



Universidad Autónoma de Querétaro
Facultad de Informática
Ingeniería en Telecomunicaciones

Transmisor Digital de Datos con Modulación BPSK mediante Tecnología FPGA

Opción de titulación **Tesis**

Que como parte de los requisitos para obtener el Grado de
Ingeniero en Telecomunicaciones

Presenta:
Luis Angel Reyes Cruz

Dirigido por:
Dr. Jesús Carlos Pedraza Ortega

Dr. Jesús Carlos Pedraza Ortega
Presidente

Firma

Dr. Juan Manuel Ramos Arreguín
Secretario

Firma

M.C. Guillermo Díaz Delgado
Vocal

Firma

Dr. Saúl Tovar Arriaga
Suplente

Firma

Dr. Efrén Gorrostieta Hurtado
Suplente

Firma

Centro Universitario
Querétaro, Qro.
Junio de 2015

RESUMEN

En éste trabajo se presenta el diseño de un transmisor de datos digital utilizando un esquema de detección y corrección de errores, además de una modulación por desplazamiento de fase (BPSK). Este diseño es implementado en una tarjeta de desarrollo con tecnología FPGA. Se propone una metodología que consiste en un análisis de los componentes y características fundamentales que involucran a los sistemas de comunicaciones en general; posteriormente se llevó a cabo un análisis de los elementos involucrados en las comunicaciones digitales con énfasis en la etapa de transmisión, de manera que se puedan describir las consideraciones y procesos que deben tenerse en cuenta para su implementación. Teniendo como base lo anterior, se desarrolló el modelo del transmisor definiendo cada una de sus etapas y comprobando este modelo mediante una simulación en Simulink. Con el esquema definido de las etapas del transmisor, se elaboró cada una de ellas en VHDL, validando el correcto funcionamiento de las etapas mediante el *toolbox* Xilinx System Generator. Una vez validado el funcionamiento del modelo, se sintetizó el código VHDL para su implementación en una tarjeta de desarrollo FPGA, se utilizó un osciloscopio digital para visualizar las señales de salida. Por último se analizaron los resultados obtenidos de la implementación y se compararon con los resultados de la simulación. Se evaluaron los recursos utilizados de la tarjeta de desarrollo, para el transmisor; se describieron las limitaciones y mejoras posibles para trabajo futuro.

(Palabras clave: transmisor, BPSK, VHDL, FPGA)

SUMMARY

This paper involves the design of a digital data transmitter using an error detection and correction scheme, and a phase-shift keying (BPSK) modulation. This design is implemented on a development board with FPGA technology. It is proposed a methodology that involves the analysis of the components and fundamental characteristics about the communications system in general. Later on it is carried out an analysis of the involved elements of the digital communications with an emphasis of the transmission stage, so that it can describe the considerations and processes that it needs to take into account for the implementation. Considering the previous statement the transmission model was developed, defining each of the stages and testing the model through a simulation on Simulink. As the defined scheme of the transmission stages, each of these were developed in VHDL, checking the correct performance of the stages through the Xilinx System Generator toolbox. Once the model performance has been checked, the VHDL code is synthesized for the implementation on a FPGA development board. A digital oscilloscope was used to visualize the output signals. Finally the obtained results of the implementation were analyzed and compared with the simulation results. The employed resources of the development board for the transmitter were evaluated. The limitations and improvements were described for future work.

(Keyword: transmitter, BPSK, VHDL, FPGA)

A mis padres, por todo el apoyo incondicional durante mi formación académica y
personal.

A mi hermana, por ser siempre un motivo para superarme.

A mi novia, por su apoyo y cariño brindado.

A mi director de tesis por su dedicación y vocación, siempre buscando la manera
de ayudar a los estudiantes.

A mis amigos, que juntos crecimos y nos ayudamos a lo largo de éste camino.

AGRADECIMIENTOS

Este trabajo es la culminación del esfuerzo de mi familia por siempre querer brindarme una formación académica de calidad, por los sacrificios realizados para que yo cumpliera mis objetivos.

A mis padres les debo todo lo que soy, me facilitaron mi camino hasta aquí y fueron un gran soporte. El saber que para mi hermana soy un ejemplo a seguir, me hace querer esforzarme cada día.

Al Dr. Jesús Carlos Pedraza, le agradezco todo el apoyo brindado durante cuatro años, todos los consejos y toda la ayuda para mi crecimiento académico, sin duda alguna, fue una guía invaluable.

Mis amigos, Enrique y Andrés, porque juntos recorrimos el mismo camino durante tres años, nos apoyamos mutuamente, siempre con el objetivo de mejorar y de aprender cosas nuevas, enfrentando los retos juntos.

A mis profesores, aquellos que entienden y ponen en práctica el lema de la Universidad, que se entregan a su profesión con vocación y por ende se entregan a los alumnos. Ustedes hacen grande a ésta facultad.

A la Universidad Autónoma de Querétaro, mi alma máter, que fue mi segunda casa por casi 8 años, me permitió tener grandes experiencias, conocer magnificas personas en su rol de compañeros, amigos y profesores. Me llevo conmigo todas esas cosas buenas, además de la satisfacción de que siempre busque la forma de devolverle a la Universidad un poco de lo mucho que ella me brindó, también me voy con la certeza de que llevaré orgulloso el nombre de la Universidad, a donde quiera que vaya y de la mejor forma posible.

ÍNDICE

AGRADECIMIENTOS.....	V
ÍNDICE.....	V
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS	XI
1. INTRODUCCIÓN	1
1.1 DEFINICIÓN DEL PROYECTO DE INVESTIGACIÓN.....	1
1.2 JUSTIFICACIÓN	2
1.3 OBJETIVOS	3
1.3.1 <i>Objetivo General:</i>	3
1.3.2 <i>Objetivos Específicos:</i>	3
1.4 ALCANCES Y LIMITACIONES.....	3
1.4.1 <i>Resultados generales que se pretenden con el proyecto de investigación</i>	4
1.4.2 <i>Participación en el proyecto de investigación</i>	4
1.5 ORGANIZACIÓN DE LA TESIS.....	5
2. MARCO TEÓRICO	6
2.1 SISTEMAS DE COMUNICACIONES	6
2.1.1 <i>Comunicaciones Digitales</i>	8
2.1.2 <i>Modulación</i>	10
2.1.3 <i>Capacidad de Información</i>	12
2.2 LENGUAJES DE DESCRIPCIÓN DE HARDWARE	13
2.3 LA FPGA	14
2.4 DESCRIPCIÓN DEL SOFTWARE Y HARDWARE	18
2.4.1 <i>Matlab-Simulink</i>	18
2.4.2 <i>Active-HDL</i>	22
2.4.3 <i>Xilinx ISE</i>	26
2.4.4 <i>Tarjeta de Desarrollo Spartan-3E Starter Kit</i>	28
3. TRANSMISORES DIGITALES	35
3.1 CODIFICACIÓN EN LOS TRANSMISORES DIGITALES DE DATOS	35

3.1.1	<i>Códigos de Hamming</i>	39
3.2	TIPOS DE MODULACIÓN DIGITALES.....	41
3.3	MODULACIÓN EN FASE (BPSK)	46
4.	DESARROLLO E IMPLEMENTACION	50
4.1	ANÁLISIS DE DISEÑO	51
4.2	SIMULACIÓN EN MATLAB/SIMULINK.....	53
4.2.1	<i>Simulación del Codificador</i>	53
4.2.2	<i>Simulación del Modulador</i>	54
4.3	ELABORACIÓN.....	55
4.4	SIMULACIÓN EN ACTIVE-HDL Y XSG	61
4.5	SÍNTESIS.....	63
4.6	IMPLEMENTACIÓN	66
5.	ANÁLISIS DE RESULTADOS	69
5.1	RESULTADOS DE LA SIMULACIÓN.....	69
5.2	RESULTADOS DE LA IMPLEMENTACIÓN	73
5.3	CONCLUSIONES	78
5.4	TRABAJO FUTURO.....	79
	REFERENCIAS	80

