo. Además se incluyen diagramas de vectores que es otra forma de visualizar las señales moduladas vectorialmente. Estos en lugar de mostrar sólo los símbolos, describen el camino recorrido entre los puntos de decisión, por lo que se observan líneas continuas. A menudo se utilizan para distinguir entre diferentes variantes de modulación, como es el caso de QPSK y OQPSK, donde ambas tienen el mismo diagrama de constelaciones pero distinto diagrama de vectores. Este es el principal motivo por el que estos diagramas son incluidos en este trabajo.

Todo el proceso de obtención de las constelaciones que se describe a continuación no forma parte de la implementación que se haría en un transmisor, sino que se utilizan solamente como herramienta de visualización correspondiente a tomar la señal de salida del modulador, aplicar los procesos de conversión hacia abajo en frecuencia y filtrado, de forma tal de poder observar la constelación de la señal y verificar que el esquema de modulación que se obtiene es el esperado. Cabe destacar que las características de las constelaciones no son totalmente representativas desde el punto de vista de la performance del modulador, porque también hay involucrados efectos de este proceso adicional que se realizó sólo a fines de la observación.

#### 4.2.1.1. Descripción del procedimiento

De acuerdo a lo previamente mencionado, una manera de mostrar los símbolos en el diagrama de constelaciones es a través de representar la amplitud y la fase del vector; pero existe otra forma de hacerlo y es en términos de lo que se conoce como señales en fase y cuadratura. También llamadas I/Q, estas señales se obtienen mezclando la señal original con un oscilador local, pero en el camino de las señales Q, la fase del oscilador es corrida  $90^\circ$ . Una representación gráfica de este procedimiento se muestra en la Figura 4.25. Si las señales I/Q se grafican una versus la otra en un eje coordenado, se obtienen los diagramas de vectores y cuando sólo se muestra el resultado en instantes de tiempo determinados, entonces se obtienen los diagramas de constelaciones.

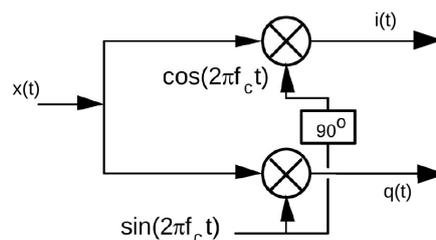


Figura 4.25: Obtención de las señales I/Q.

El procedimiento para obtener las constelaciones se realizó a través de scripts

de *Python* y utilizando el programa *LTSpice* para la simulación de los circuitos. El primer paso es obtener las señales moduladas PSK, lo que se hace a través de la realización de varios testbenches (bancos de pruebas). La simulación, descrita en la Sección 5.1, se lleva a cabo en el software *ModelSim*, el cual es un entorno para simulaciones de diseños en HDLs compatible con el programa *Libero SoC*. El *ModelSim* exporta los datos en formato *.vcd*, el cual tiene la forma que se muestra en la Figura 4.26. Los archivos *.vcd* contienen primero un encabezado con fecha, hora, versión del simulador y el nombre de los bloques involucrados en el proceso. Luego aparecen el valor que toman las señales precedidos por un símbolo de número (#) y en la siguiente línea el momento de tiempo en los que ocurre un cambio de valor de tensión seguido de un signo de admiración (!).

```
1 $date
2
3 $end
4 $version
5
6 $end
7 $timescale
8     1ps
9 $end
10 $scope module top_modulator tb $end
11 $scope module TOP_MODULADOR_0 $end
12 $scope module Modulators_0 $end
13 $scope module BPSK_0 $end
14 $scope module Modulator_BPSK_0 $end
15 $scope module mux_BPSK_0 $end
16 $var wire 1 ! mux_bpsk_o $end
17 $upscope $end
18 $upscope $end
19 $upscope $end
20 $upscope $end
21 $upscope $end
22 $upscope $end
23 $enddefinitions $end
24 #0
25 x!
26 #13900
27 0!
28 #15331700
29 1!
30 #20335700
31 0!
32 #25339700
33 1!
34 #30343700
```

Figura 4.26: Ejemplo de señales VCD exportadas por el ModelSim.

Este formato no es reconocido por el *LTSpice*, por lo que antes de importarlo es necesario hacerlo compatible. El primer paso consiste en quitar el encabezado y después generar pares de valores de tensión y tiempo en forma de tabla. Luego, se agregan puntos intermedios entre cada cambio de valor de tensión, lo cual permite reconstruir el tren de pulsos y mejorar el funcionamiento del filtro. Finalmente se traslada la señal en el eje *y* para ubicarla entre -1 y 1, evitando así espectros espurios en el proceso de simulación. Todo este procedimiento se realiza en *Python* y se exporta un archivo de texto con los nuevos datos, los cuales son cargados en el *LTSpice*.

En este programa de simulación de circuitos el primer paso efectuado consiste en aplicar un filtrado pasabajos a la señal modulada con el objetivo de eliminar las componentes espectrales de mayor frecuencia. El próximo paso es generar las

señales en fase y cuadratura, para lo cual las señales filtradas son mezcladas con un oscilador que tiene frecuencia igual a la de la señal original, lo que permite que la señal se traslade hacia banda base y al doble de frecuencia. Después, cada una de las señales I/Q se pasa por otro filtro pasabajos con el que se eliminan las frecuencias altas y se obtiene solamente el espectro centrado en frecuencia cero. Las señales I/Q en banda base obtenidas se exportan del *LTSpice* y estos datos son utilizados para obtener a través de un script de Python los diagramas de constelaciones y de vectores.

#### 4.2.2. Procesamiento numérico para obtener los cambios de fase

Una forma alternativa para determinar la fase es procesar numéricamente la serie temporal que obtenemos como salida en el software ModelSim. Suponiendo que se tiene una serie temporal descrita por una ecuación de la forma  $x(t) = A \cos(\omega t + \varphi)$ , la cual consiste en un conjunto de puntos de la forma  $(t_1, x(t_1)); \dots; (t_i, x(t_i)); \dots; (t_N, x(t_N))$ , donde la frecuencia ( $\omega$ ) es conocida y se quiere determinar la fase ( $\varphi$ ).

Se puede definir un procedimiento para encontrar dicha fase, para el cual se toma una función de referencia  $y(t) = \cos(\omega t + \phi)$ , que se usará para comparar con  $x(t)$ . Una posible estrategia es definir la nueva función

$$g(\phi) = \int_0^T x(t)y(t)dt, \quad (4.7)$$

entonces, sustituyendo en la ecuación 4.7 los valores de  $x(t)$  e  $y(t)$  anteriormente definidos e integrando, se tiene que

$$g(\phi) = \frac{A\pi}{\omega} \cos(\phi - \varphi), \quad (4.8)$$

de donde se puede ver que la función  $g(\phi)$  es máxima cuando  $\phi = \varphi$ . Teniendo en cuenta esto último se puede definir un procedimiento numérico para encontrar la fase de una serie temporal de la que se conoce su frecuencia. En este caso se aproxima la integral anterior como una suma

$$g_N(\phi) = \sum_{i=1}^N x_i \cos(\omega t_i + \phi), \quad (4.9)$$

de modo que si evaluamos la función  $g_N(\phi)$  en un conjunto de valores de  $\phi \in [0, 2\pi]$ , el valor de  $\phi$  para el que la función sea máxima, será la aproximación numérica a la fase que se busca.

El procedimiento mostrado anteriormente puede ser aplicado a la señal de datos modulada PSK luego de ser filtrada, ya que ésta es una señal sinusoidal de la forma  $x(t)$ . Pero además, también es aplicable a la señal cuadrada pura que sale de la FPGA

como se demuestra a continuación:

El desarrollo de Fourier de una señal cuadrada es

$$x[t] = A_0 \cos(\omega t + \varphi) + \sum_{n=1}^{\infty} A_n \cos((2n + 1)\omega t + \varphi). \quad (4.10)$$

Siguiendo el procedimiento anterior, se puede definir una función que sea la integral en un período de la multiplicación de  $x[t]$  e  $y(t)$ , siendo el resultado:

$$h(t) = g(\phi) + \int_0^T \sum_{n=1}^{\infty} \frac{A_n}{2} (\cos((2n + 1)\omega t + \varphi - \varphi) + \cos((2n + 1)\omega t + \varphi + \varphi)). \quad (4.11)$$

Las funciones sinusoidales integradas en un número entero de períodos dan como resultado cero, por lo que  $h(t) = g(\phi)$ . A partir de esta igualdad, se puede inferir que el procedimiento numérico descrito anteriormente es también aplicable a una señal cuadrada. De esta manera se puede proceder a usar el mismo procedimiento numérico para calcular el desfasaje.



# Capítulo 5

## Resultados

### 5.1. Verificación por simulación del funcionamiento de los bloques del modulador

Una vez realizada la descripción del hardware del modulador digital PSK se procede a simularlo. Las simulaciones lógicas de la arquitectura se realizaron en el programa *ModelSim*, compatible con el *Libero SoC*. El *ModelSim* al ser un software desarrollado para simulaciones de proyectos en lenguajes de descripción de hardware, brinda las opciones de simulación y visualización necesarias para este trabajo.

Para realizar la simulación se creó un testbench, el cual es también una entidad descrita en VHDL, que posibilita aplicar estímulos al circuito que se quiere simular y observar las respuestas del mismo de forma virtual. Los testbench soportan estructuras que incluso pueden ser no sintetizables en la FPGA, pues su objetivo no es formar parte del diseño final, sino simular las entradas del dispositivo bajo prueba (DUT), en este caso el modulador digital, y verificar si la salida es la esperada.

En este trabajo se realizaron cuatro procesos de simulación, uno para cada uno de los moduladores PSK con los que cuenta el diseño. Para ello se establecieron como estímulos las entradas de los esclavos AHB y APB, los cuales simulan los datos a transmitir y las configuraciones de los moduladores, respectivamente. Si bien los valores absolutos, como son la frecuencia de operación, se encontrarán supeditados a las necesidades particulares de la aplicación final en la que se integre este modulador, a los fines de realizar la evaluación funcional del mismo se estableció como período de reloj  $T_{clk} = 27,8 \text{ ns}$ , que es equivalente a que la frecuencia de reloj sea  $f_{clk} \approx 36 \text{ MHz}$ . Se puede verificar el valor anterior en la Figura 5.1, donde los cursores miden el período de un ciclo de reloj que es efectivamente el configurado.

A partir de la relación expresada en la ecuación 4.6 y conociendo el período del sistema fijado para la simulación, se puede determinar que el período de portadora es  $T_c = 10008 \text{ ns}$ , por lo que la frecuencia de la portadora es  $f_c \approx 99920 \text{ kHz}$ . En el

contexto de los valores de ejemplo seleccionados para este ejercicio, es posible obtener entonces una tasa de datos de aproximadamente 100 *kbps* para modulaciones BPSK, 200 *kbps* para modulaciones QPSK/OQPSK y 300 *kbps* para 8PSK.

El período de símbolo está dado por  $T_s = N_c \cdot T_c$ , donde  $N_c$  es el número de ciclos de portadora que se transmiten antes de cambiar de símbolo. Este valor es el mismo para todas las simulaciones y se establece como  $N = 100$ . Sustituyendo este valor y el período de la portadora en la expresión anterior, se obtiene que  $T_s = 1,000800$  ms.

Por otro lado, como señal de datos se decidió usar una trama de 256 bits, que es un tamaño que garantiza que los tiempos de simulación no sean excesivos y al mismo tiempo es una cantidad de datos suficiente para verificar el correcto funcionamiento del modulador. El Randomizador agrega 32 bits de marca de sincronismo, por lo que a la entrada del Codificador se tienen un total de 288 bits. Como se vio anteriormente, este último bloque genera 2 bits de datos codificados por cada bit de entrada, por lo tanto la cantidad de bits a modular será 576. A partir de este número se puede deducir que se transmitirán 576, 288 y 192 símbolos para BPSK, QPSKs y 8PSK, respectivamente.

El tiempo de simulación mínimo necesario será  $t_{sim} = T_s \cdot N_s$ , donde  $N_s$  es el número de símbolos a transmitir. Teniendo en cuenta lo anterior y dado que los datos pasan por varios bloques antes de modularse, se hace necesario simular un poco más de tiempo para lograr obtener todos los datos modulados. Los tiempos de simulación escogidos para este trabajo son los mostrados en la siguiente tabla:

Modulación	$t_{sim}$
BPSK	578 ms
QPSKs	290 ms
8PSK	194 ms

**Tabla 5.1:** Tiempos configurados para cada una de las simulaciones.

Las secciones siguientes explican, apoyándose en el uso de imágenes de la simulación, el funcionamiento de los bloques del modulador. La señal de datos colocada como estímulo es la misma para todas las simulaciones, lo que cambia es la técnica de modulación escogida. Debido a esto, el comportamiento de los bloques APB, AHB, Randomizador, Codificador y Controlador FIFO será el mismo en todos los casos, por lo que la verificación de estos se muestra una sola vez.

Las imágenes de las simulaciones en el *ModelSim* muestran a la izquierda las señales de cada uno de los bloques que han sido testeados y sus respectivos valores en un instante de tiempo dado. Un etiqueta con el nombre de cada bloque indica las señales que pertenecen a él. Las señales identificadas con un rombo azul y una flecha verde a la izquierda, representan las señales de entrada, mientras que las de salida tendrán la flecha a la derecha del rombo. Las que no tienen flechas representan se-

## 5.1 Verificación por simulación del funcionamiento de los bloques del modulador 51

ñales internas del bloque. En la parte derecha de la ventana se observa la pantalla principal que muestra los resultados de las simulaciones, en la que se ven los valores que toman las señales en cada instante de tiempo. Los cursores en amarillo se utilizan para fijar momentos de tiempo y medir tiempo transcurrido entre intervalos de eventos. Estos valores medidos se observan en la parte inferior de la pantalla principal. La Figura 5.1 es un ejemplo de simulación, en ella se pueden observar todos los detalles descritos anteriormente.

### 5.1.1. Funcionamiento de los esclavos APB y AHB

El conjunto de 256 bits que se muestra a continuación (en formato hexadecimal), fue generado de manera aleatoria en Python para representar la entrada de datos del modulador:

```
7F97A50DDBEBA0CAA358C12DA9A7EE67B2103926275688A47C05C778B361E70A. (5.1)
```

Estos datos son entregados de a un byte por vez al esclavo AHB con las correspondientes señales de control del mismo, simulando así lo que haría un maestro AHB. De la misma forma se simula como entrada del esclavo APB el valor de ciclos de portadora y además, se establece como estímulo la información que determina la técnica de modulación que se quiere aplicar. Ésta última también es entregada al esclavo APB, quien se encarga de derivar estos datos a los bloques de selección de la modulación. La correspondencia entre la señal de datos del esclavo y la modulación a escoger se muestra en la siguiente tabla:

Señal de datos APB	Modulación
0x00000001	BPSK
0x00000002	QPSK
0x00000003	OQPSK
0x00000004	8PSK

**Tabla 5.2:** Correspondencia entre señal de datos APB y modulación seleccionada.

En la Figura 5.1 se puede comprobar que hay sólo un modulador habilitado a la vez. En ella se observa que hay un *enable* en alto, lo que habilita al modulador seleccionado, y el resto están en estado bajo. La señal *sel\_modulation\_i* es quien define la modulación. En este caso tiene en valor *0x00000003* que según la Tabla 5.2 corresponde a OQPSK. Como es de esperar el *enable* correspondiente a esta modulación (*en\_oqpsk\_o*) pasa a valer '1', mientras que el resto permanecen en '0'. Estos valores se mantienen así hasta tanto un nuevo cambio de modulación sea requerido.

Teniendo en cuenta el funcionamiento de ambos esclavos, la simulación de la

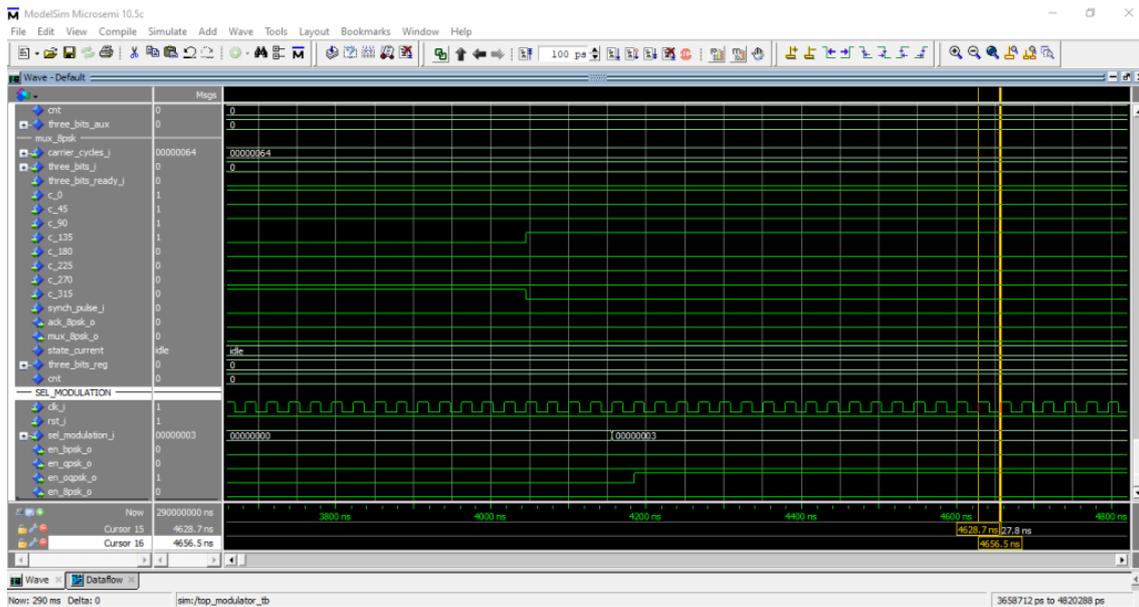


Figura 5.1: Valores de los *enables* durante la simulación.

entrega de los datos se realiza una vez que se compruebe que las señales de *ready* de los dos están en '1'. La Figura 5.2 muestra la simulación del bloque esclavo APB, en la que se puede observar cómo la señal de direcciones del mismo (*paddr*) toma primero el valor  $0x00000002$  que es quien establece el valor de  $N_c$ . Se puede comprobar mirando la señal *carrier\_cycles* que efectivamente el valor se establece en  $0x00000064$  que representa en notación hexadecimal al número 100. Seguidamente la dirección cambia a  $0x00000001$  que determina la modulación a utilizar. En el caso de la imagen, el valor del dato es  $0x00000002$  que según la Tabla 5.2 corresponde a la modulación QPSK.