ÍNDICE DE FIGURAS

Figura 1. Esquema general de un sistema de comunicaciones (Wayne, 2003).....	7
Figura 2. Diagrama a bloques general de un sistema de comunicación digital (Couch II, 1998).	9
Figura 3. Categorías de los dispositivos lógicos programables (Aceves y Ramos, 2012). 14	
Figura 4. Estructura de los bloques de la FPGA (Xilinx Inc., 2011).	15
Figura 5. Estructura de los CLB's (Xilinx Inc., 2011).	16
Figura 6. Estructura de las LUT dentro de un CLB (Xilinx Inc., 2011).	17
Figura 7. Librerías de Simulink y bloques de XSG (Xilinx Inc., 2011).	20
Figura 8. Bloques fundamentales de XSG (Xilinx Inc., 2011).	21
Figura 9. Metodología para el desarrollo de circuitos digitales.	23
Figura 10. Interfaz de Active-HDL.	24
Figura 11. Visor de formas de onda de Active-HDL.	25
Figura 12. Interfaz de Xilinx ISE.....	26
Figura 13. Interfaz de PlanAhead.....	27
Figura 14. Interfaz de iMPACT.....	28
Figura 15. Tarjeta de desarrollo Spartan-3E Starter Kit (Xilinx Inc., 2011).	29
Figura 16. Ubicación y pin del oscilador interno de 50MHz de la Tarjeta Spartan-3E Starter Kit (Xilinx Inc., 2011).	30
Figura 17. Ubicación del DAC y de los pines de salida para los 4 canales (Xilinx Inc., 2011).	31
Figura 18. Esquema de Conexión de la Spartan-3E y el DAC LTC2624 (Xilinx Inc., 2011).	32

Figura 19. Protocolo de 24 bits de comunicación del DAC LTC2624 (Linear Technology Corporation, 2004).....	34
Figura 20. Tipos de Modulación (Wayne, 2003).....	41
Figura 21. Modulación ASK. a) Señal de información. b) Portadora. c) Portadora Modulada.	43
Figura 22. Modulación FSK. a) Señal de información. b) Portadora. c) Portadora Modulada.	44
Figura 23. Modulación BPSK. a) Señal de información. b) Portadora. c) Portadora Modulada.....	45
Figura 24. Diagrama de un modulador BPSK.	46
Figura 25. Fases de BPSK. a) Tabla de verdad. b) diagrama fasorial. c) Diagrama de constelación (Wayne, 2003).....	47
Figura 26. Espectro BPSK para una portadora de 100KHz y una moduladora de 100Hz.	49
Figura 27. Metodología utilizada.	50
Figura 28. Diagrama del Transmisor de datos digital.	51
Figura 29. Diagrama del Modulador BPSK.	53
Figura 30. Diagrama del Modelo del Codificador de Hamming.	54
Figura 31. Diagrama del Modelo del Modulador BPSK.	55
Figura 32. Entidad general del transmisor digital de datos.	57
Figura 33. Entidad del Acumulador de fase.	58
Figura 34. Entidad del Modulador.	59
Figura 35. Esquema de la Máquina de Estados Finita para el DAC.....	60
Figura 36. Simulación en Active-HDL.	61
Figura 37. Modelo de la simulación del transmisor en XSG.	63

Figura 38. Asignación de las señales de entrada y salida.....	65
Figura 39. Síntesis mediante Xilinx ISE.	65
Figura 40. Selección del archivo (.bit).	66
Figura 41. Programación de la tarjeta de desarrollo.....	67
Figura 42. Conexión de la tarjeta de desarrollo y el osciloscopio digital.	68
Figura 43. Resultados de la simulación de la codificación de Hamming en Simulink.	69
Figura 44. Resultados de la modulación BPSK en Simulink. a) Señal de información. b) Portadora. c) Señal BPSK.	70
Figura 45. Ancho de banda de la señal BPSK simulada.	71
Figura 46. Simulación en Active-HDL.	72
Figura 47. Simulación en XSG y Simulink.....	72
Figura 48 . Implementación en la Tarjeta de desarrollo.....	73
Figura 49. Visualización en el osciloscopio digital. a) Portadora. b) Señal de información.	74
Figura 50. Señal Modulada en el osciloscopio.	74
Figura 51. Señal binaria alternada y señal modulada.....	75
Figura 52. Espectro de la señal BPSK.	76
Figura 53. Bandas laterales del espectro BPSK.....	76

ÍNDICE DE TABLAS

Tabla 1. Comandos del DAC LTC2624.....	33
Tabla 2. Direcciones para la selección de canal en el DAC LTC2624.....	33
Tabla 3. Dispositivos deshabilitados.	34
Tabla 4. Desarrollo del código de Hamming (7,4).	52
Tabla 5. Tabla para la asignación de pines en la FPGA.....	64
Tabla 6. Utilización de recursos de la tarjeta de desarrollo.	77

1. INTRODUCCIÓN

El presente trabajo tiene como objetivo realizar el diseño de un transmisor de datos digitales, incorporando un codificador de canal y un modulador BPSK, implementado en una tarjeta de desarrollo FPGA. Todo esto usando las herramientas de diseño y modelado de Matlab-Simulink, Active-HDL, Xilinx ISE y System Generator.

1.1 Definición del Proyecto de Investigación.

En la actualidad los avances tecnológicos han tenido un gran efecto en la sociedad al grado de que en nuestra vida cotidiana ya no podemos prescindir de los servicios que ésta nos proporciona, como las comunicaciones (telefonía, internet, televisión, radio, etc.) y su evolución tan acelerada que nos permite como usuarios tener una variedad de opciones de compañías que ofrecen los servicios, reduciendo los costos y teniendo un notable aumento en la calidad del servicio. Esto se debe en gran parte a los avances que la tecnología electrónica ha tenido y en especial la electrónica digital, que ha evolucionado a las comunicaciones digitales, de manera que se están migrando los servicios de tecnología analógica a tecnología digital, debido a las grandes ventajas que poseen las comunicaciones digitales como: un menor ancho de banda necesario, mayor calidad en el servicio, mejora en la utilización de la banda de frecuencias, facilidad de procesamiento, facilidad de multiplexado e inmunidad al ruido (Romero, 2007).

Debido a lo anterior es de gran importancia como ingeniero en telecomunicaciones generar conocimientos prácticos sobre sistemas de comunicaciones modernos, conocer los procesos de desarrollo e implementación de sistemas digitales; de tal manera que permitan aplicar los conceptos aprendidos durante el transcurso de la carrera así como la generación de nuevos conocimientos

teóricos y prácticos en el área de las comunicaciones y los sistemas digitales (Sghaier, Areibi, & Dony, 2008).