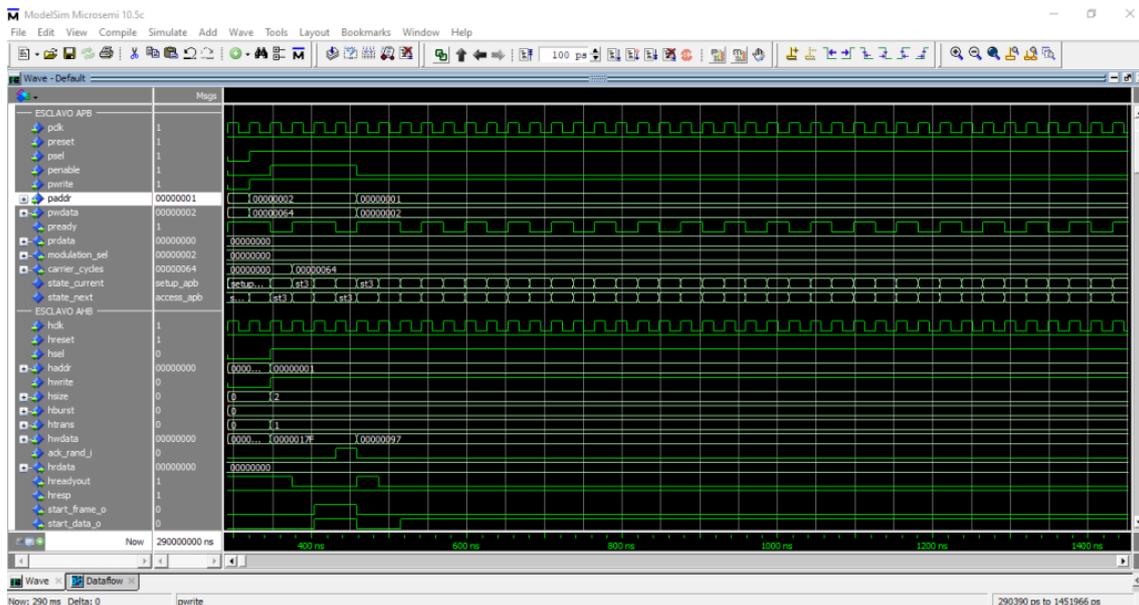


Figura 5.2: Funcionamiento del esclavo APB.

## 5.1 Verificación por simulación del funcionamiento de los bloques del modulador 53

El comportamiento del esclavo AHB es mostrado en la Figura 5.3. En la misma se puede observar que los últimos 8 bits de la señal de datos (*hwdata*), lo cuales representan la información a transmitir, son entregados al randomizador a través de la señal de salida del esclavo *data\_o*. El bit número 8 de esta señal indica el inicio de una nueva trama y este valor se transmite al Randomizador a través de la señal *start\_frame\_o*. Es por esto que esta señal se pone en alto sólo durante el tiempo que dure el primer valor de la trama de datos en el bus de comunicación (*0x7F*). La señal de inicio de data (*start\_data\_o*) se pone en alto cada vez que se genera un acceso al esclavo APB, al tiempo que se coloca un nuevo valor en *data\_o*.

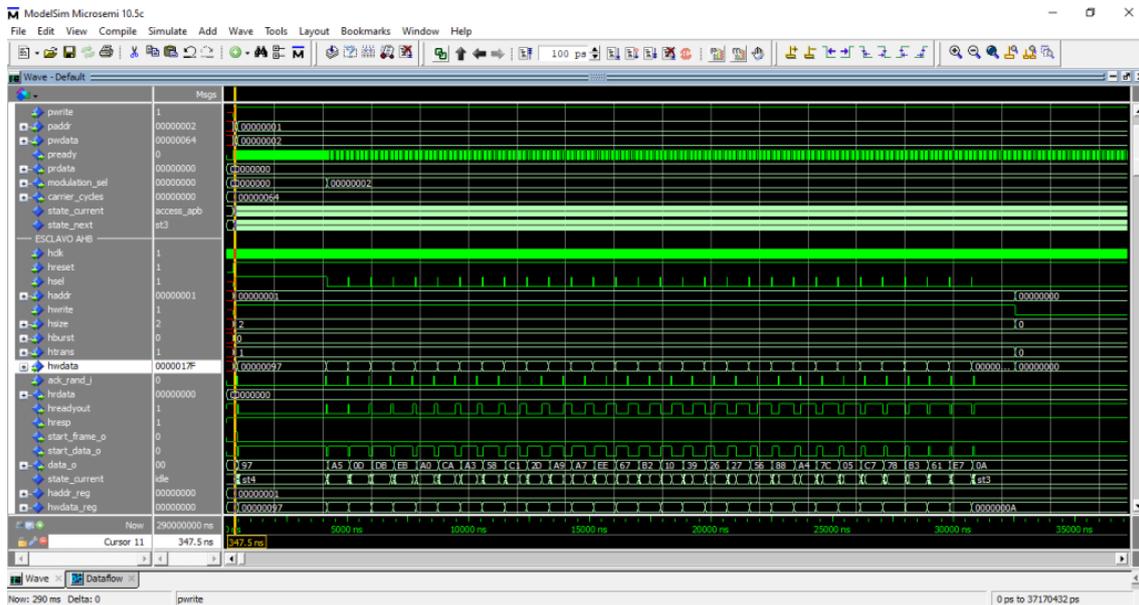


Figura 5.3: Funcionamiento del esclavo AHB.

Con el objetivo de comprobar que la FIFO es leída correctamente y que se vacía cuando son leídos todos los datos que se escribieron en ella, una vez que son entregados los 256 bits de una trama de datos no se generan más accesos al esclavo AHB. Esto se puede verificar con la señal *hsel* que cambia a '0' luego de entregar el último byte de la trama (*0x0A*).

### 5.1.2. Funcionamiento del Randomizador

Las señales de simulación correspondientes al Randomizador se muestran en la Figura 5.4. Se puede verificar que la señal de salida del esclavo AHB (*data\_o*) es la entrada del Randomizador (*data\_i*). Además, se puede observar a través de la señal *data\_rand\_o*, que representa la señal de datos de salida del bloque, que se colocan primero los cuatro bytes de la marca de sincronismo, uno por vez y luego se comienzan a entregar los datos randomizados. La imagen está en una escala tal que se pueda ver que todos los valores de la trama que entran al randomizador son

entregados, lo que hace que el primer valor de la misma no se vea a simple vista. Es por esto que se colocó un cursor encima de este dato, así el valor de la señal en ese momento se muestra en la parte izquierda de la pantalla, comprobándose que es  $0x1A$  como se esperaba. También se pueden observar los valores que toma la señal que avisa al Codificador que los datos randomizados están listos (*data\_ready\_o*). Esta señal se mantiene en '1' todo el tiempo que el dato está disponible, y baja cuando el Randomizador recibe la señal de *ack* del Codificador.

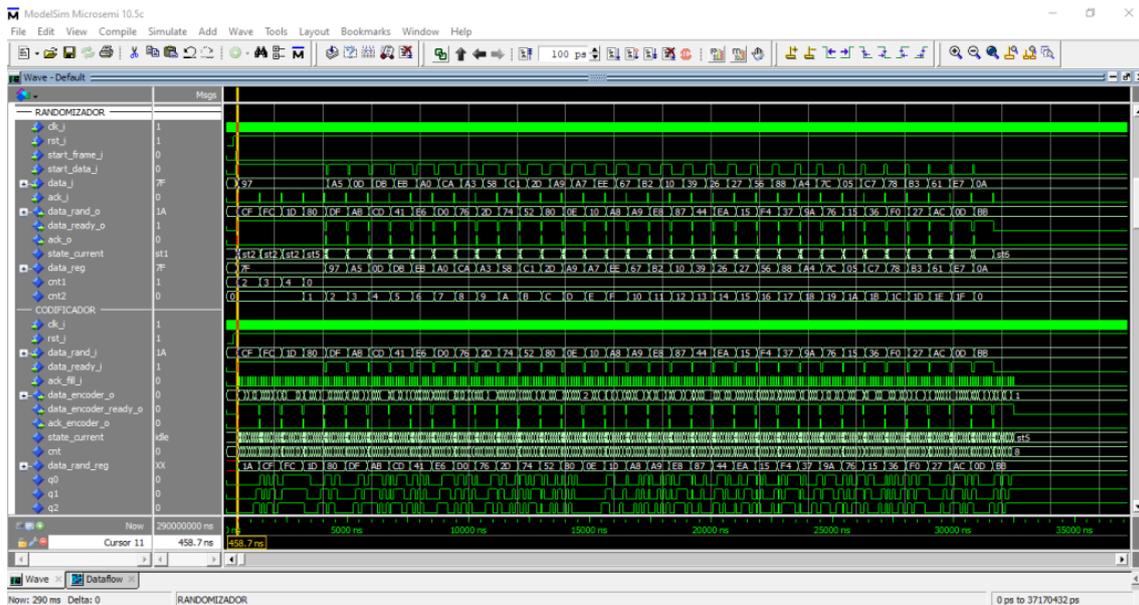


Figura 5.4: Funcionamiento del randomizador y agregado de la ASM.

Para comprobar que los datos están efectivamente randomizados de la manera sugerida por el estándar [7], se puede realizar la operación XOR entre la señal de datos de estímulo mostrada en 5.1 y la secuencia pseudoaleatoria mostrada en 4.2. El resultado de este cálculo, expresado en notación hexadecimal, es el siguiente:

$$80DFABCD41E6D0762D7452800E10A8A9E88744EA15F4379A761536F027AC0DBB, \quad (5.2)$$

que corresponde efectivamente a los valores que va tomando *data\_rand\_o* luego de sacar la marca de sincronismo.

### 5.1.3. Funcionamiento del Codificador

El bloque Codificador recibe la señal randomizada, la pasa a través del codificador convolucional que se muestra en la Figura 4.7 y coloca a su salida dos bits codificados por cada bit de datos de entrada. Como en el caso anterior, cada vez que se tenga un nuevo dato codificado se activará una señal que avisa que el dato está disponible (*data\_encoder\_ready\_o*).

## 5.1 Verificación por simulación del funcionamiento de los bloques del modulador 55

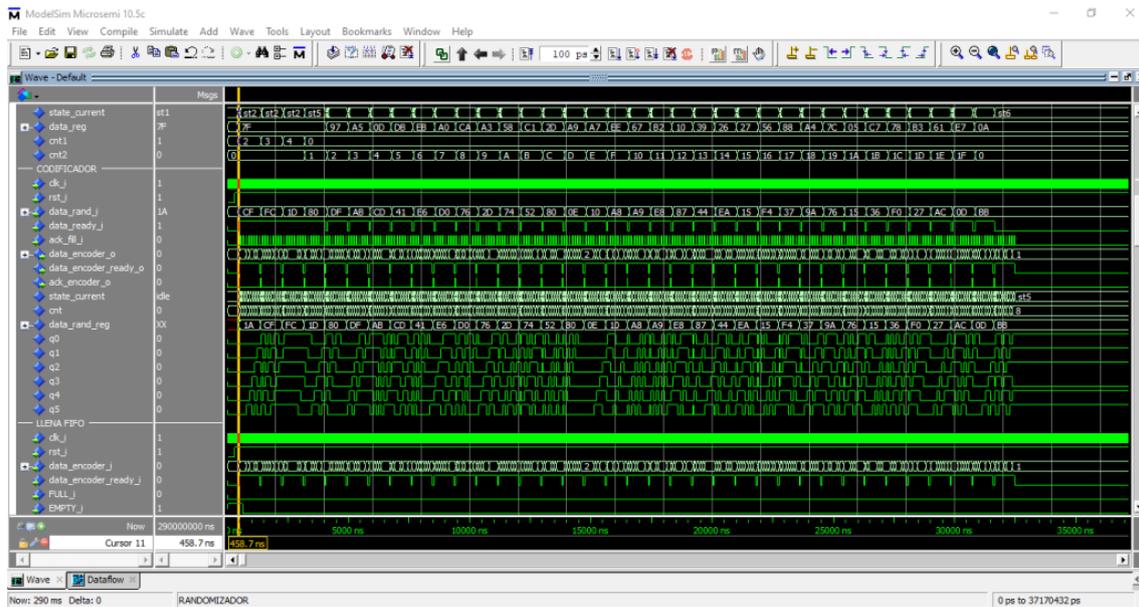


Figura 5.5: Funcionamiento del codificador.

Observando en la Figura 5.5 la señal *data\_encoder\_o*, la cual representa la salida del bloque, se puede ver que los valores codificados son los esperados. Esto se puede comprobar con la Tabla de verdad 5.3 que muestra los estados del registro de desplazamiento del codificador convolucional, así como las salidas del mismo, cada vez que se coloca un dato nuevo en su entrada. Los cálculos se realizan partiendo de que en el estado inicial todos los flip-flops del registro están en '0' y utilizando como entradas los tres primeros bytes que llegan al codificador ( $0x1AC$ ), tal como en la simulación. Tal cálculo se puede extender al resto de los bytes mostrados en expresión 5.1 obteniendo los resultados que se muestran en la simulación, aunque para no hacer muy largo este informe se decidió incluir solo esta primera parte.

q5	q4	q3	q2	q1	q0	C1	C0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	1	1	0	0	0	1
0	1	0	1	1	0	0	0
1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	0
0	1	1	0	1	0	1	1
0	0	1	1	0	1	0	0

Tabla 5.3: Tabla de verdad del codificador convolucional.

### 5.1.4. Funcionamiento de la FIFO

El correcto funcionamiento de la estructura de almacenamiento se puede comprobar en la Figura 5.6. La señal *bit\_fifo\_o* representa el bit que será colocado en la FIFO cuando la señal de escritura *WE\_o* esté activa. Si se analiza de manera secuencial los valores *bit\_fifo\_o* cuando *WE\_o* = 1, se puede ver que los valores con los que se llena esta estructura son 101010010000..., los mismos que salen del codificador. Se puede observar que la señal que indica si la FIFO está vacía (*EMPTY\_i*) está en '1' al inicio de la simulación, verificándose así que no contiene ningún dato. Luego que *WE\_o* se active por primera vez, es decir se almacenan datos, *EMPTY\_i* pasa a valer '0'.

La verificación de la lectura se verá en la sección 5.1.5, donde los bloques de selección de bits extraen de la FIFO los datos que serán modulados. En esa sección también se pueden ver figuras con el proceso de simulación completo en las que se verificará que una vez sean leídos todos los datos de la FIFO, *EMPTY\_i* volverá a tomar valor '1'. Además, ahí se puede comprobar que la estructura de almacenamiento nunca se llena, como es de esperar, dado que fue configurada con una capacidad de 1024 bits y sólo se almacenan en ella 576 bits.

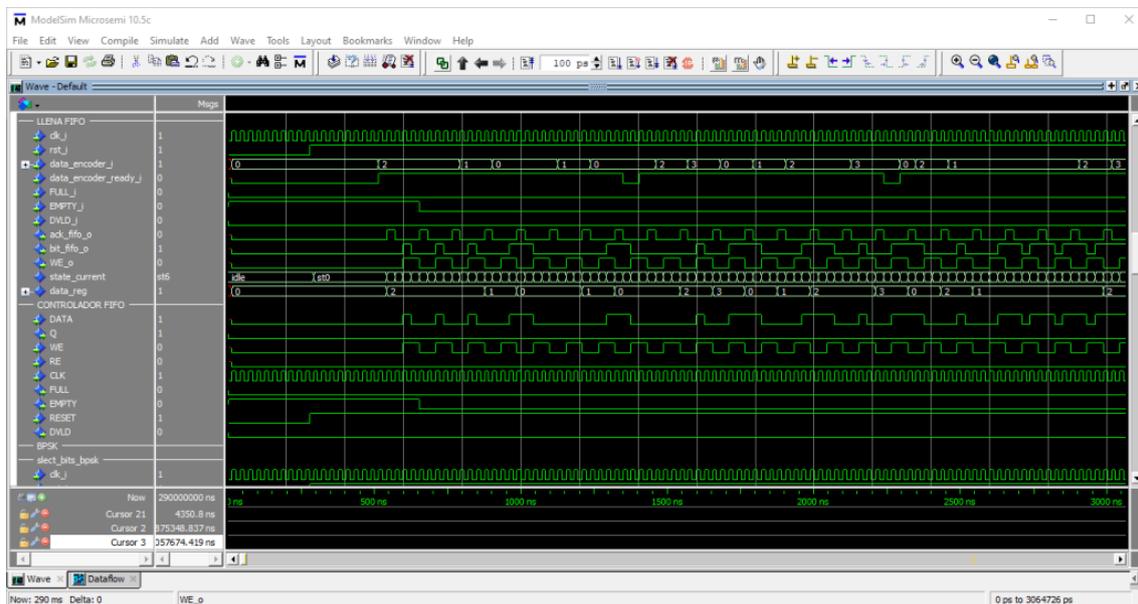


Figura 5.6: Funcionamiento de la FIFO.