1.2 Justificación

En base a lo anterior y con los conocimientos adquiridos durante la carrera, es posible implementar un transmisor digital eficiente mediante dispositivos lógicos programables que gracias a su arquitectura permitan la facilidad y rapidez de diseño, flexibilidad y portabilidad; capaz de funcionar de manera correcta en distintas áreas de aplicación como: aplicaciones RFID como un transmisor (Van de Beek, Ciacci, Al-Kadi, Kompan & Stark, 2012); en redes inalámbricas mediante la creación de un sistemas en FPGA's usando esquemas como OFDM y modulación BPSK, para la implementación del estándar IEEE 802.11 (García y Cumplido, 2005); en sistemas de comunicaciones de alta velocidad implementándose en arquitecturas de FPGA especiales para telecomunicaciones (Tsutsui, Miyazaki, & Ohta, 1995); en biomedicina como el diseño de un modulador para telemetría biomédica programada en VHDL, los cuales tienen varias aplicaciones en la ésta área, como los dispositivos transcutaneos que contienen sensores con tecnología similar a la de los sistemas RFID mediante señales de radiofrecuencia (Elamary, Chester & Neasham, 2009) y (Mandal & Sarpeshkar, 2008), etc.

El diseño de estos sistemas en tecnología FPGA es posible debido al diseño jerárquico y modular de tal manera que se puede adaptar realizando cambios en el software sin la necesidad de realizar un nuevo desarrollo en el hardware del transmisor; ahorrando costos, tiempos de implementación y desarrollo.

1.3 Objetivos

1.3.1 Objetivo General:

Diseñar e implementar un transmisor digital con una modulación BPSK mediante dispositivos lógicos programables de tecnología FPGA para la transmisión de datos.

1.3.2 Objetivos Específicos:

- Analizar las propiedades de la modulación PSK y las consideraciones para la transmisión de datos.
- Diseñar la descripción de la Arquitectura de hardware utilizando la técnica de diagramas descriptivos de bloques para su implementación en FPGA.
- Realizar un simulador mediante las herramientas: Matlab, Simulink y Xilinx System Generator del sistema de comunicación con modulación PSK para verificar la funcionalidad de la propuesta de implementación.
- Realizar pruebas del sistema y analizar los resultados obtenidos para validar el transmisor digital de datos implementado.

1.4 Alcances y limitaciones

Para el desarrollo e implementación del transmisor digital de datos es necesario definir los elementos y características que lo conformará, como el tipo de codificación y modulación. Estas características del transmisor dependen de factores como la capacidad de la tarjeta de desarrollo a utilizar, la frecuencia del oscilador y el tipo de convertidor. La frecuencia de trabajo del transmisor está limitada en estos tipos de desarrollos, por lo que la frecuencia de la portadora se

encontrara en el orden de las frecuencias bajas (LF, Low Frequency). Otra limitación del sistema es el transductor de salida para la propagación de la señal, ya que es necesario un circuito extra (con filtros y acopladores de señal) a la salida del convertidor para la propagación adecuada de la señal. Este circuito depende del medio de transmisión requerido, como puede ser un medio alámbrico o inalámbrico.

El alcance de este trabajo se limita a la implementación de los bloques del transmisor digital de datos, relacionados con el procesamiento y adaptación adecuada de la señal para una comunicación fiable, utilizando las ventajas que ofrece la tecnología FPGA.

1.4.1 Resultados generales que se pretenden con el proyecto de investigación.

- Análisis de un Transmisor Digital de Datos
- Simulaciones del sistema de transmisión con las herramientas de Matlab-Simulink y Xilinx System Generator.
- Diseño de la arquitectura de Hardware para el transmisor digital de datos y su correspondiente programación en VHDL.
- Implementación del transmisor digital en una tarjeta de desarrollo de tecnología FPGA.

1.4.2 Participación en el proyecto de investigación

Desarrollar el simulador del transmisor digital de datos mediante las herramientas de: Matlab, Simulink, Active-HDL y Xilinx System Generator. Una vez obtenida la simulación satisfactoriamente, proceder al diseño de la descripción del hardware para programarlo en VHDL y su posterior implementación en una tarjeta con tecnología FPGA así como la realización de pruebas del sistema completo para obtener los resultados y por último llevar a cabo un análisis y comparación de los

resultados obtenidos con los resultados esperados de acuerdo a la simulación, para así, obtener las conclusiones de este trabajo.

1.5 Organización de la tesis

En el Capítulo 2, se habla sobre la teoría utilizada a lo largo de la tesis, dando un panorama general acerca de los sistemas de comunicaciones y de los dispositivos programables con la finalidad de entender conceptos como: codificación, modulación, lenguajes descriptivos de hardware, las partes y elementos de una FPGA; así como la descripción del Hardware y del Software utilizado. Estos conceptos son muy importantes para el desarrollo de este trabajo, ya que son las partes claves para llevar a cabo una implementación adecuada y funcional.

En el Capítulo 3 se revisa la teoría sobre los sistemas de comunicaciones digitales, donde se define cada uno de los elementos que contendrá el transmisor digital. Éste Capítulo es muy relevante, debido a que sirve como preámbulo para la generación de la metodología para el desarrollo del transmisor, abarcando todos sus componentes.

El desarrollo del transmisor digital se muestra en el Capítulo 4, donde se describe cada uno de los pasos de la metodología utilizada; se analiza el diseño correspondiente a los requerimientos establecidos del transmisor; se realiza una simulación para validar las propiedades del diseño, posteriormente se desarrolla el diseño en VHDL y mediante otra simulación se verifica el funcionamiento adecuado del transmisor. La última etapa del desarrollo consiste en la síntesis y la implementación en hardware del sistema, usando la tarjeta de desarrollo especificada, además de la validación correspondiente del sistema.

En el Capítulo 5 se analizan los resultados de la implementación física del transmisor digital de datos. Se comparan los resultados obtenidos con los de las simulaciones realizadas y se evalúan los recursos utilizados de la FPGA, así como las posibles mejoras del sistema y el trabajo a futuro.

2. MARCO TEÓRICO

2.1 Sistemas de Comunicaciones

El objetivo básico de un sistema de comunicaciones, es la transferencia de información de un punto geográfico a otro. En general todos los sistemas de comunicaciones se componen de tres subsistemas principales: el transmisor, el canal de comunicación y el receptor.

El mensaje de la fuente en el transmisor está representado por la forma de onda de la información de entrada. El mensaje entregado en el receptor puede no ser el mismo que el enviado en el transmisor debido a distorsiones por el ruido de canal o atenuación que puede ocasionar el canal de comunicaciones.

La información en el mensaje puede ser análoga o digital, además de que puede haber una multiplexación en la entrada de tal manera que puedan existir múltiples fuentes de entrada y también múltiples salidas. Los espectros de frecuencias tanto de entrada como de salida son consideradas señales de banda base o señales de muy baja frecuencia.

En la parte del procesamiento de la señal en el transmisor, se condiciona la señal para una transmisión más eficiente, efectiva y confiable (que puede incluir: convertidores, ADC, filtros), además se puede añadir una codificación de canal que tenga la capacidad de detectar errores y en algunos casos corregirlos en el receptor.

La parte de la modulación en el transmisor convierte la señal de banda base a una banda de frecuencia en la que se desea transmitir y que sea adecuado para el canal, es decir, mediante este proceso, se acopla la señal a transmitir con las propiedades adecuadas para el canal de comunicación por medio de una onda portadora.

Los canales de comunicaciones o medios de transmisión, es el enlace entre el transmisor y el receptor; se pueden clasificar en alámbricos e inalámbricos. Estos medios pueden ser: un par de alambres, un cable coaxial, ondas de radiofrecuencia o enlaces de fibra óptica. Pero sin importar el tipo de medio de transmisión, todos se caracterizan por propiedades como la atenuación, es decir, la disminución progresiva en la potencia de la señal conforme aumenta la distancia, esta propiedad varía dependiendo el medio de transmisión y las características del sistema de comunicaciones; algunos otros factores como el ruido de canal, distorsión y múltiples trayectorias provocan cambios en la señal que viaja por el canal. En general las modulaciones digitales y analógicas se pueden aplicar a cualquier tipo de canal, aunque este favorezca a algunos tipos de modulación.