### 5.1.5. Funcionamiento de los moduladores

El correcto funcionamiento de cada una de las partes que componen el bloque *Moduladores* mostrado en la Figura 4.11 es descrito a continuación. Para hacer más amena la lectura de esta sección se toma como muestra para la explicación el mo-

## 5.1 Verificación por simulación del funcionamiento de los bloques del modulador 57

dulador OQPSK. Las imágenes de los otros tres moduladores son recogidas en el Apéndice 6.1.

### 5.1.5.1. Simulaciones generales

La Figura 5.7 muestra la simulación completa cuando el modulador se estimula con la trama de datos mostrada en la expresión 5.1 para el modulador OQPSK. Dado que el tiempo de simulación es muy grande comparado con el período de reloj del sistema, la escala no permite que en esta imagen se vean todos los detalles, pero es importante como paso inicial para saber que hay información en todos los puertos.

Además, sirve para cerciorarse de que todos los bits son leídos de la FIFO. Esto lo podemos ver con la señal *EMPTY\_i* que está en '1' al inicio de la simulación, lo cual significa que aún no se le han cargado datos. Luego permanece en '0' durante casi todo el tiempo de simulación para finalmente regresar a '1' cuando todos los datos son leídos. También se puede comprobar a través de la señal *carrier\_cycles\_i* que los ciclos de portadora están establecidos en 100 ( $0x00000064$ ) y, que la señal de enable (*en\_oqpsk\_dmux\_i*), quién selecciona que modulación PSK ha sido configurada, está en '1'.

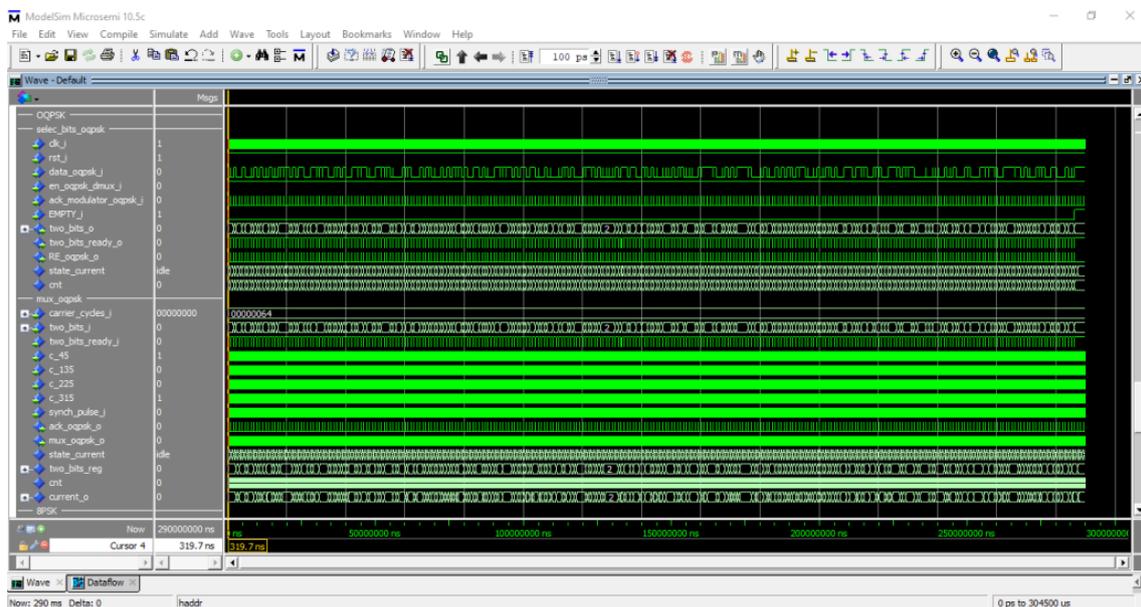


Figura 5.7: Vista general de la simulación OQPSK.

### 5.1.5.2. Verificación de las señales portadoras

Un zoom para una mejor visualización de las señales portadoras se muestra en la Figura 5.8. En ellas se puede observar la medición mediante el uso de cursores del período de las portadoras y los desfases entre ellas. Se ubica un cursor en un flanco ascendente que marca el inicio de un período para cada una de las señales



## 5.1 Verificación por simulación del funcionamiento de los bloques del modulador 59

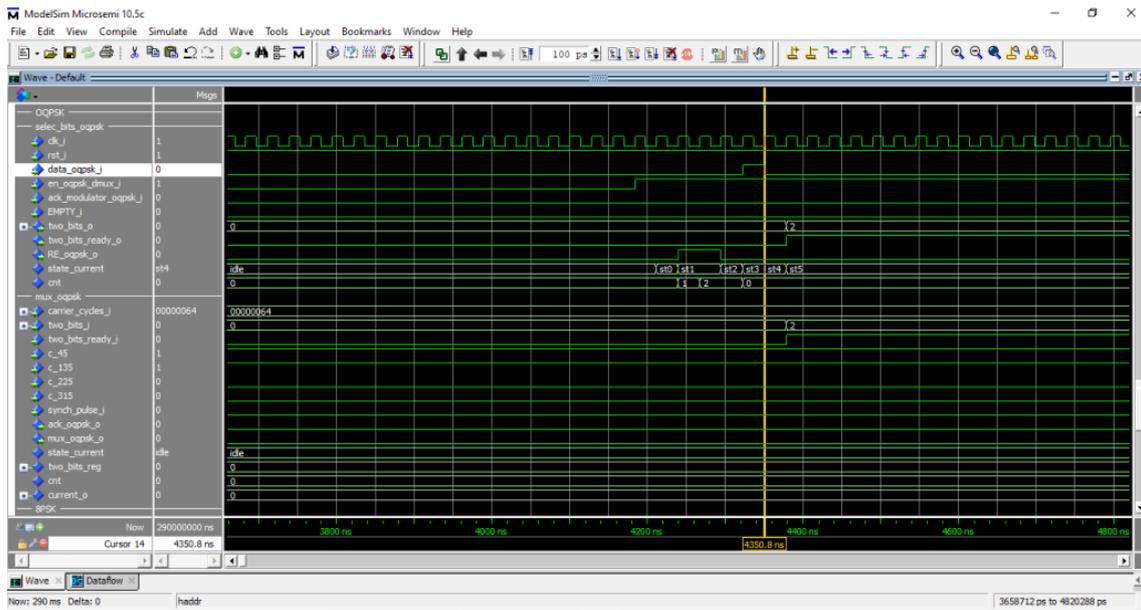


Figura 5.9: Ejemplo de selección de bits en la modulación OQPSK.

Como se ha explicado con anterioridad, OQPSK tiene estados de transición para que no ocurran cruces por cero, en los que la señal modulada toma valores correspondientes a símbolos consecutivos durante medio período. Es por ello que en las imágenes se observa que el sistema se mantiene en el *st2* durante un tiempo igual a 50 ciclos de portadora, en el que el modulador está colocando a su salida el valor de una portadora correspondiente a un símbolo de transición. Luego de transcurrido este tiempo, se pasa al estado *st3* en el que se permanece durante la mitad restante del período de símbolo. En este estado, la señal de salida cambia a la portadora equivalente al símbolo que realmente se quiere transmitir.

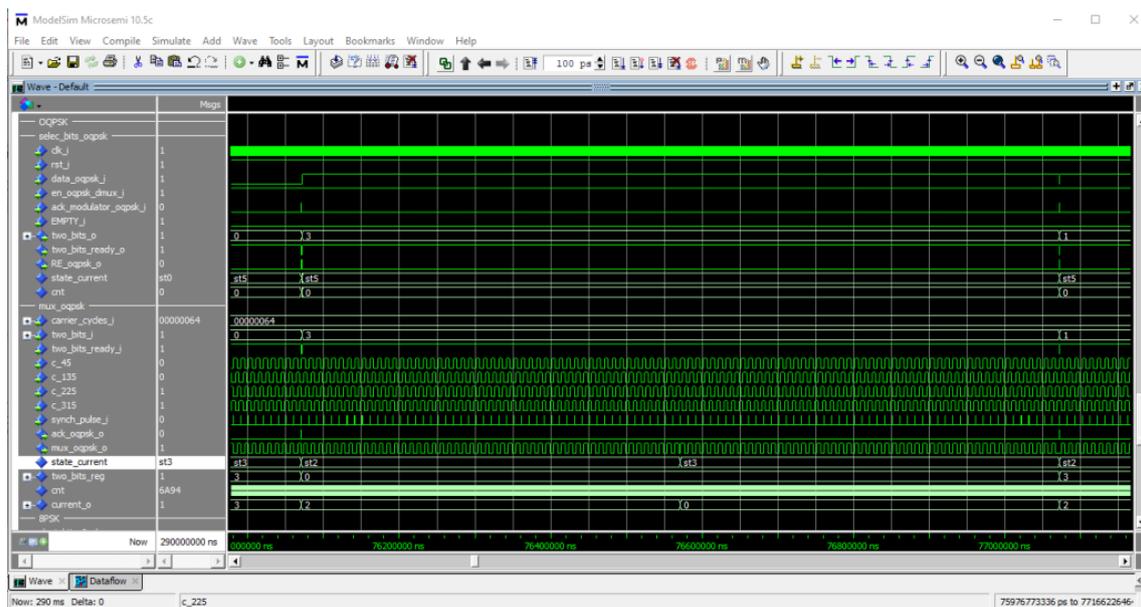


Figura 5.10: Visualización de cambios de fase en modulación OQPSK.

### 5.1.5.5. Verificación del período de símbolo

Debido a que los cursores a esta escala pueden interferir en la visualización de los flancos de los pulsos, se agrega la Figura 5.11 para mostrar el período de símbolo. La imagen es esencialmente la misma que la anterior, solo que se incluyen los cursores. El cursor intermedio mide el tiempo entre los cambios de fase reales ocurridos y el intermedio que se usa para la transición. Se puede verificar que durante la primera mitad del período de símbolo la señal modulada responde al símbolo intermedio y durante la última mitad la señal portadora es la que define el símbolo que se desea transmitir. Se puede observar, además, que el período de símbolo es 1000800 ns tal como se calculó anteriormente.

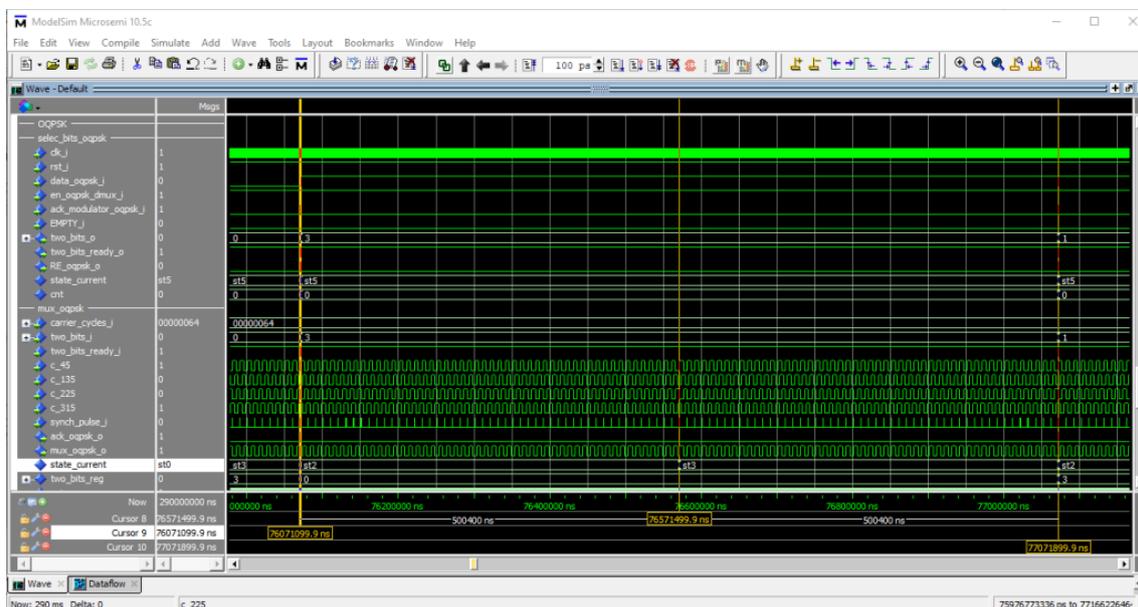


Figura 5.11: Medición del período de símbolo en modulación OQPSK.

### 5.1.5.6. Proceso de elección de portadora

En las Figuras 5.12 y 5.13 se muestra un zoom de la simulación en un momento de cambio de fase. Con estas imágenes se puede comprobar que las señales moduladas son las que se esperan a partir de los bits leídos de la FIFO, verificándose así el correcto funcionamiento del modulador OQPSK. Para analizar el proceso de modulación se analizan dos imágenes, así la escala permite observar mejor el paso por el estado intermedio. Ambas imágenes pertenecen al mismo proceso de cambio de fase, viéndose en la Figura 5.12 una ventana de tiempo inmediata anterior a la mostrada en la Figura 5.13. La señal *current\_o* muestra los bits actuales que definen la portadora a la salida del modulador. En la imagen de arriba se puede observar que el valor de *current\_o* es "11", por lo que el valor de *mux\_oqpsk\_o* es igual a *c\_225*. Durante ese tiempo la señal *two\_bits\_i*, que contiene el valor leído de la FIFO, vale

## 5.1 Verificación por simulación del funcionamiento de los bloques del modulador 61

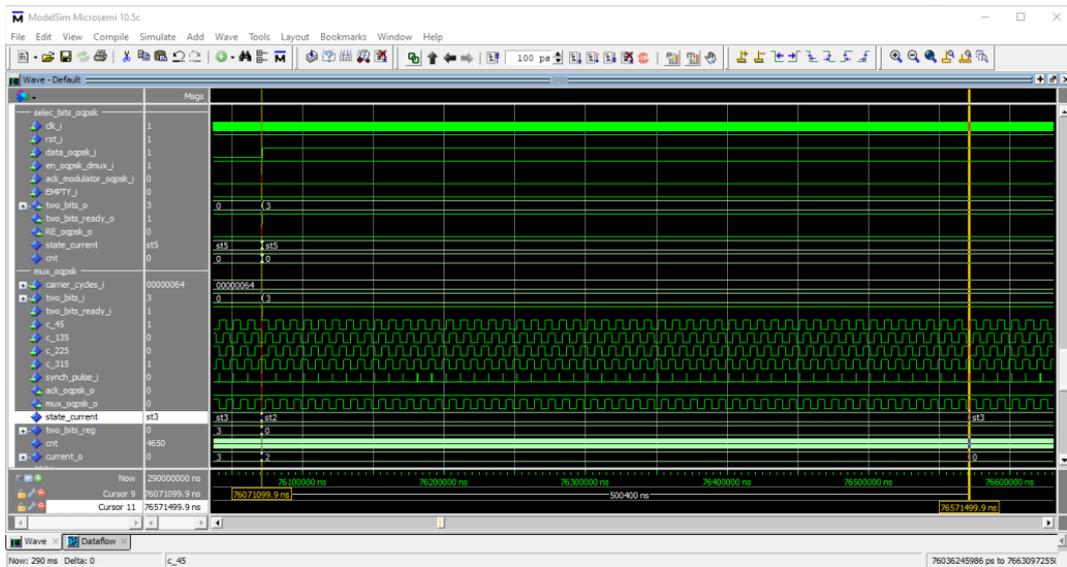


Figura 5.12: Proceso de selección de la señal portadora OQPSK (a).