La función del receptor es extraer del canal la señal deseada y entregarla a la salida, para llevar a cabo lo anterior, en el receptor la señal distorsionada entra y se demodula para convertirse en una señal de banda base que se puede procesar en la salida del sistema de comunicación. Como las señales son frecuentemente muy débiles, como resultado de la atenuación, el receptor debe tener varias etapas de amplificación (Wayne, 2003). La siguiente figura muestra el esquema general de un sistema de comunicaciones con los elementos anteriormente descritos.

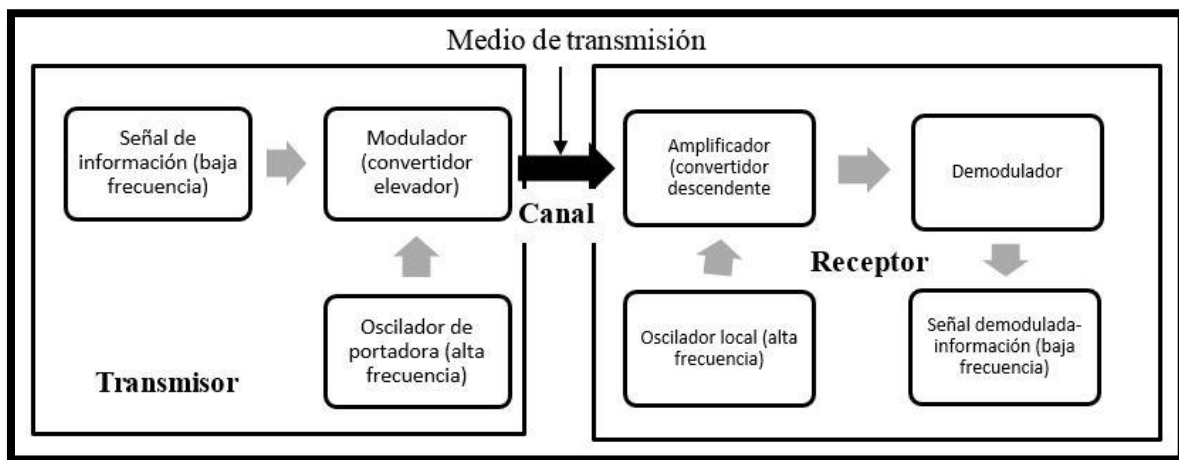


Figura 1. Esquema general de un sistema de comunicaciones (Wayne, 2003).

2.1.1 Comunicaciones Digitales

Los sistemas tradicionales de comunicaciones que usan esquemas de modulación analógica como la modulación de amplitud (AM), modulación de frecuencia (FM) y la modulación de fase (PM) se están sustituyendo rápidamente debido al gran avance de la electrónica digital y por consiguiente, el desarrollo de sistemas de comunicaciones digitales, los cuales son más modernos y poseen ventajas sobre los sistemas analógicos como la facilidad de procesamiento, facilidad de multiplexado e inmunidad al ruido. Los sistemas de comunicaciones digitales se encuentran conformados por portadoras de frecuencias relativamente altas que se modulan mediante señales de información digital de bajas frecuencias; además también se incluyen los sistemas que se basan en la transmisión de pulsos digitales.

Estos sistemas de comunicaciones se pueden dividir en la transmisión digital y la radio digital. En la transmisión digital se requiere un canal de comunicaciones físico entre el transmisor y el receptor, ya sea mediante un par de hilos, un cable coaxial o un cable de fibra óptica. Por otra parte, en los sistemas de radio digital, el medio de transmisión podría ser además la propagación en la atmósfera terrestre, mediante señales de radiofrecuencia. En estos sistemas de comunicaciones, la entrada de información puede ser digital o analógica, pero si la entrada es analógica, debe convertirse a un pulso digital mediante un convertidor analógico digital (ADC). La figura 2 muestra el diagrama a bloques general de un sistema de comunicación digital.

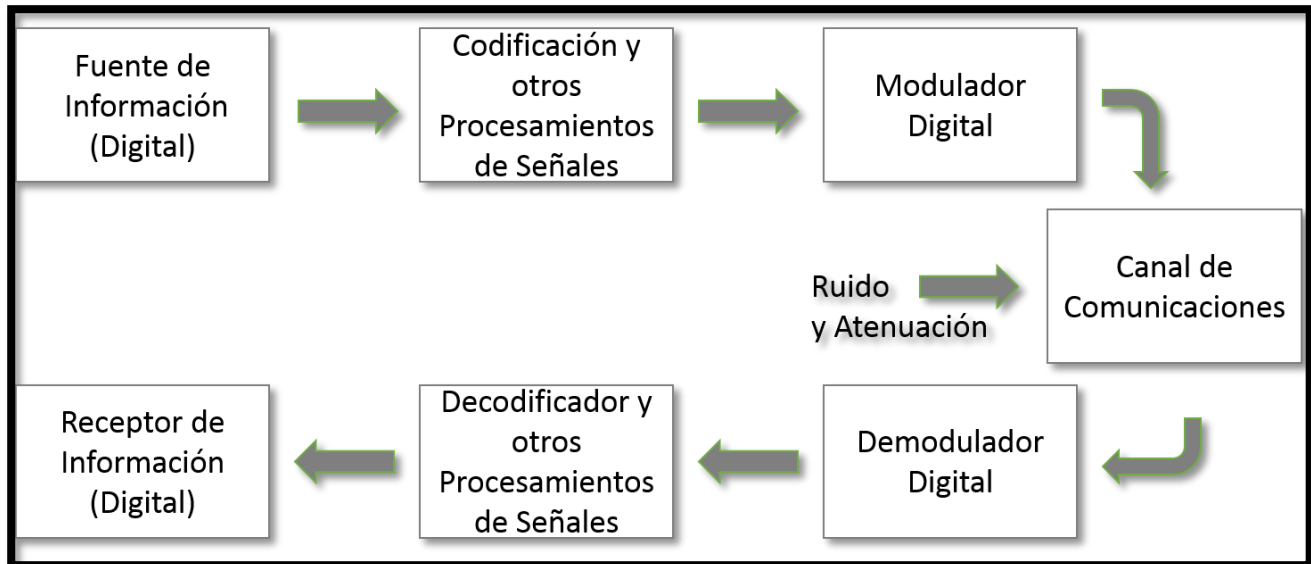


Figura 2. Diagrama a bloques general de un sistema de comunicación digital (Couch II, 1998).

2.1.1.1 Fuente de Información

La fuente de información puede ser cualquier tipo de dato que se quiera transmitir, incluso datos analógicos con su correcta conversión a datos digitales mediante un Convertidor Analógico-Digital (ADC). Podemos encontrar ejemplos de fuentes de información en la voz, imágenes, video, texto plano, etc.

2.1.1.2 Codificador y otros Procesamientos de Señales

Como ya se vio anteriormente, no es factible propagar la señal de información tal cual como llega en la entrada del transmisor, sino que es necesario realizar ciertos procesamientos de la señal previos a la transmisión por el canal de comunicaciones. Algunos de estos procesamientos incluyen: la codificación de fuente y la codificación de canal.

La codificación de fuente se utiliza cuando la aplicación requiere proporcionar un formato específico para los datos a transmitir con el objetivo de lograr una transmisión más eficiente. El codificador toma la señal binaria de entrada y la transforma en símbolos de un código binario más adecuado para su transmisión

y que además sea decodificable en forma única. En este punto se puede encontrar los protocolos para la compresión de archivos como JPEG, MPEG, MP3, etc.

En la codificación de canal se parte del hecho de que los sistemas de comunicaciones digitales tienen errores que ocurren, disminuyendo la eficiencia en la transmisión de datos. Estos errores pueden reducirse usando algún esquema de codificación para la detección y/o corrección de errores. Las dos técnicas principales son: la Requisición de Repetición Automática (ARQ, Automatic Repeat reQuest) y la Corrección de Errores Directa (FEC, Forward Error Correction).

En los sistemas ARQ, cuando el receptor detecta algún error de paridad en un mensaje, el receptor envía una solicitud para que el mensaje sea retransmitido, mientras que en un sistema FEC, al mensaje se le agregan bits codificados para que el receptor sea capaz de detectar y corregir los errores. La decisión de utilizar alguna de estas dos técnicas, depende de la aplicación en particular del sistema de comunicaciones. Los sistemas ARQ se usan a menudo en comunicaciones dúplex, como en las redes de computadoras donde es posible la implementación de acuses de recibo (ACK) por parte del receptor. Por otra parte, los sistemas FEC se utilizan en comunicaciones simplex donde la retransmisión de los mensajes no es factible y la eficiencia en la transmisión debe ser mayor.

2.1.2 Modulación.

La eficiencia y capacidad de un sistema de comunicaciones, depende en gran medida de la modulación, tan es así que el tipo de modulación a utilizar es una decisión fundamental para el diseño de un sistema de comunicaciones, por esta razón muchas técnicas de modulación han evolucionado. A pesar de la multitud de variedades, es posible identificar dos tipos básicos de modulación en relación a la clase de onda portadora: la modulación de onda continua, en la cual la portadora es

simplemente una forma de onda senoidal, y la modulación por pulsos, en la cual la portadora es un tren periódico de pulsos (Bruce, Crilly & Rutledge, 2002).

2.1.2.1 Razones de porque se modula

Las principales razones por las que se realiza la modulación en un sistema de comunicaciones son: Por facilidad de radiación, por la asignación de frecuencias para la transmisión y para la multiplexación.

Modulación por facilidad de radiación. Para una radiación de energía electromagnética, es necesario el uso de antenas que permitan la propagación de las señales por el espacio libre, pero muchas señales (en banda base) como las señales de voz, tienen componentes en el orden de los 100 Hz, para lo cual, si se quisiera irradiar estas señales, se necesitarían antenas de unos 300 Km de longitud aproximadamente. Si se utiliza un esquema de modulación, se puede trasladar la señal a una con mayor frecuencia, es decir, se utiliza una señal portadora de alta frecuencia para imprimir la señal que se quiere transmitir, con lo que se logra una reducción del tamaño de la antena.

Modulación por asignación de frecuencia. Para que varias señales independientes viajen por el mismo medio de transmisión sin causar interferencia, es necesario que cada una de estas señales tenga asignada una portadora con una frecuencia diferente a las demás.

Modulación por multiplexación: Cuando se requiere transmitir muchas señales en forma simultánea entre dos puntos. Las técnicas de multiplexación son formas de modulación, las cuales permiten la transmisión de múltiples señales sobre un canal de comunicaciones. De esta manera, cada una de las señales pueden ser captadas de manera correcta por el extremo receptor.

2.1.3 Capacidad de Información

La capacidad de información de un sistema de comunicaciones se representa mediante la cantidad de símbolos que pueden transmitirse por el sistema, en un tiempo determinado. Esta capacidad de información es expresada comúnmente en bits por segundo (bps). Claude E. Shannon relacionó la capacidad de información, con el ancho de banda y la relación señal ruido de un canal de comunicaciones, a esta fórmula se le conoce como límite de Shannon y se expresa de la siguiente forma:

$$I = B \log_2 \left(1 + \frac{S}{N} \right)$$

En donde: I = Capacidad de información (bps)

B = Ancho de banda (Hz)

$\frac{S}{N}$ = Relación de potencia de señal a ruido (adimensional)

De esta manera, Shannon proporciona un límite teórico que puede aproximarse en los sistemas de comunicaciones prácticos, dependiendo de varios factores como el esquema de modulación utilizado y el tipo de código para la detección de errores. Otro límite fundamental para la transmisión digital fue dado por Harry Nyquist, quien demostró que si un pulso representa un bit de datos, se puede enviar pulsos a través de un canal a una velocidad no mayor a $2B$ pulsos/s, donde B es el ancho de banda del canal en Hertz, en otras palabras la velocidad de transmisión de datos que se puede conseguir con un ancho de banda de B (Hz) es de $2B$ (bps).

2.2 Lenguajes de Descripción de Hardware

En la actualidad una de las herramientas más poderosas con las que se cuenta para el diseño de sistemas digitales son los Lenguajes Descriptivos de Hardware que permiten realizar una estructura jerárquica de las etapas del diseño logrando una optimización en el trabajo.

El lenguaje descriptivo Verilog-HDL describe un sistema embebido como un set de módulos. Cada uno de estos módulos tiene una interfaz a otros módulos. Un módulo representa una unidad lógica que puede ser descrita ya sea por su estructura lógica interna o describiendo su comportamiento a manera de programa. Verilog-HDL es reconocido como el lenguaje descriptivo de estándar europeo.

VHDL (Very High-speed Integrated Circuit Hardware Description Language) es el lenguaje descriptivo más usado en la actualidad, su popularidad se debe principalmente a que es un estándar del IEEE (Institute of Electrical and Electronics Engineers). El modelado de circuitos en VHDL se basa principalmente en la definición de arquitecturas que representan cada una de las estructuras digitales que pueden aplicarse al diseño del circuito. A partir de cada una de estas estructuras, se pueden declarar entidades con sus correspondientes elementos para llevar acabo la implementación del circuito y de su simulación (Sánchez, García, Saldívar, Montiel, Sepúlveda & Charles, 2006).

Los lenguajes descriptivos sirven como una herramienta para obtener soluciones a problemas de diseño de manera rápida y eficiente. Se han desarrollado componentes de Hardware que permiten la programación mediante estos lenguajes descriptivos, estos componentes van desde pequeños dispositivos con unas cuantas compuertas lógicas, hasta dispositivos más complejos con una capacidad mayor de integración de miles de compuertas agrupadas en bloques de funcionalidad (Aceves & Ramos, 2012).

2.3 La FPGA

La evolución de los sistemas de comunicaciones ha generado cambios significativos en los diseños e implementaciones, reducción del factor forma y de consumo de potencia. Los dispositivos lógicos programables (PLD's) ofrecen características que satisfacen estas exigencias como facilidad para programar alta velocidad de operación, minimización del número de dispositivos de conexión y el procesamiento en paralelo.

Los dispositivos lógicos programables se pueden dividir de acuerdo a su funcionalidad como se muestra en la figura 3.