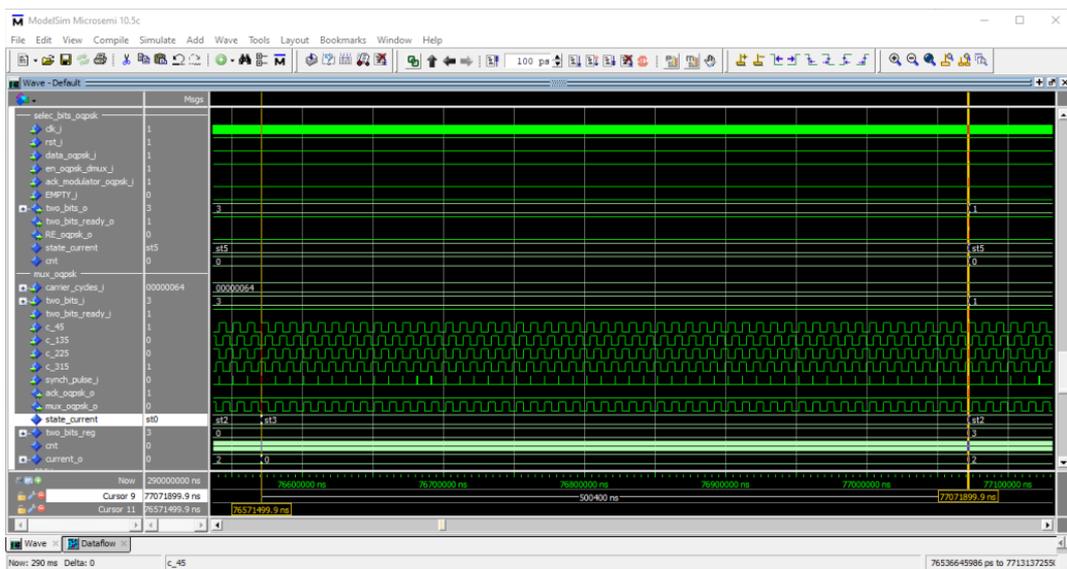


Figura 5.13: Proceso de selección de la señal portadora OQPSK (b).

"00", por lo que esos serán los datos a modular en el próximo período de símbolo. Debido a que los símbolos "11" y "00" están opuestos en el diagrama de constelaciones, no se puede cambiar de manera directa a la portadora equivalente a "00", por lo que se pasa por un valor de transición. Se puede comprobar en la imagen que durante todo el tiempo que dura el estado `st2` la señal modulada tiene el mismo valor que la portadora `c_135`, la cual corresponde a "10", símbolo adyacente a "00". Luego de transcurrir los 500400 ns, se pasa al estado `st3` donde la señal modulada será igual a la portadora `c_0`, equivalente a los bits "00" que se necesitaban modular. En la Figura 5.13 se comprueba cómo el sistema permanece en el estado `st3` hasta terminar el período de símbolo completo. En ese momento se pasarán a modular los nuevos datos leídos ("11"), que como vuelven a representar un símbolo opuesto

al recién modulado, será necesario también colocar a la salida del modulador una portadora equivalente a un símbolo adyacente durante medio período de símbolo y luego cambiar al valor real correspondiente.

## 5.2. Utilización de recursos

Una de las características de gran relevancia en todo diseño basado en lógica programable es la cantidad de recursos de FPGA requeridos para implementar una funcionalidad dada. En el caso particular implementado en este trabajo, dado el contexto de aplicación y el hecho de que este se concibe como un bloque a ser integrado a un diseño mayor, la ocupación resultante es un parámetro de particular interés. Esta característica puede ser interpretada como un indicativo del impacto de agregar un bloque particular a un sistema mayor.

Para realizar la evaluación, se utilizarán los reportes de ocupación generados por el proceso de síntesis, y se compararán con los recursos disponibles en algunas FPGAs típicamente usadas para este tipo de aplicaciones. Es importante remarcar que estas comparaciones no son de carácter exacto, sino más bien estimativo, producto de las diferencias existentes entre las tecnologías de celda base que implementan las FPGAs de distintas familias y fabricantes.

La FPGA *Radiation-Tolerant ProASIC3 RT3PE3000L* de *Microchip* fue una de las escogidas para comparar con el diseño desarrollado en este trabajo. Esta selección se debe a que este tipo de FPGAs, como su nombre lo sugiere, son tolerantes a la radiación, lo que las hace preferidas para aplicaciones satelitales. Las celdas base de las *ProASIC3*, denominadas *VersaTiles* por el fabricante, pueden consistir en una LUT-3 (LookUp Table de tres entradas), un latch con clear o set, un Flip-Flop (FF) D con clear o set, o un FF D con clear o set y enable [8]. Por lo general, en este tipo de implementaciones una LUT-3 puede resolver en promedio una función equivalente a tres compuertas lógicas. La Figura 5.14 muestra los recursos de la FPGA en cuestión especificados por el fabricante en la hoja de datos de la misma.

Otras FPGAs utilizadas para analizar el uso de recursos del modulador digital diseñado en este trabajo son las de la familia *Zynq UltraScale+ MPSoC* de *Xilinx*, las cuales también son preferidas este tipo de aplicaciones debido al rendimiento y nivel de integración que proveen. Las FPGAs provistas por este fabricante tienen como celdas básicas las CLB (Configurable Logic Block) que están formadas al mismo tiempo por ocho LUT-6 y dieciséis FF. Las LUT-6 pueden configurarse como una LUT de 6 entradas con una salida, o como dos LUTs de 5 entradas con salidas separadas pero con entradas comunes [9]. La Figura 5.15 muestra los recursos de algunas FPGAs de esta familia.

RT ProASIC3 Devices	RT3PE3000L
System Gates	3 M
VersaTiles (D-flip-flops)	75,264
RAM kbits (1,024 bits)	504
4,608-Bit Blocks	112
FlashROM Bits	1 k
Secure (AES) ISP	Yes
Integrated PLL in CCCs	6
VersaNet Globals	18
I/O Banks	8
Maximum User I/Os	620
Package Pins CCGA/LGA	CG/LG484, CG/LG896

**Figura 5.14:** Recursos disponibles en las FPGAs RT ProASIC3 de acuerdo a la hoja de datos [8].

Conociendo los recursos que ofrecen las FPGAs recién mencionadas, se procede a analizar el resultado del proceso de síntesis del diseño, que en este trabajo fue realizado con la herramienta *Synplify*. El resultado de la síntesis es mostrado en la Figura 5.16, donde se puede observar que el diseño utiliza 5379 Core Cells, las cuales son la unidad base de recursos provista por este software.

El desglose de estos resultados ofrecidos por el *Synplify* establece que de el total de Core Cells, 4606 son LUTs y el resto son FFs. Dado que las VersaTiles pueden representar LUTs o FFs, esta división no es relevante, por lo que la comparación se realiza uno a uno con los Core Cell. Para el caso de las FPGAs de Xilinx, el fabricante ofrece en la hoja de datos la capacidad en bloques CLB LUT y CLB FFs, entonces si es conveniente utilizar los resultados más detallados. En promedio, y en forma conservadora, se puede asumir que una LUT-6 tiene una capacidad equivalente a 6 compuertas lógicas. De esta forma, la cantidad de recursos de una FPGA con estas características equivale al menos al doble de una con LUT-3. La Tabla 5.4 muestra los porcentajes de utilización de recursos de dos FPGAs por el modulador desarrollado en este trabajo.

	Síntesis	% de uso de <i>ProASIC3 RT3PE3000L</i>	% de uso de <i>UltraScale+ ZU1CG</i>
CLB LUTs	4538	-	$\frac{4538}{37,440 \times 2} = 6,1 \%$
CLB FFs	841	-	$\frac{841}{74,880} = 1,1 \%$
VersaTiles	5379	$\frac{5379}{75264} = 7,1 \%$	-

**Tabla 5.4:** Porcentaje de ocupación de los recursos de las FPGAs.

La ocupación de memoria RAM ofrecida por el *Synplify* muestra que se utiliza un sólo bloque de memoria, el cual equivale a 4608 bits. La *ProASIC3 RT3PE3000L* ofrece 504 kbits, por lo que se ocuparía menos de un 1 % de ésta. En el caso de la *UltraScale+ ZU1CG* se utiliza un porcentaje aún menor, ya que la FPGA tiene 1 Mbit de RAM distribuida y 3,8 Mbit de bloques de RAM, para un total de 4,8 Mbit (4915

	ZU1CG	ZU2CG	ZU3CG	ZU4CG	ZU5CG	ZU6CG	ZU7CG	ZU9CG
Application Processing Unit	Dual-core Arm Cortex-A53 MPCore with CoreSight; NEON & Single/Double Precision Floating Point; 32KB/32KB L1 Cache, 1MB L2 Cache							
Real-Time Processing Unit	Dual-core Arm Cortex-R5F with CoreSight; Single/Double Precision Floating Point; 32KB/32KB L1 Cache, and TCM							
Embedded and External Memory	256KB On-Chip Memory w/ECC; External DDR4; DDR3; DDR3L; LPDDR4; LPDDR3; External Quad-SPI; NAND; eMMC							
General Connectivity	214 PS I/O; UART; CAN; USB 2.0; I2C; SPI; 32b GPIO; Real Time Clock; WatchDog Timers; Triple Timer Counters							
High-Speed Connectivity	4 PS-GTR; PCIe Gen1/2; Serial ATA 3.1; DisplayPort 1.2a; USB 3.0; SGMII							
System Logic Cells	81,900	103,320	154,350	192,150	256,200	469,446	504,000	599,550
CLB Flip-Flops	74,880	94,464	141,120	175,680	234,240	429,208	460,800	548,160
CLB LUTs	37,440	47,232	70,560	87,840	117,120	214,604	230,400	274,080
Distributed RAM (Mb)	1.0	1.2	1.8	2.6	3.5	6.9	6.2	8.8
Block RAM Blocks	108	150	216	128	144	714	312	912
Block RAM (Mb)	3.8	5.3	7.6	4.5	5.1	25.1	11.0	32.1
UltraRAM Blocks	0	0	0	48	64	0	96	0
UltraRAM (Mb)	0	0	0	13.5	18.0	0	27.0	0
DSP Slices	216	240	360	728	1,248	1,973	1,728	2,520
CMTs	3	3	3	4	4	4	8	4
Max. HP I/O <sup>(1)</sup>	156	156	156	156	156	208	416	208
Max. HD I/O <sup>(2)</sup>	24	96	96	96	96	120	48	120
System Monitor	1	2	2	2	2	2	2	2
GTH Transceiver 16.3Gb/s <sup>(3)</sup>	0	0	0	16	16	24	24	24
GTY Transceivers 32.75Gb/s	0	0	0	0	0	0	0	0
Transceiver Fractional PLLs	0	0	0	8	8	12	12	12
PCIe Gen3 x16	0	0	0	2	2	0	2	0
150G Interlaken	0	0	0	0	0	0	0	0
100G Ethernet w/ RS-FEC	0	0	0	0	0	0	0	0

Figura 5.15: Recursos de las FPGAs Zynq UltraScale+ de acuerdo a la hoja de datos [9].

Area Summary			
Core Cells	5379	IO Cells	188
Block RAMs (v_ram)	1		
<a href="#">Detailed report</a>			

Figura 5.16: Resultado de la síntesis desarrollada en el Synplify.

kbits)

Si bien la cantidad de bloques de entrada y salida (I/O) utilizados en el diseño es menor a la capacidad provista por las FPGAs, el análisis del uso de los mismos no es relevante a este nivel. Esto se debe a que el modulador es considerado una parte de un diseño mayor al cual este será integrado por el usuario final, proceso en el cual muchas de estas interfaces se absorben internamente.

### 5.3. Obtención de los diagramas de constelaciones

A partir de las imágenes mostradas en la sección 5.1.5 que verifican el funcionamiento del modulador PSK desarrollado en este trabajo, se puede ver que las señales de salida de los moduladores cambian de fase según los bits de datos a modular, pero la manera tradicional de visualizar estos cambios es a través de un diagrama de

constelaciones tal como se explica en la Sección 4.2.1. Las secciones siguientes muestran el proceso de obtención de las constelaciones por el método de simulación y a través de realizar procesamientos numéricos a la señal.

### 5.3.1. Proceso de simulación en el LTSpice

Las simulaciones en este programa se realizaron para los cuatro moduladores digitales desarrollados en este trabajo. Para mayor comodidad en la lectura de este documento, en esta sección sólo se incluyen imágenes que muestran todo el proceso para la modulación QPSK, mientras el resto de las simulaciones son recogidas en el Apéndice 6.2.

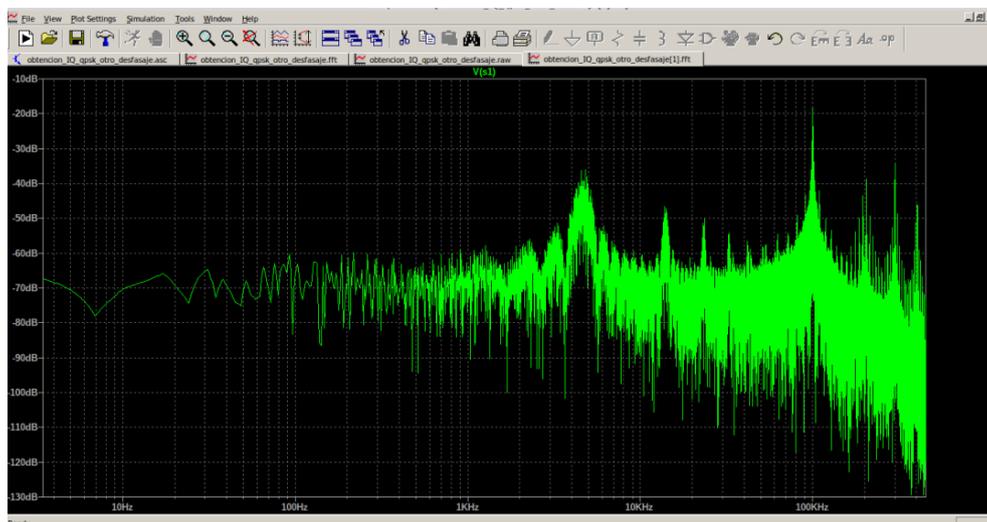


Figura 5.17: Espectro de la señal modulada QPSK.

El primer paso es importar las señales de datos en dicho programa y graficarlas. Los tiempos de simulación establecidos son los mismos que en el programa *Model-Sim*, mostrados en la Tabla 5.1. A partir de esta información y utilizando el mismo programa, se visualiza el espectro de la señal, el cual se muestra en las Figuras 5.17.

Tal como se puede observar en la imagen anterior el espectro está distribuido alrededor de la frecuencia 99920 kHz y los múltiplos impares de ella, teniendo la mayor cantidad de densidad espectral alrededor del armónico principal. Por lo tanto, es necesario colocar un filtro que sea capaz de quedarse con esa frecuencia principal y atenuar lo suficiente los armónicos. Otra característica importante que debe tener el filtro es presentar retardo de grupo constante. Si esto no se cumple pueden aparecer retardos diferentes ante cada componente espectral de la señal, lo cual puede causar distorsión en los símbolos de la señal modulada. El filtro escogido, que se muestra en la Figura 5.18, fue diseñado utilizando la herramienta *LC Filter Design Tool* [10]

de Marki<sup>1</sup>. Es un filtro de Bessel de orden 5 con frecuencia de corte en 110 kHz tal como se puede observar en la Figura 5.19 (a) que muestra la magnitud de la atenuación del mismo. Se puede comprobar en la Figura 5.19(b) que el retardo de grupo es constante alrededor de las frecuencias de interés.

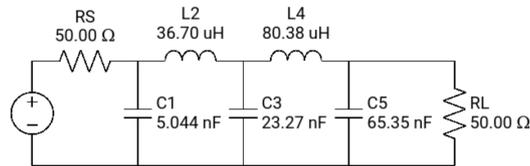


Figura 5.18: Filtro pasabajos Bessel, orden 5,  $f_c = 110$  kHz [10].

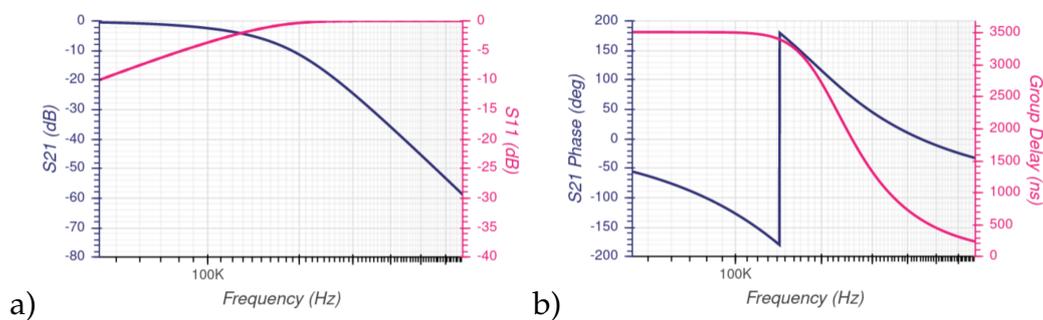


Figura 5.19: Características del filtro pasabajos Bessel, orden 5,  $f_c = 110$  kHz.

En la Figura 5.20 se muestra el espectro de la señal modulada QPSK luego de pasar por el filtro que se ha descrito. Como se puede ver prevalece la frecuencia principal y el espectro en aproximadamente 300 kHz está atenuado lo suficiente ( $\approx 40$  dB).