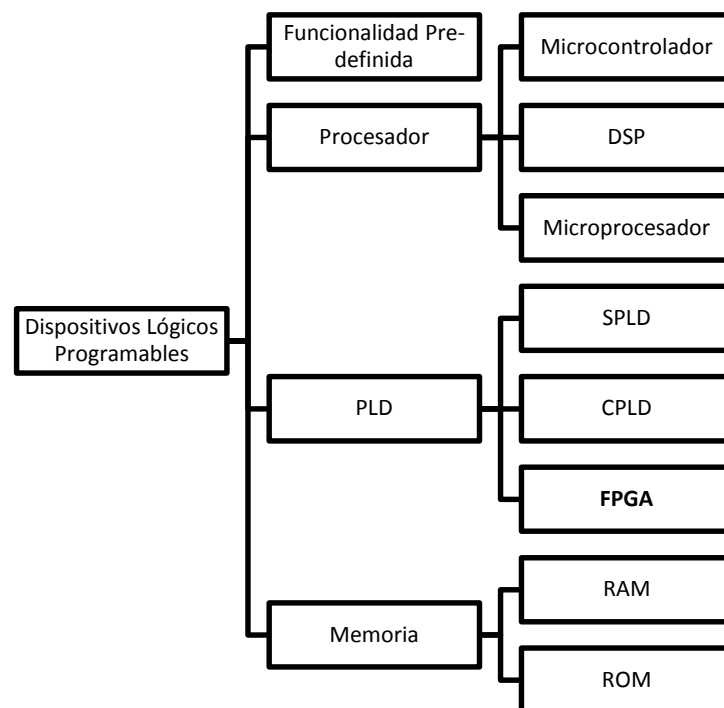


Figura 3. Categorías de los dispositivos lógicos programables (Aceves y Ramos, 2012).

Los PLD's son muy usados en la actualidad debido a su bajo costo, su flexibilidad y facilidad de implementación. Estos dispositivos contienen bloques de lógica digital que permiten conectarse entre sí, de tal manera que se pueden configurar los bloques de acuerdo a la funcionalidad requerida. Los tres tipos de PLD's que existen son: SPLD's, CPLD's y FPGA's (García y Pérez, 2006).

Las FPGA's (Field-Programmable Gate Arrays) son dispositivos que contienen bloques de lógica digital que se pueden programar. Mediante estos dispositivos se pueden implementar sistemas completos en un solo chip (Maxinez y Jara, 2002). Su arquitectura se basa en arreglos de bloques lógicos como se muestra en la figura 4.

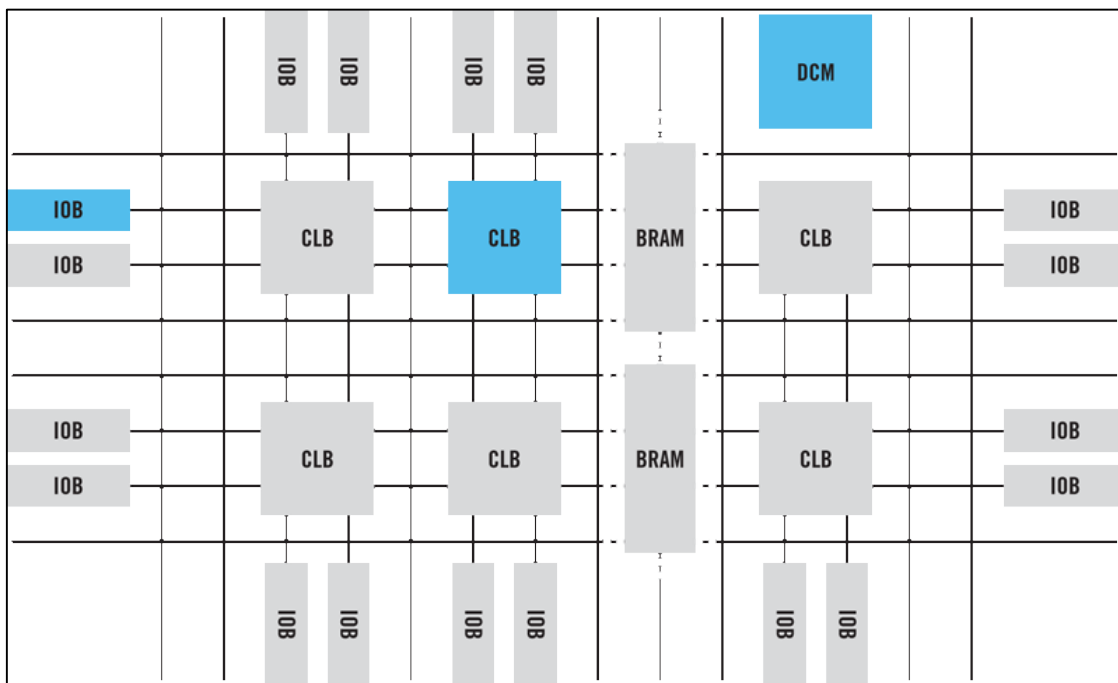


Figura 4. Estructura de los bloques de la FPGA (Xilinx Inc., 2011).

En la figura 4 se pueden observar los tres elementos básicos en una FPGA; el bloque lógico Configurable (CLB), las interconexiones y los bloques de entrada/salida (I/O).

El CLB es la unidad lógica básica en una FPGA, su número y características varían dependiendo el dispositivo. En general los CLB's consisten en cuatro segmentos, cada segmento está conformado por dos Look Up Table (LUT) para la implementación de lógica y dos más dedicadas al almacenamiento que puede ser usado como flip-flops, cada una de las LUT cuenta con cuatro entradas y una salida; también los CLB's cuentan con multiplexores y componentes de acarreo y aritmética lógicos. Cada CLB puede configurarse para implementar lógica combinacional y registros de almacenamiento. Estos CLB's se unen para crear bloques lógicos más complejos. La figura 5 muestra un esquema de los CLB's correspondientes a la familia de FPGA's Spartan-3E.

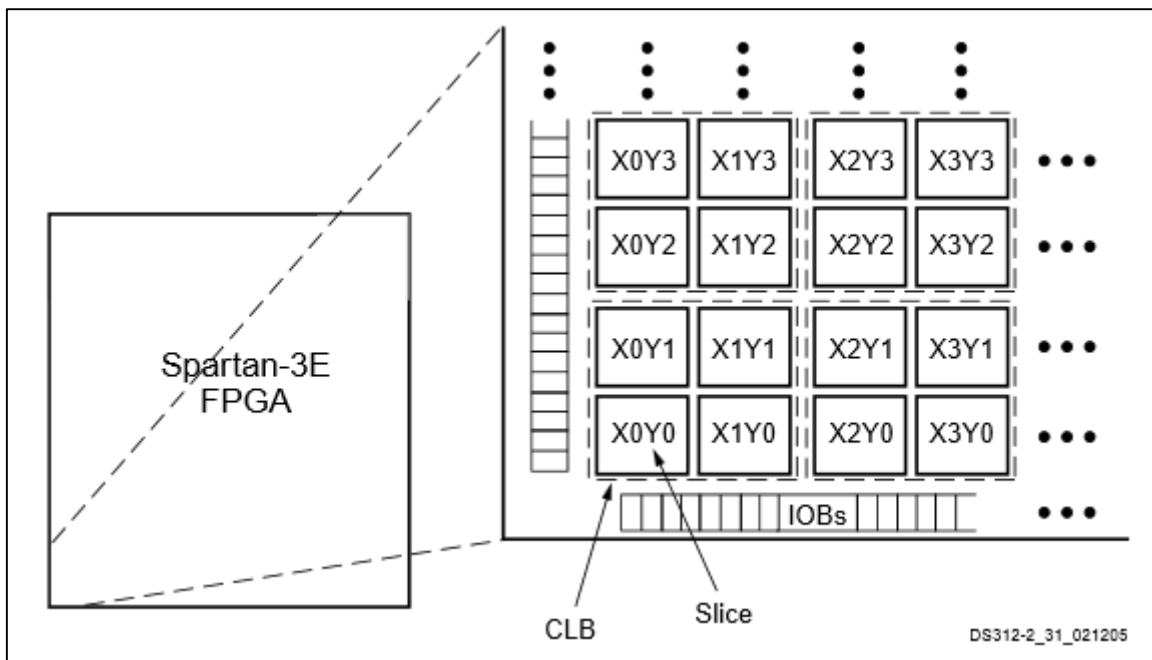


Figura 5. Estructura de los CLB's (Xilinx Inc., 2011).

Las LUT son bloques de RAM basados en generadores de funciones y es el principal recurso para la implementación de funciones lógicas. Una LUT con cuatro entradas y una salida es capaz de generar cualquier función booleana de cuatro variables. Para la implementación de funciones de más variables, se puede usar LUT's conectadas en cascada. La figura 6 muestra como la salida en una LUT se puede conectar al multiplexor, al acarreo y aritmética lógica, a un componente de almacenamiento o directamente a la salida del CLB.

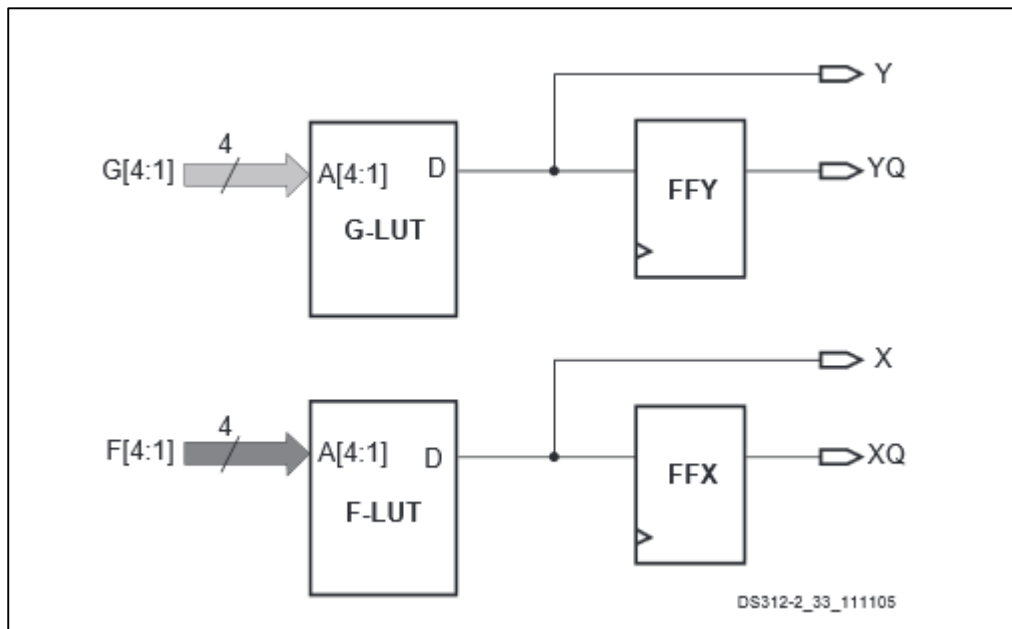


Figura 6. Estructura de las LUT dentro de un CLB (Xilinx Inc., 2011).

Los multiplexores son utilizados para combinar las LUT con la finalidad de crear operaciones lógicas más complejas. Esto sirve para multiplexar las dos LUT dentro de un mismo segmento del CLB, multiplexar las entradas de o los segmentos entre sí del CLB. Los componentes de acarreo y aritmética lógica son utilizados para generar operaciones matemáticas de manera rápida y eficiente, usando compuertas lógicas dedicadas (Xilinx, 2011)

Las FPGA's tienen varias características que las hacen eficientes y flexibles para el diseño de sistemas digitales completos (Aceves y Ramos, 2012):

- Reprogramables y reconfigurables. Se pueden reprogramar y reconfigurar de acuerdo a las necesidades del sistema.
- Procesamiento en Paralelo
- Bajo costo de desarrollo de aplicaciones.
- Reducción en tiempos de diseño
- Desarrollos 100% utilizables
- Portabilidad de las aplicaciones

2.4 Descripción del Software y Hardware

2.4.1 Matlab-Simulink

Matlab, cuyo significado es Laboratorio de Matrices, es un conjunto de herramientas matemáticas que hacen de éste, un software muy poderoso en el ámbito de la ingeniería y la ciencia ya que posee una gran variedad de funciones matemáticas implementadas. Algunas de las herramientas que contiene este software son utilizadas para aplicaciones como: Procesamiento de Imágenes, Desarrollo de Algoritmos, Control Automático, Telecomunicaciones, Robótica, entre otras; además permite la integración con otras herramientas externas tanto de hardware como de software.

Simulink es una herramienta incorporada en Matlab para el modelado y simulación mediante un método gráfico de diagrama de bloques con el cual se puede evaluar el comportamiento, funcionamiento y eficiencia de sistemas complejos. Las simulaciones en Simulink se conforman mediante diagramas de bloques basados en tiempo que generan gráficos del flujo de datos en las salidas y los estados de éstas salidas son simuladas en cada paso.

Matlab y Simulink juegan un papel muy importante en el diseño industrial, ya que permiten un desarrollo de soluciones de diseño y facilidades para la generación de código, permitiendo un diseño y modelado heterogéneo mediante la integración de módulos con diferentes niveles funcionales de abstracción, para una validación e implementación rápida de prototipos (Müller, Schwannecke & Fengler, 2012).

2.4.1.1 Xilinx System Generator

Xilinx System Generator (XSG) es un conjunto de bloques desarrollado por Xilinx Inc. para el uso en Simulink con la finalidad de trabajar con aplicaciones de procesamiento digital de señales (DSP, Digital Signal Processor) complejas, por lo que se basa en la misma metodología de Simulink para el desarrollo de modelos. Ésta herramienta utiliza el concepto de cajas negras (Black Box) y de abstracción de hardware para llevar acabo diseño de bloques funcionales y de esta manera lograr una integración de alto nivel para poder realizar simulaciones utilizando bloques ya existentes en Simulink. Los modelos creados con los bloques de XSG pueden generar automáticamente el código en VHDL y sintetizar el diseño hasta la creación del archivo de configuración (.bit) para la programación de las FPGA. En la figura 7 se puede observar la librería de Simulink con el conjunto de bloques de XSG.

Debido a que los bloques de Simulink usan tipos de variables propias de Matlab como dobles, vectores o matrices, y no pueden ser mapeados a una lógica binaria, es necesario una caja negra donde se programe el código VHDL necesario para el bloque requerido y de esta manera poder sintetizar el modelo completo (Maximiliano, Guillermo, Alejandro & Rodolfo, 2011).