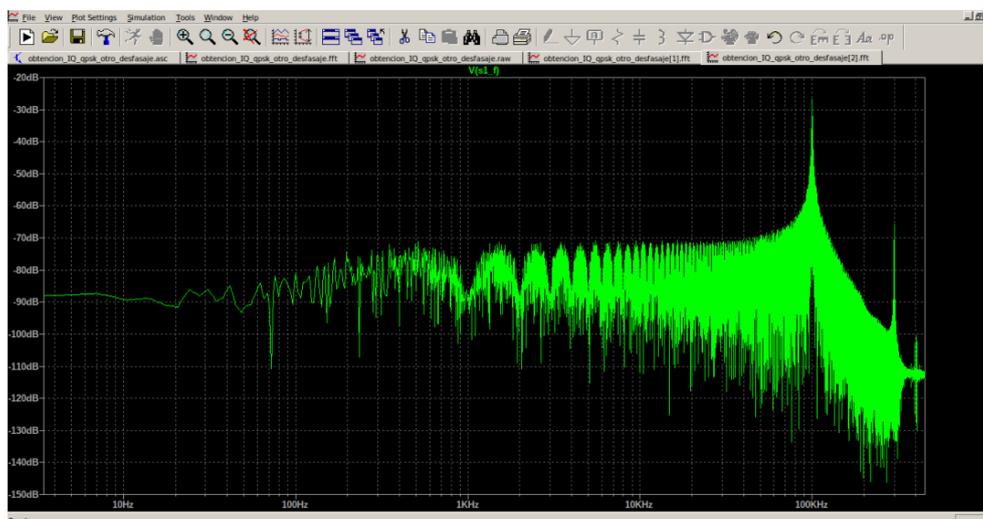


Figura 5.20: Espectro de las señales QPSK filtradas.

<sup>1</sup>Empresa de tecnología fundada en 1991 inmersa en el campo de las radiofrecuencias y microondas [34]

La señal filtrada es mezclada con dos señales sinusoidales desfasadas  $90^\circ$  entre sí, tal como se explicó anteriormente, para generar las señales I/Q. Los espectros de ambas señales resultantes para cada modulador son mostradas en las Figuras 5.21 y 5.22. A partir de ella se puede ver que es necesario un filtro capaz de preservar el espectro de banda base y rechazar lo que se encuentre alrededor del doble de la frecuencia, en este caso  $\approx 200 \text{ kHz}$ .

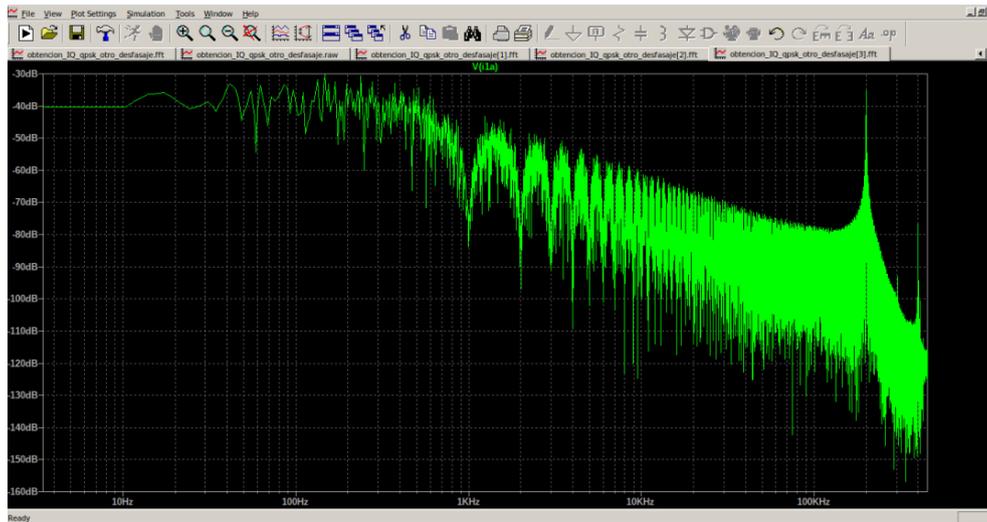


Figura 5.21: Espectro de las señales I QPSK.

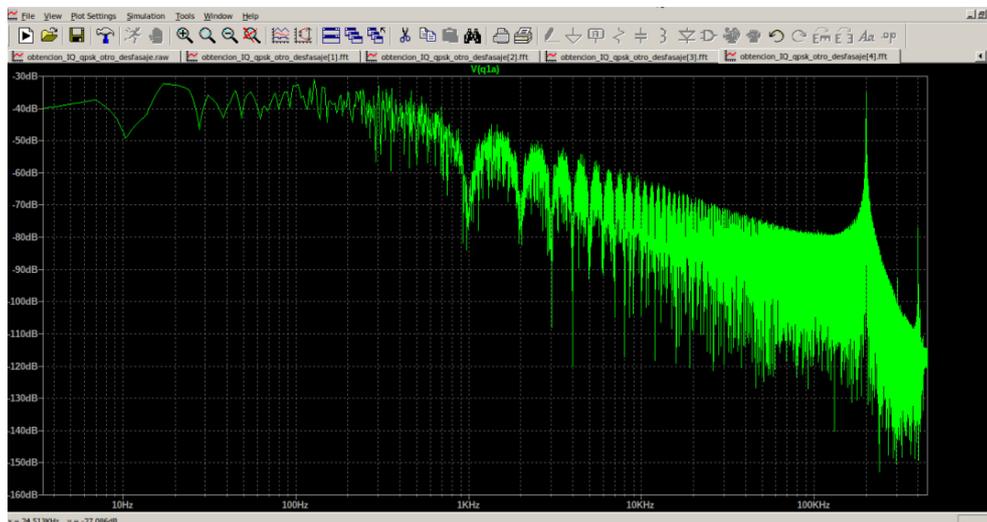


Figura 5.22: Espectro de las señales Q QPSK.

El filtro utilizado para las señales I/Q es también un filtro de Bessel de orden 6, esta vez con frecuencia de corte en 20 kHz y es mostrado en la Figura 5.23. La atenuación del mismo y su retardo de grupo, también constante en la zona de interés, se muestran en la Figura 5.24 (a) y (b), respectivamente.

Los espectros de las señales QPSK en fase y cuadratura, luego de haber sido filtradas son mostrados en las Figuras 5.25 y 5.26. Se puede verificar que las altas

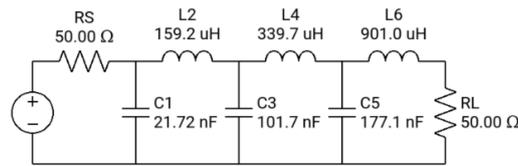


Figura 5.23: Filtro pasabajos Bessel, orden 6,  $f_c = 20$  kHz[10].

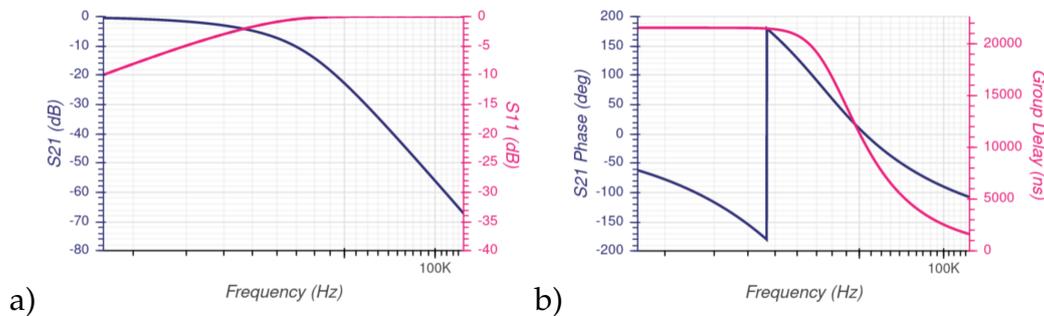


Figura 5.24: Características del filtro pasabajos Bessel, orden 6,  $f_c = 20$  kHz [10].

frecuencias han sido atenuadas y sólo hay densidad espectral en banda base.

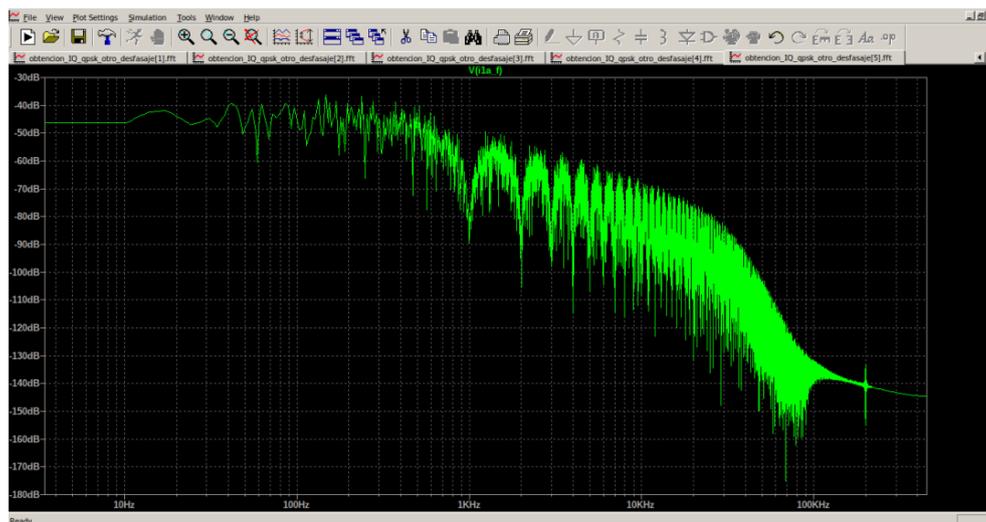


Figura 5.25: Espectro de la señal I QPSK filtrada.

En la Figura 5.27 se muestran las señales de la simulación completa de QPSK en el LTSpice. Las señales  $s1$  y  $s1_f$  representan la señal original modulada QPSK y ésta luego de haber pasado por el filtro, respectivamente. Como en casos anteriores, la escala no permite visibilizar la forma de estas señales mostrando todo el tiempo de simulación. Es por ello que más adelante se muestra un intervalo de simulación más corto, que permite ver en detalle la forma de  $s1$  y  $s1_f$ . Las señales  $i1a$  y  $q1a$  son las señales en fase y cuadratura, así como  $i1a_f$  y  $q1a_f$  representan las señales anteriores después de ser filtradas. Se puede observar que cada una de las señales I/Q toma 1 de 4 valores posibles en cada período de símbolo, donde la magnitud de estos valores van a cambiar en función de los desfases de los osciladores locales

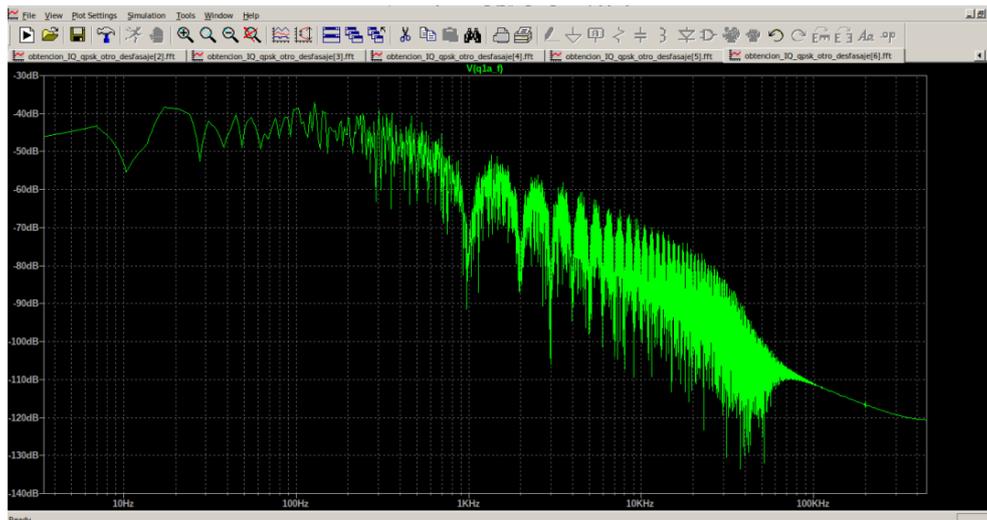


Figura 5.26: Espectro de la señal Q QPSK filtrada.

con los que se generen estas señales. Cuatro combinaciones posibles de los valores que toman las señales en fase y cuadratura generarán los cuatro símbolos de la modulación QPSK.

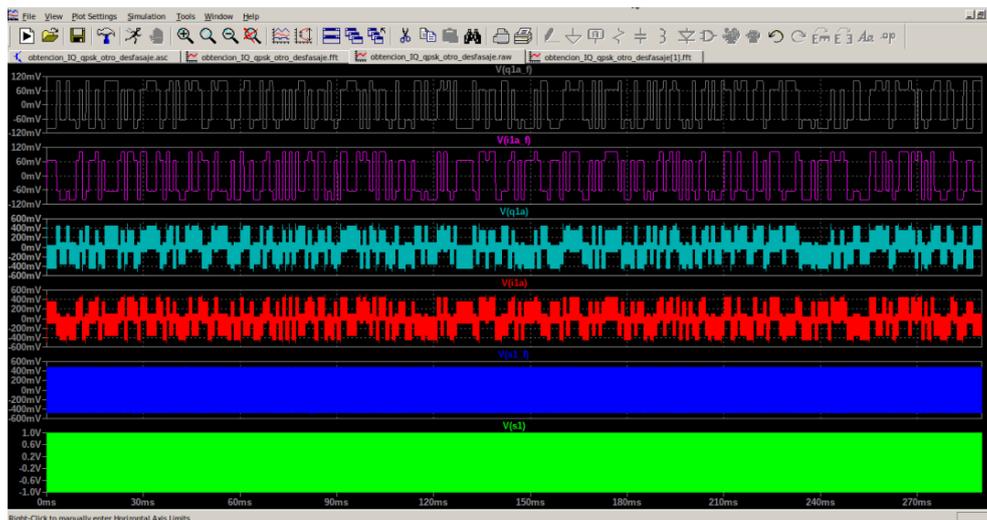


Figura 5.27: Señales simuladas QPSK en LTSpice.

Un acercamiento a la imagen para observar mejor el comportamiento de las señales durante la simulación se muestra en la Figura 5.28. En ella se ven 15 ms de simulación en los cuales, dado que el tiempo de símbolo es aproximadamente 1 ms, se pueden ver 15 cambios de fase, es decir, 15 símbolos. En concordancia con lo mencionado en la sección 5.1.3 sobre el funcionamiento del codificador convolucional y según el proceso de llenado de la FIFO visto en 5.1.4, se puede ver que los primeros bits almacenados son 10 10 10 01 00 00 01 00 00 10 11 00 01 10 10, que tomados de a dos por vez determinarán los símbolos QPSK. Se puede comprobar en la imagen que el comportamiento de las señales I/Q siguen el patrón de los datos

de la FIFO.

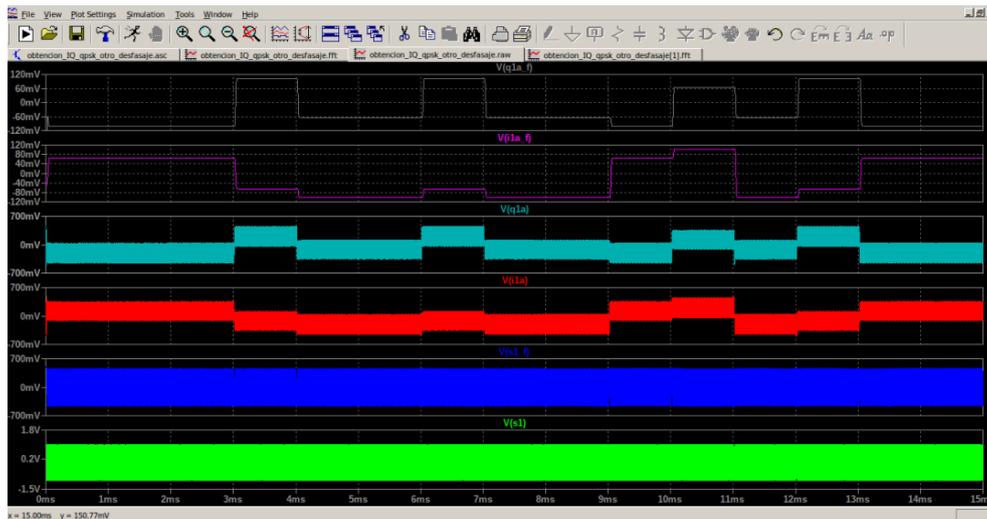


Figura 5.28: Primeros 15 ms de simulación QPSK en LTSpice.

La Figura 5.29 muestra un zoom de la simulación, el cual es suficiente para ver los períodos de portadora de la señal modulada. Se puede observar un cambio de fase alrededor  $t \approx 6,01$  ms en la señal original, donde hay un ensanchamiento del pulso. Este cambio es comprobable también en el resto de las señales de la imagen.

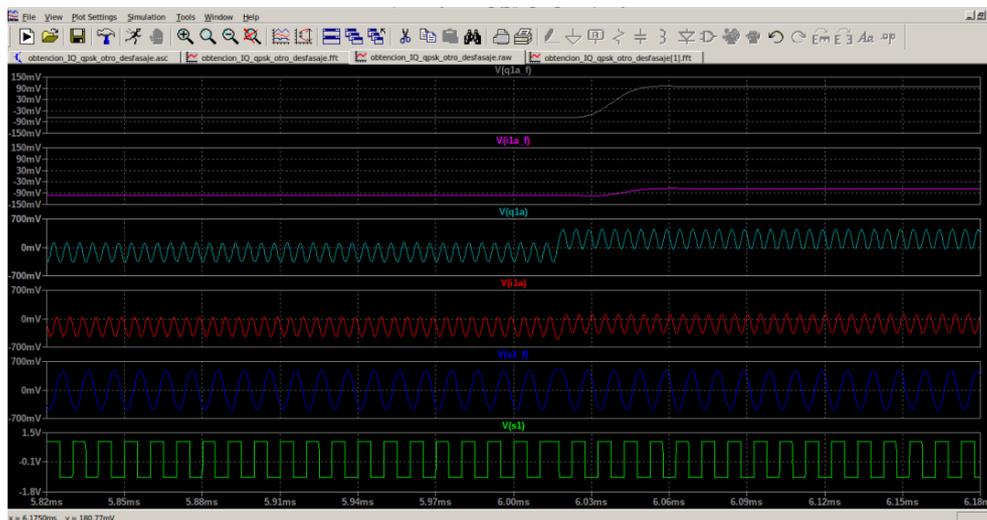
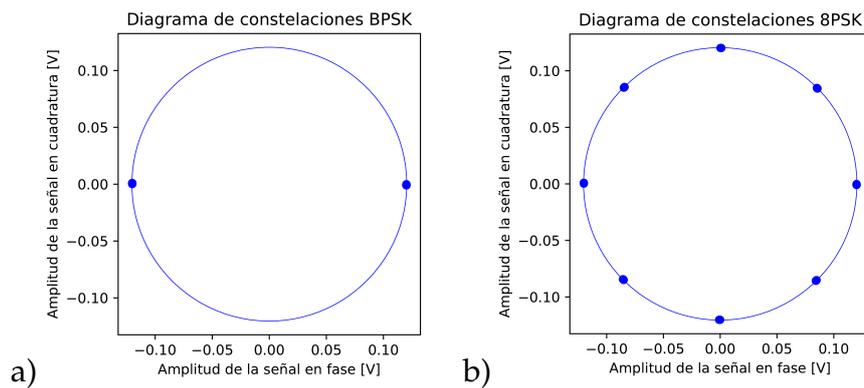


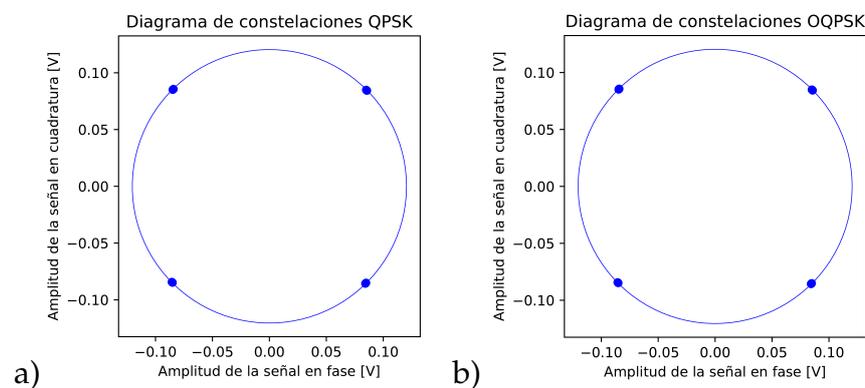
Figura 5.29: Visualización de un cambio de fase QPSK en LTSpice.

Los resultados de las señales en fase y cuadratura filtradas son exportadas del LTSpice y graficadas en Python. El resultado de graficar I versus Q en los puntos donde ocurren los cambios de fase, da como resultado los diagramas de constelaciones. Las Figuras 5.31 y 5.30 muestran los diagramas de constelaciones de todas las modulaciones desarrolladas en el trabajo.

Tal como es de esperar, a una misma entrada de datos a modular, las constelaciones QPSK y OQPSK son las mismas, pero la diferencia se puede notar con los

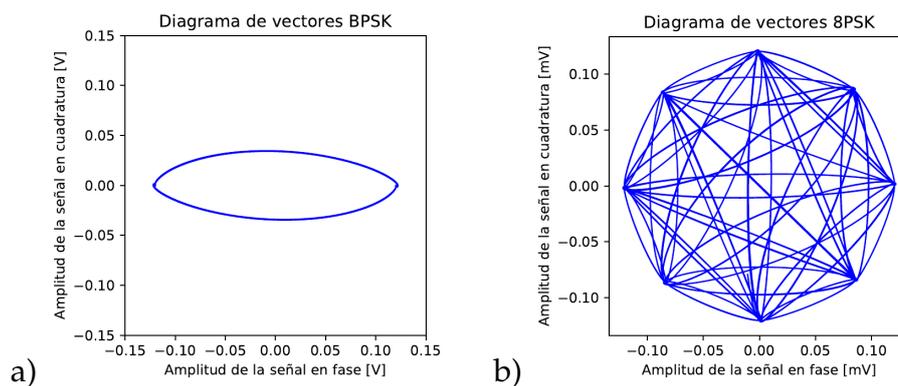


**Figura 5.30:** Diagrama de constelaciones a) BPSK y b) 8PSK.



**Figura 5.31:** Diagrama de constelaciones a) QPSK y b) OQPSK.

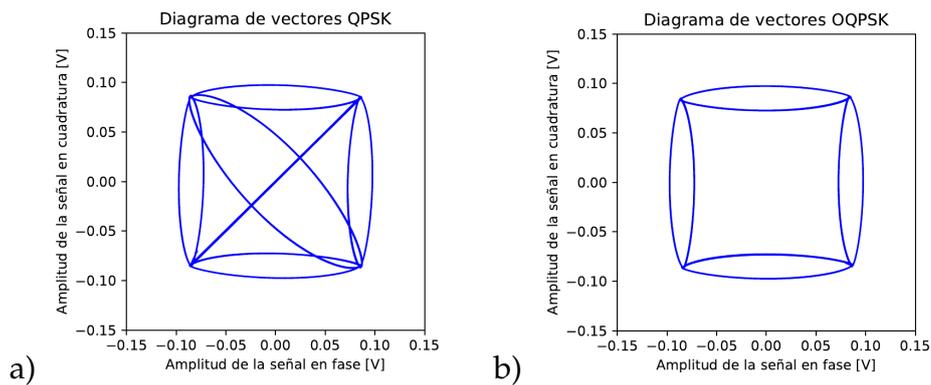
diagramas de vectores. Las Figuras 5.32 y 5.33 muestran estos gráficos para cada una de las modulaciones. Se puede distinguir que en la modulación QPSK existen cruces por cero, mientras que en OQPSK las transiciones ocurren evitando este punto.



**Figura 5.32:** Diagrama de vectores a) BPSK y b) 8PSK.

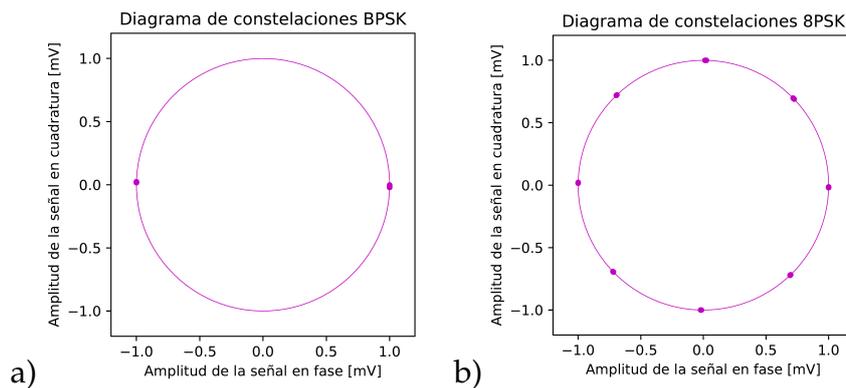
### 5.3.2. Obtención de las constelaciones con métodos numéricos

La implementación del procedimiento numérico descrito en la Sección 4.2.2 fue desarrollada en Python. La misma se aplicó al conjunto de datos exportados por el

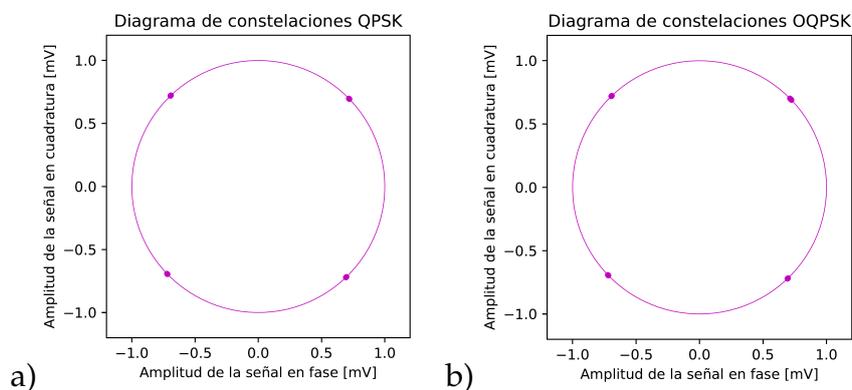


**Figura 5.33:** Diagrama de vectores a) QPSK y b) OQPSK.

*ModelSim*, es decir, el conjunto de trenes de pulsos que definen las señales moduladas. Los diagramas de constelaciones obtenidos como resultado de la aplicación del método para cada una de las modulaciones (BPSK, QPSK, OQPSK y 8PSK) son mostrados en las Figuras 5.34 y 5.35.



**Figura 5.34:** Diagrama de constelaciones a) BPSK y b) 8PSK.



**Figura 5.35:** Diagrama de constelaciones a) BPSK y b) 8PSK.

# Conclusiones y perspectivas

Los grandes avances en tecnología satelital, sobre todo alrededor del segmento de pequeños satélites, sirvieron de contexto para el planteamiento de este Proyecto Final Integrador. Mantener el control sobre estas aeronaves en todo momento a través de los enlaces de Telemetría, Telecomando y Control es sumamente importante, por lo que es necesario establecer enlaces de comunicaciones robustos. Las modulaciones PSK con una cantidad de símbolos menor a 16 tienen tasas de errores de bits que garantizan que este tipo de enlaces sean confiables. Es por ello que en este trabajo se desarrolló un modulador digital configurable basado en FPGA, que permite escoger entre BPSK, QPSK, OQPSK u 8PSK según las necesidades de comunicación en cada momento. La decisión de que el diseño del modulador se realice sobre FPGAs radica en que estos dispositivos suelen ser la base de una amplia mayoría de las plataformas de los pequeños satélites, por lo cual integrar este bloque generaría en principio un impacto muy bajo sobre el costo de la solución, en el sentido amplio del concepto. A su vez, la técnica planteada para generar las modulaciones también fue pensada para ser muy eficiente en términos de utilización de recursos, contribuyendo así también a la economía de la solución.

Todo el diseño de la arquitectura del modulador se llevó a cabo en VHDL, que al ser un lenguaje de descripción de hardware estándar permite ser implementado en cualquier FPGA sin importar el tipo o fabricante.

También se incluyeron en el diseño un bloque randomizador, que además agrega una marca de sincronismo, y un codificador convolucional, siguiendo las recomendaciones del estándar *CCSDS 131.0-B-3 para sincronización y codificación de canal*. Debido a que el período de símbolo de la señal modulada es mucho mayor que el tiempo de procesamiento de los datos en los bloques randomizador y codificador, fue necesario incluir una estructura de almacenamiento tipo First In First Out que guarde los datos a modular hasta tanto los moduladores los necesiten.

La entrada de datos a transmitir, así como la configuración de la modulación y la selección de ciclos de portadora por símbolo, se realiza a través de interfaces AHB y APB, respectivamente. La elección de éstas, basadas en protocolos estandarizados, permite el uso del modulador digital desarrollado como una caja negra, que puede ser incluido fácilmente en diseños más complejos.

El diseño es sintetizable en términos de uso de recursos, ya que ocupa aproximadamente el 7,1 % de los recursos de una Radiation-Tolerant ProASIC3 RT3PE3000L y menos del 1 % de la memoria RAM disponible. Esta familia de FPGA es preferida para aplicaciones satelitales por las ventajas que provee al ser tolerante a la radiación. También se realizó la comparación con la FPGA UltraScale+ ZU1CG, debido a que su rendimiento y nivel de integración que proveen las hacen también preferidas en aplicaciones satelitales. Del análisis se obtuvo que el diseño ocupa un 12,1 % de los bloques CLB LUTs y el 1,1 % de los CLB Flip-Flops provistos por la placa. Además, utiliza menos del 1 % del total de memoria RAM disponible.

La verificación del correcto funcionamiento del diseño fue llevada a cabo a través de la realización de una serie de testbenches simulados en el software ModelSim. En él se comprueba que los desfases ocurridos en la señal de salida del modulador digital son los esperados. Además, se verifica que la señal ha sido randomizada y codificada de manera correcta, antes de pasar por el bloque modulador.

Una segunda verificación se realizó a través de la obtención de los diagramas de vectores y diagramas de constelaciones de cada una de las modulaciones. Por un lado, las representaciones gráficas se lograron a partir de las señales en fase y cuadratura de la señal modulada de salida de la FPGA. La señal exportada del ModelSim es procesada, a través de un script de Python, para hacerla compatible con el programa LTSpice. En este software la señal importada es filtrada, mezclada con dos osciladores desfasados  $90^\circ$  entre sí y filtradas de nuevo para generar las señales I/Q en banda base. Estos resultados son exportados y graficados para obtener los diagramas. Por otro lado, a la señal modulada de salida de la FPGA se le realizan procesamientos numéricos con los cuales se obtienen también diagramas de constelaciones que constatan que el modulador funciona como es esperado.

Se proponen como posibles trabajos futuros relacionados la implementación física del modulador, así como también el diseño de un demodulador digital análogo al modulador desarrollado en este trabajo para complementar en lo que respecta a la recepción de telecomandos.

# Capítulo 6

## Apéndices

### 6.1. Simulación en el programa ModelSim

En este apéndice se muestran las imágenes que verifican el funcionamiento de los moduladores BPSK, QPSK y 8PSK a partir de las simulaciones realizadas en el software ModelSim., como complemento de lo explicado en la sección 5.1.

#### 6.1.1. Simulaciones generales

En las Figuras 6.1-6.3 se observa la simulación general de cada modulación que permite comprobar que se han configurado correctamente los ciclos de portadora y la modulación escogida, y que todos los bits almacenados en la FIFO son leídos, al analizar la señal *EMPTY\_i*.

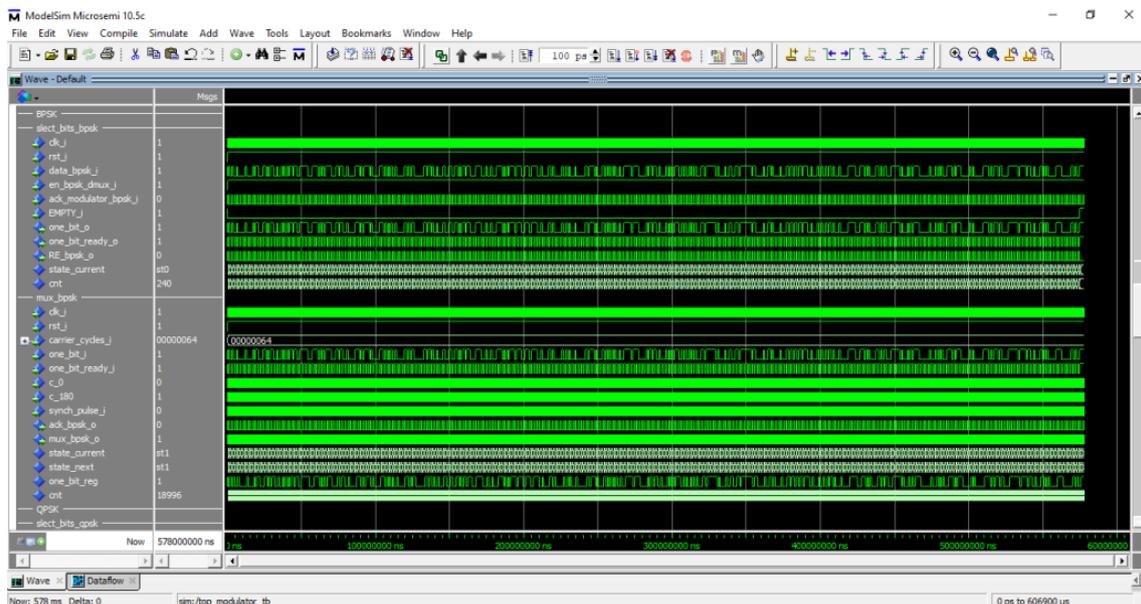


Figura 6.1: Vista general de la simulación BPSK.

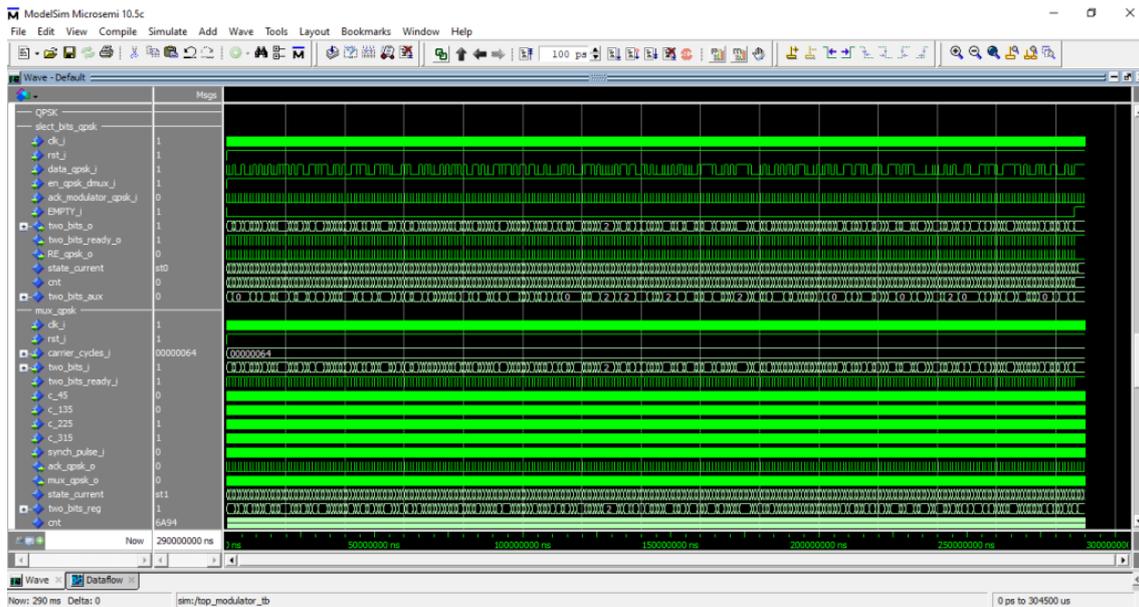


Figura 6.2: Vista general de la simulación QPSK.

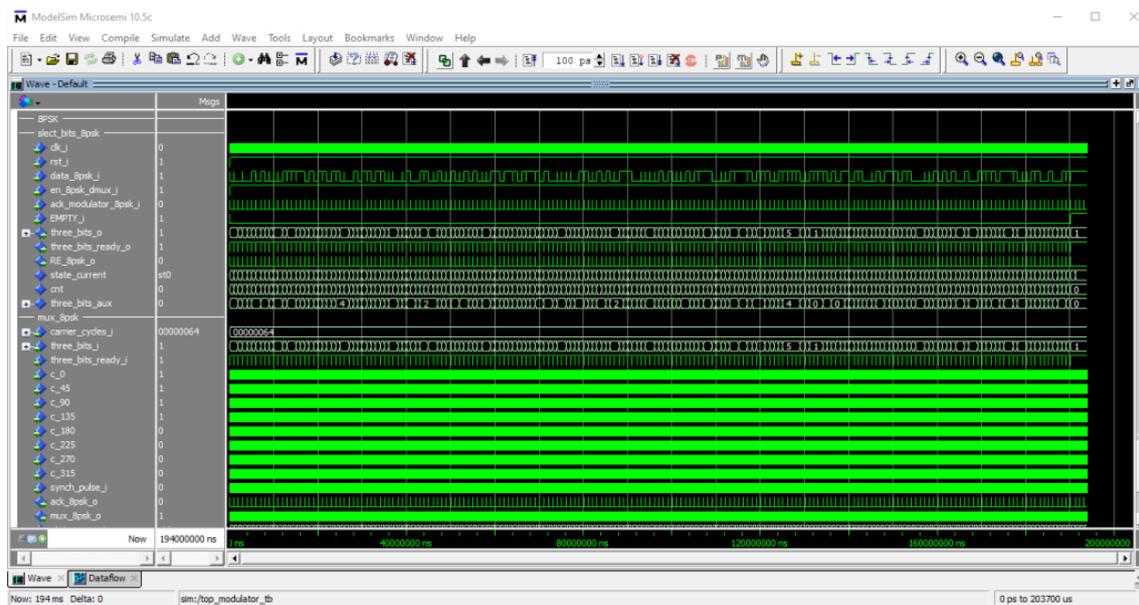


Figura 6.3: Vista general de la simulación 8PSK.

## 6.1.2. Verificación de las señales portadoras

Se puede observar en las imágenes 6.4-6.6 las señales portadoras generadas. En el caso de BPSK, que cuenta con dos portadoras:  $c_0$  y  $c_{180}$ , el tiempo medido por los cursores es  $5004\text{ ns}$ , con lo que se puede inferir que las portadoras están desfasadas  $180^\circ$ . La diferencia de tiempos entre cursores es de  $2502\text{ ns}$  para QPSK y  $1251\text{ ns}$  para 8PSK, por lo que los desfases son efectivamente  $90^\circ$  y  $45^\circ$ , respectivamente. En todos los casos se puede comprobar que el periodo de portadora es  $10008\text{ ns}$ .

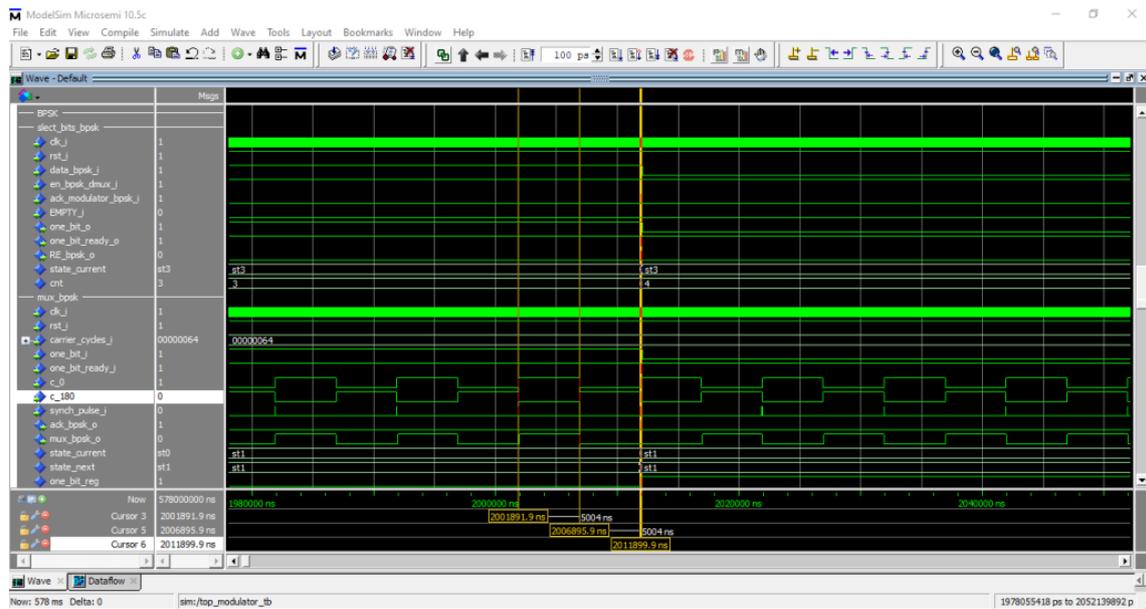


Figura 6.4: Señales portadoras BPSK.

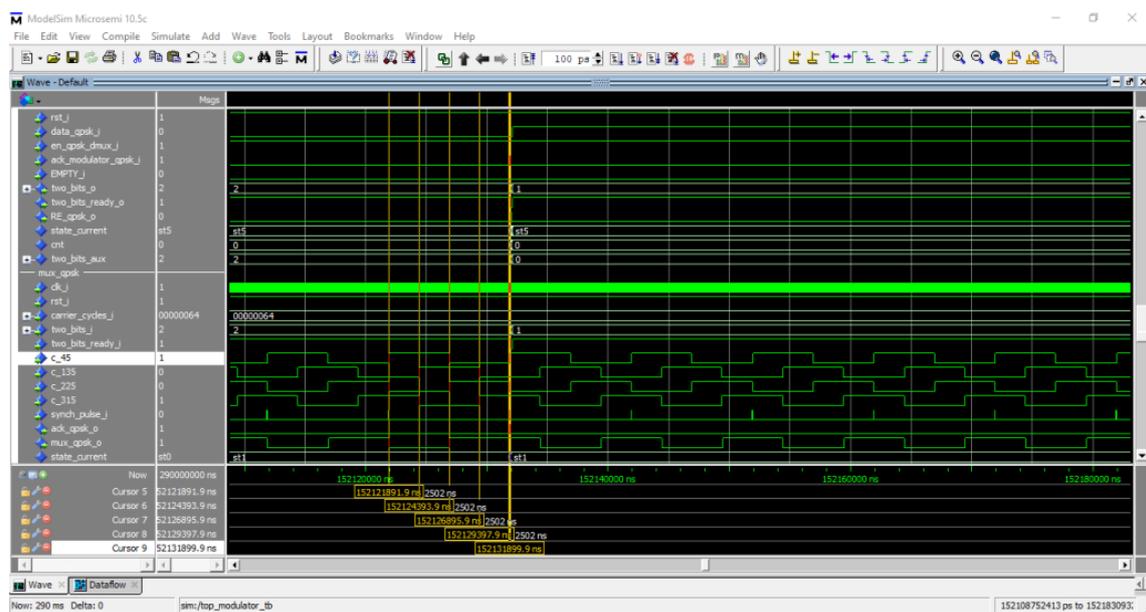


Figura 6.5: Señales portadoras QPSK.

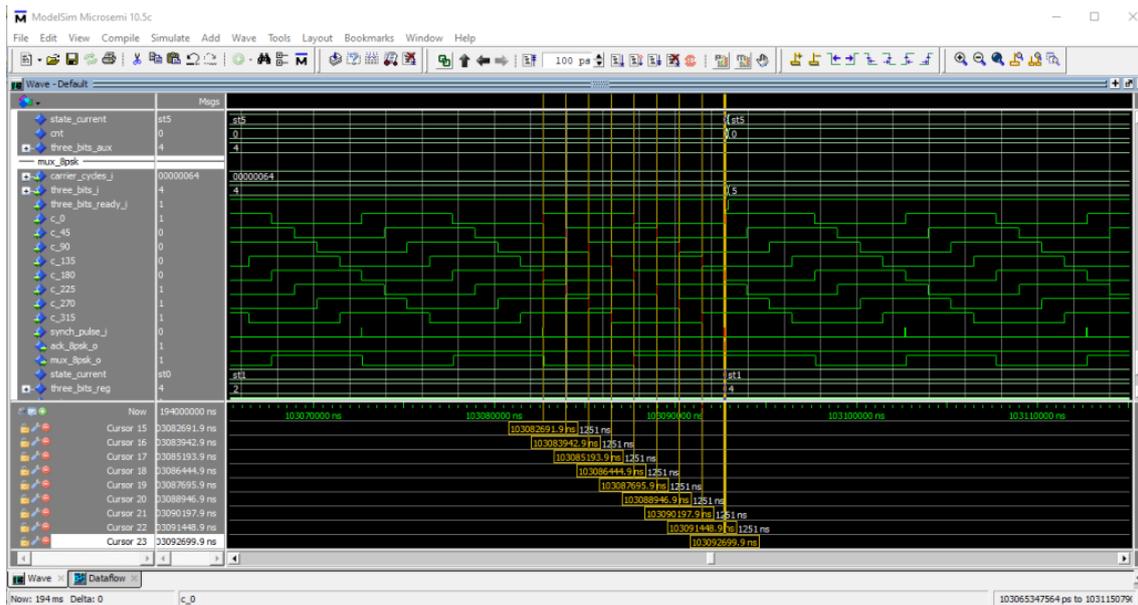


Figura 6.6: Señales portadoras 8PSK.

### 6.1.3. Lectura de la FIFO

El proceso de lectura de la estructura de almacenamiento se puede ver en las Figuras 6.7-6.9. Como para la modulación BPSK es necesario solamente un bit para representar un símbolo, la señal *RE\_bpsk\_o* está activa sólo durante un ciclo de reloj por cada proceso de lectura. Tal como especifica el manual de la FIFO, el dato está disponible dos ciclos de reloj después de haberse generado el acceso. Este bloque de selección de bits, entrega en el estado *st3* el bit leído al mismo tiempo que levanta una señal (*one\_bit\_ready\_o*) para avisar que el dato está disponible.

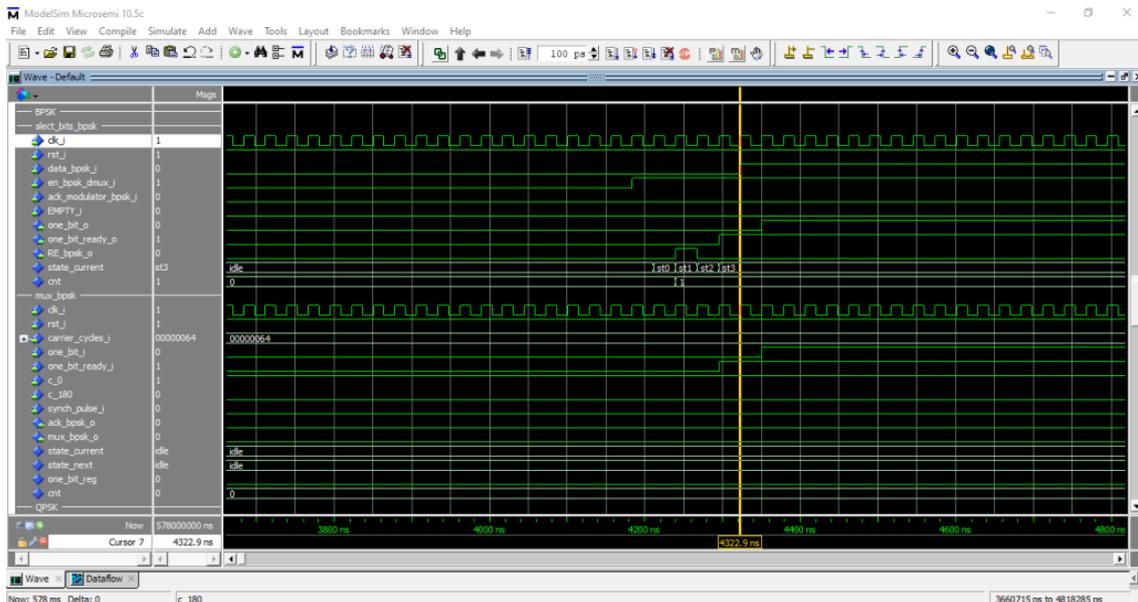


Figura 6.7: Ejemplo de selección de bits en la modulación BPSK.



### 6.1.4. Verificación de los cambios de fase

En cada una de las imágenes 6.10-6.12 se puede verificar la ocurrencia de cambios de fase. Se observan ensanchamientos de los pulsos de la señal modulada debido al cambio de la señal portadora en los instantes de tiempo correspondientes.

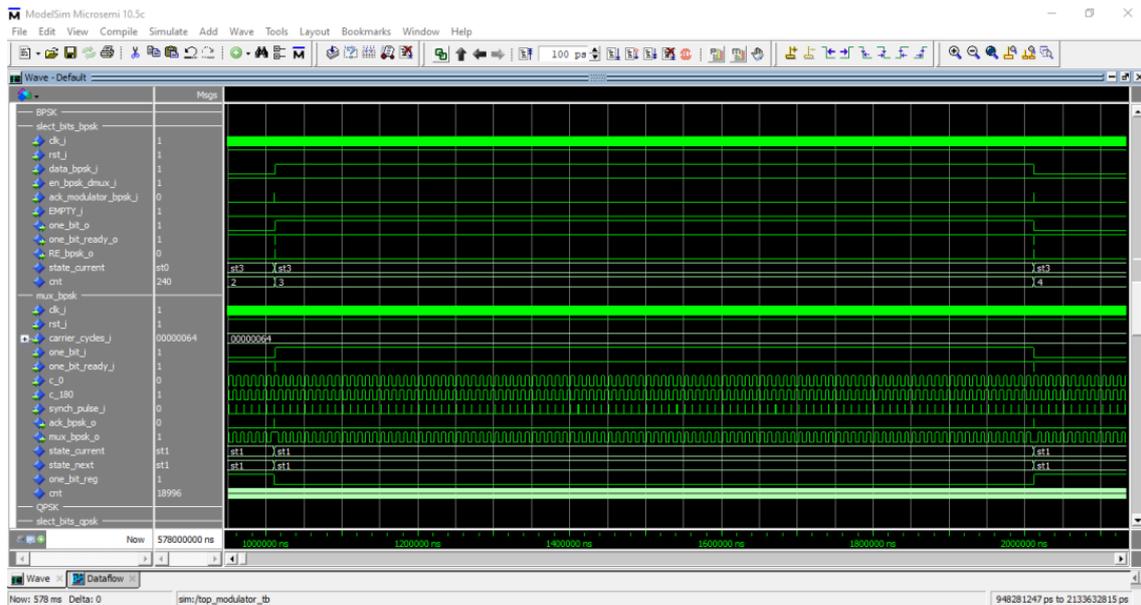


Figura 6.10: Visualización de cambios de fase en modulación BPSK.

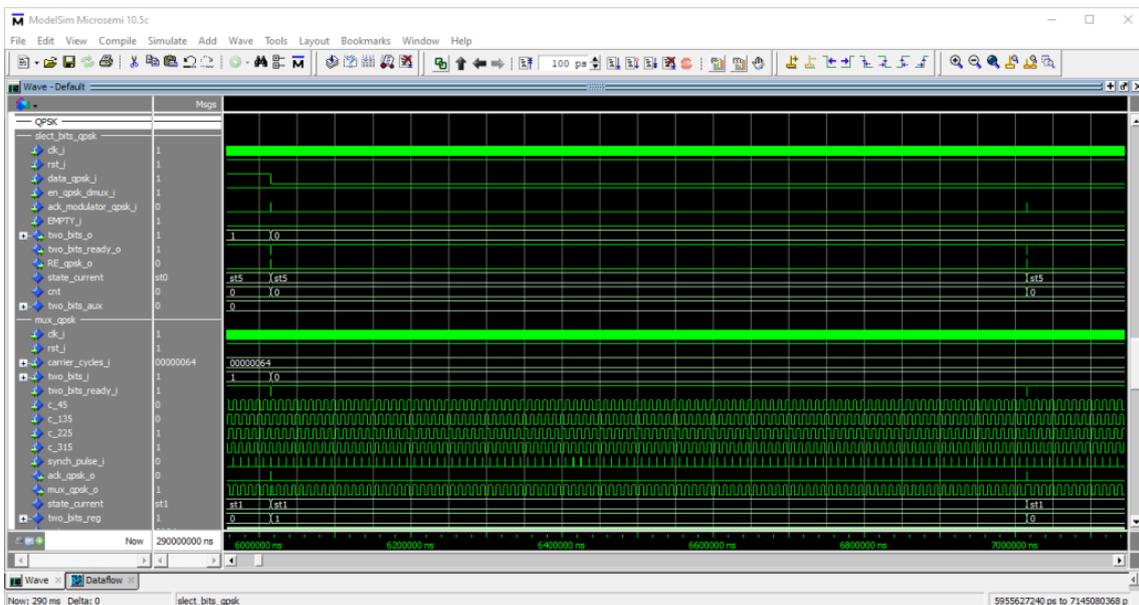


Figura 6.11: Visualización de cambios de fase en modulación QPSK.

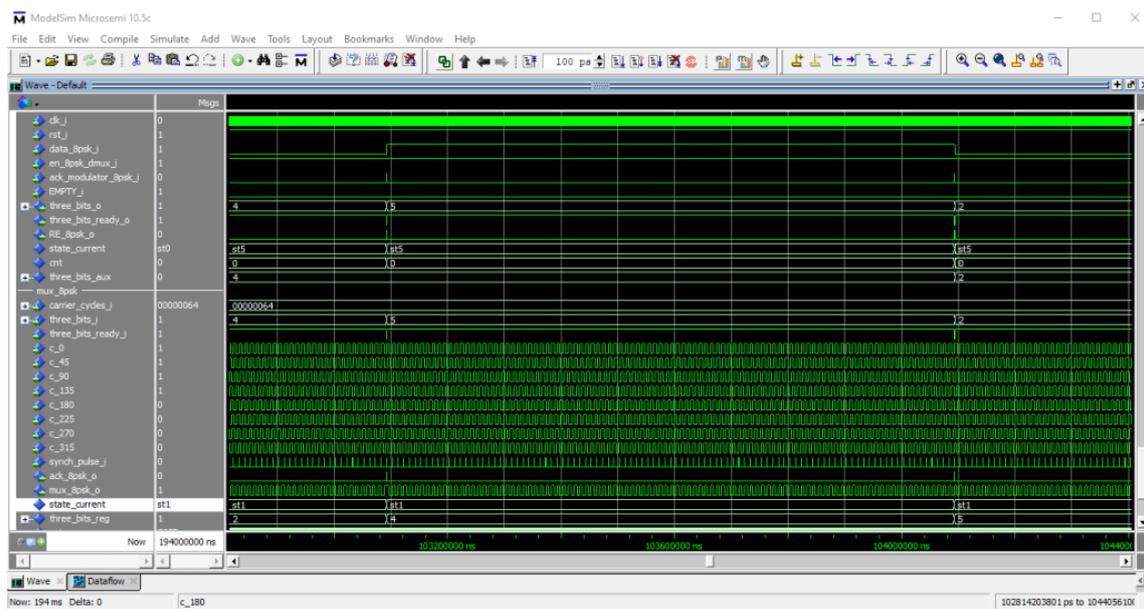


Figura 6.12: Visualización de cambios de fase en modulación 8PSK.

### 6.1.5. Verificación del período de símbolo

Los cursores de las Figuras 6.13-6.15 al inicio de cada símbolo miden el periodo de símbolo, mostrando así que es 1000800 ns tal como se esperaba.

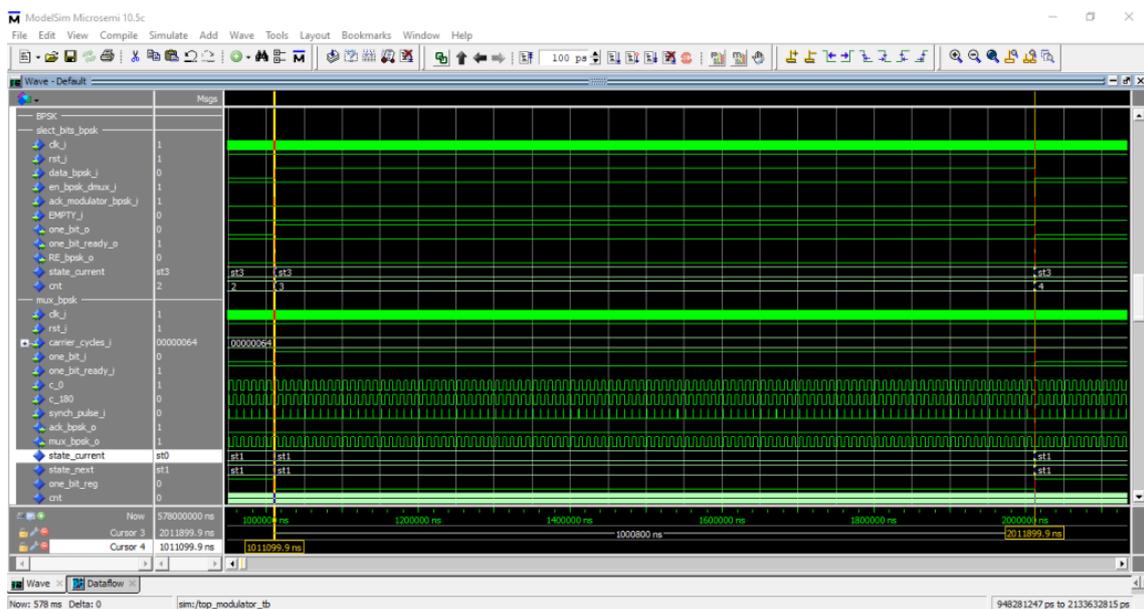


Figura 6.13: Medición del período de símbolo en modulación BPSK.

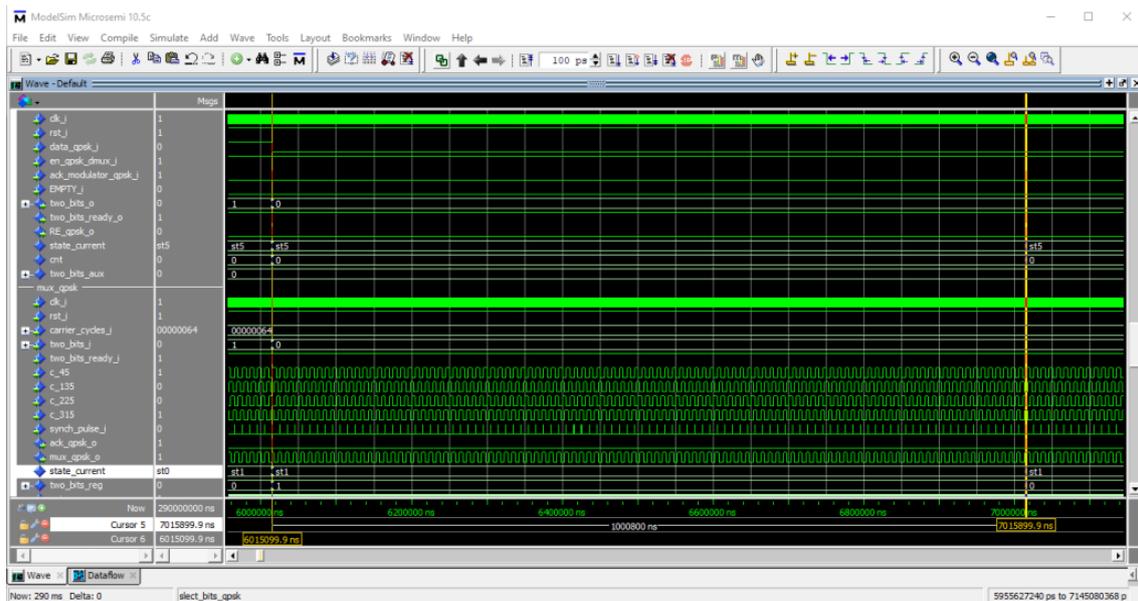


Figura 6.14: Medición del período de símbolo en modulación QPSK.

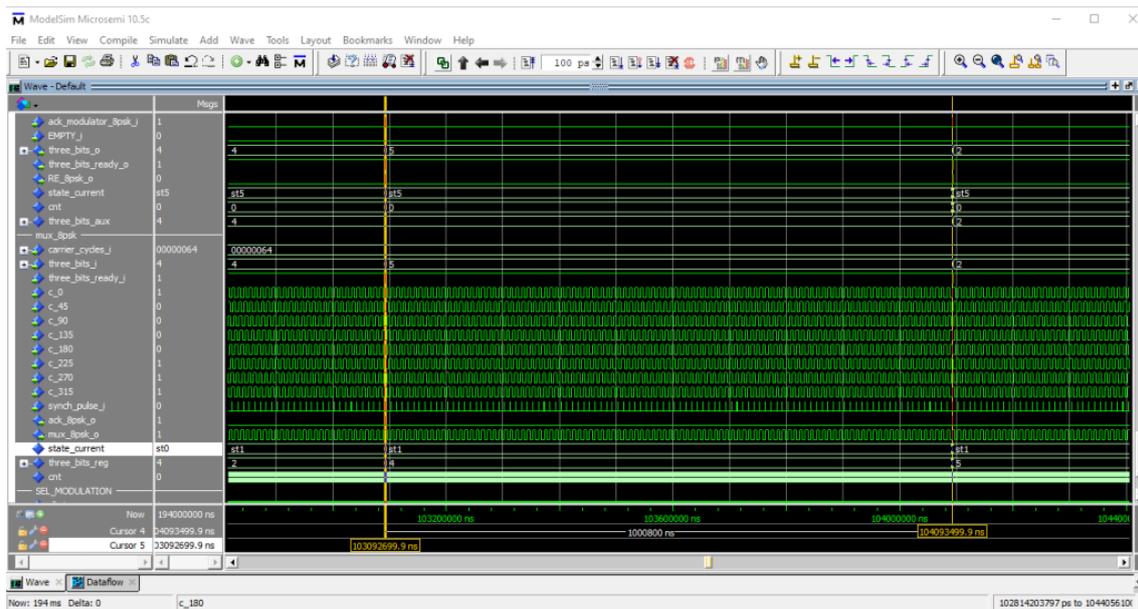


Figura 6.15: Medición del período de símbolo en modulación 8PSK.

## 6.1.6. Proceso de elección de portadora

En el caso de la modulación BPSK mostrada en la Figura 6.16, se observa que la señal modulada (`mux_bpsk_o`) antes del cursor es la misma que la portadora `c_0`, la cual corresponde al valor '0' de la señal `one_bit_reg`, el bit que fue registrado luego de ser leído de la FIFO. Luego del instante en el que se sitúa el cursor, el bit leído es '1', que se encontraba en el bus de la señal `one_bit_i`. Este valor pasa a registrarse y la señal modulada cambia en consecuencia a `c_180`.

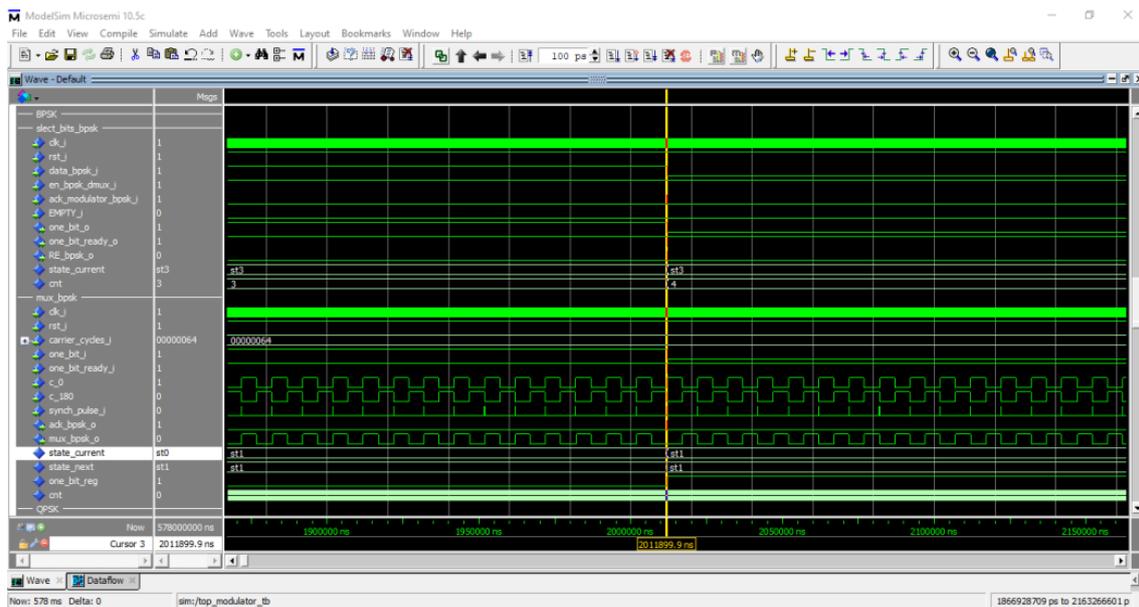


Figura 6.16: Proceso de selección de la señal portadora BPSK.

Como en el caso anterior, para la modulación QPSK (Figura 6.17) se puede ver que la señal modulada (*mux\_bpsk\_o*) antes del instante marcado por el cursor es la portadora *c\_0*. Esta señal corresponde al valor '00' registrado en *two\_bits\_reg* luego de haberlo leído de la FIFO. Después de pasar el período de símbolo, se registra el nuevo valor extraído de la estructura de almacenamiento, esta vez '01'. La señal modulada, tal como se espera, pasa a tomar los valores de la portadora *c\_315*.

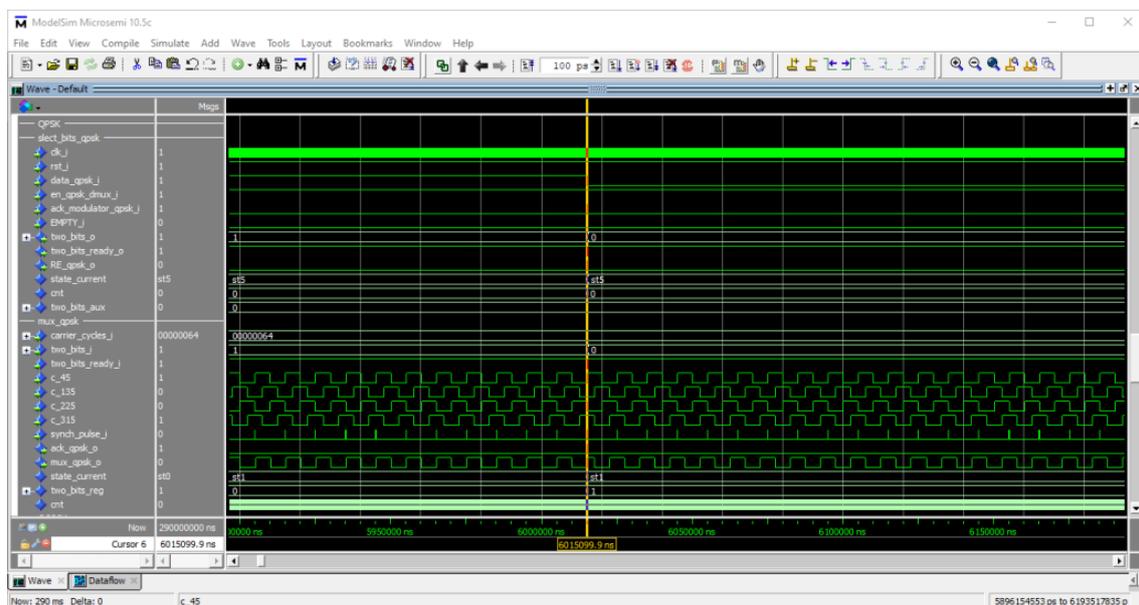


Figura 6.17: Proceso de selección de la señal portadora QPSK.

La modulación 8PSK no es diferente a las anteriores, la selección de la portadora dependerá de los bits leídos de la FIFO, ahora tres por vez. Como se observa en la Figura 6.18 la señal *three\_bits\_i* tiene un valor de '100', que será el dato a modular