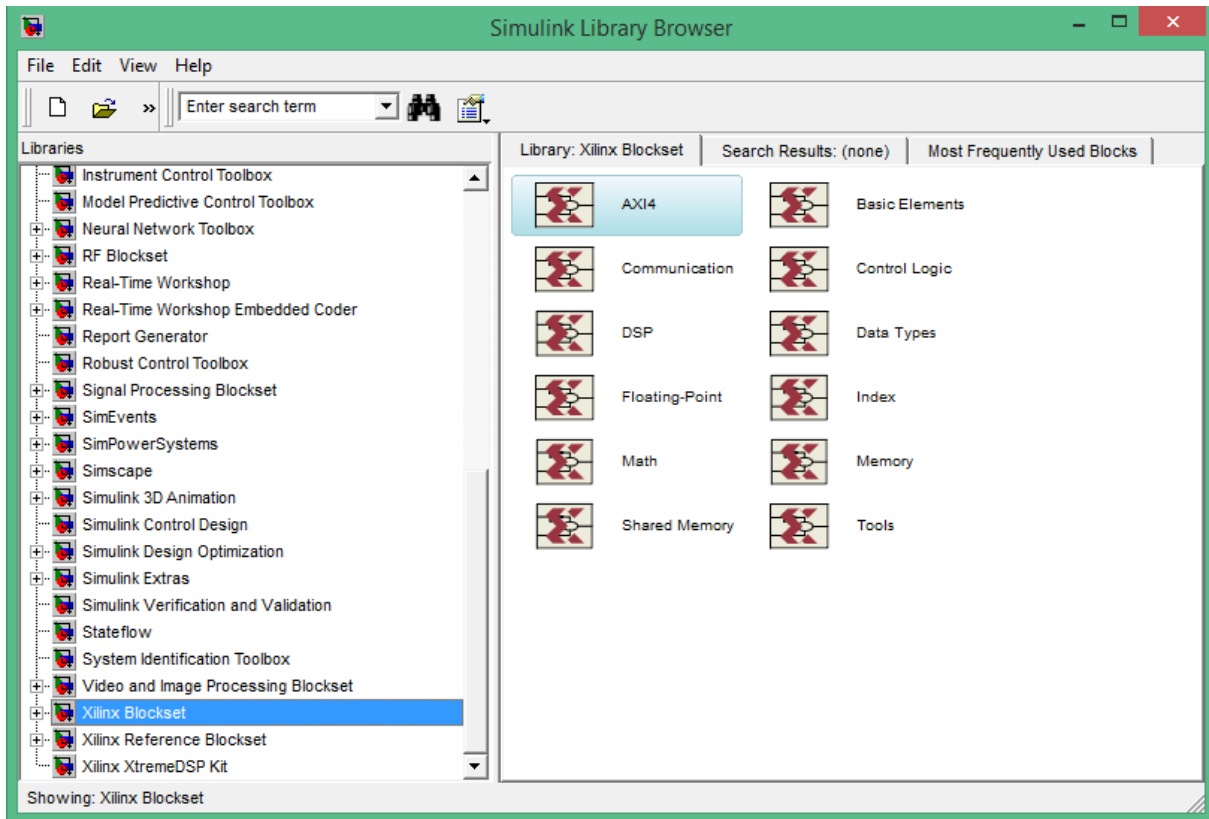


Figura 7. Librerías de Simulink y bloques de XSG (Xilinx Inc., 2011).

Los bloques fundamentales para el uso de un modelo con XSG son: el bloque *System Generator*, *Gateway In* y *Gateway Out*; estos bloques se observan en la figura 8. El bloque de *System Generator* sirve para especificar las características de la FPGA a utilizar como el tipo de chip y la velocidad del reloj. El bloque de *Gateway In* sirve como una interfaz de entrada para los bloques de Simulink que no pueden ser sintetizados, por lo que este bloque hace una conversión de tipo de dato apropiado para la sintetización. El bloque de *Gateway Out* funciona como interfaz para los bloques de visualización de Simulink y la salida de los bloques de XSG.

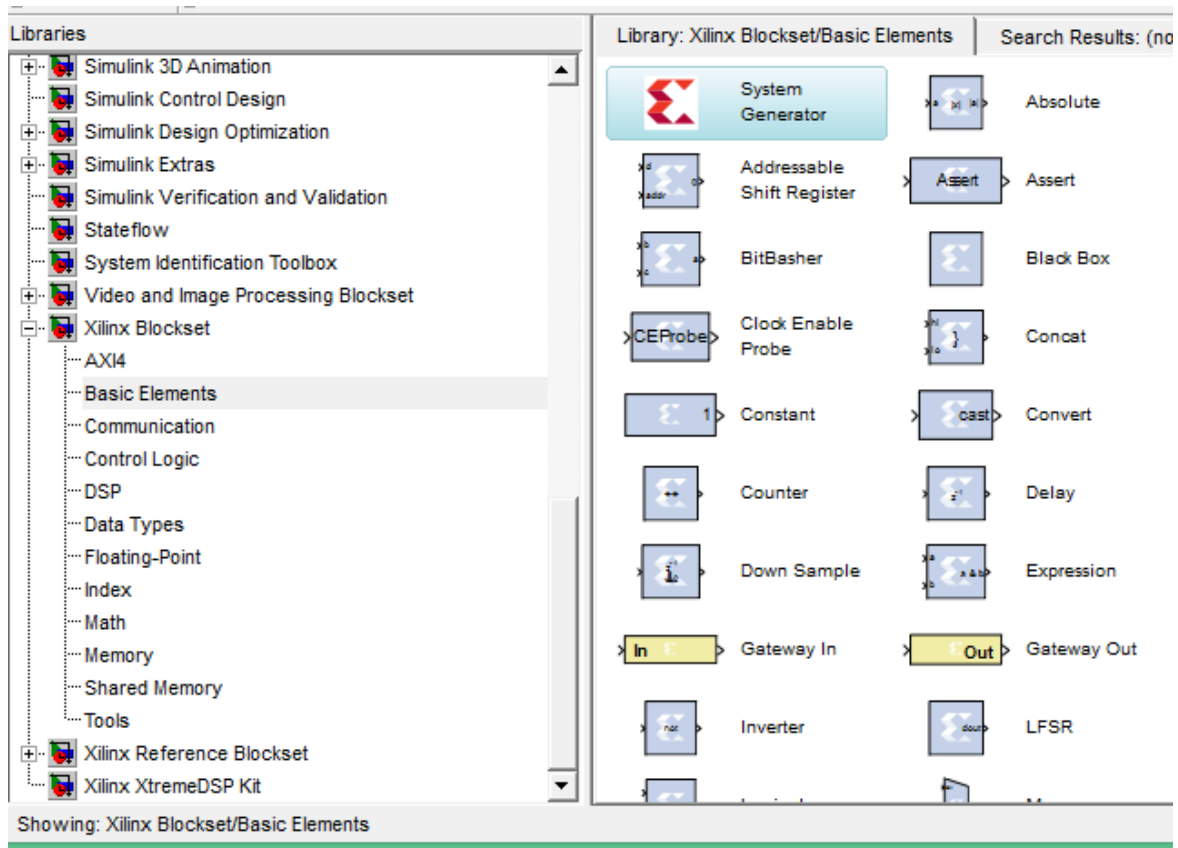


Figura 8. Bloques fundamentales de XSG (Xilinx Inc., 2011).

El Bloque de *Back Box* permite la integración de código VHDL dentro del entorno de Simulink mediante la creación de un bloque basado en código VHDL previamente programado. Este bloque se genera con referencia a los puertos declarados en el código VHDL (Kumar, Akhter & Gupta, 2012).

El Procedimiento para el uso de una Black Box se describe a continuación.

1. Iniciar Matlab y Abrir Simulink.
2. Crear un nuevo modelo y salvarlo con la correspondiente extensión .mdl en el mismo directorio que contiene los archivos VHDL. El nombre del modelo no debe ser igual a alguno de los archivos VHDL.
3. Del conjunto de bloques de XSG agregar al modelo el bloque de System Generator y modificar las propiedades del bloque de acuerdo al tipo de tarjeta de desarrollo que se quiera utilizar.
4. Colocar en el modelo el bloque de Black Box del conjunto de bloques básicos de XSG y en la ventana de configuración del bloque, seleccionar el archivo VHDL de más alto nivel y en el apartado de simulador, seleccionar ISE Co-Simulator.
5. Después de modificar las propiedades del bloque de Black Box, Matlab autogenerara un archivo .m con código que contiene las descripciones necesarias para la ejecución del código VHDL involucrado.
6. En este punto, al salvar el modelo, Matlab generara las entradas y salidas del archivo VHDL seleccionado en el bloque de Black Box.

2.4.2 Active-HDL

El modelado de circuitos digitales en VHDL consta básicamente de cinco pasos fundamentales para su desarrollo como se puede observar en el diagrama correspondiente a la figura 9.

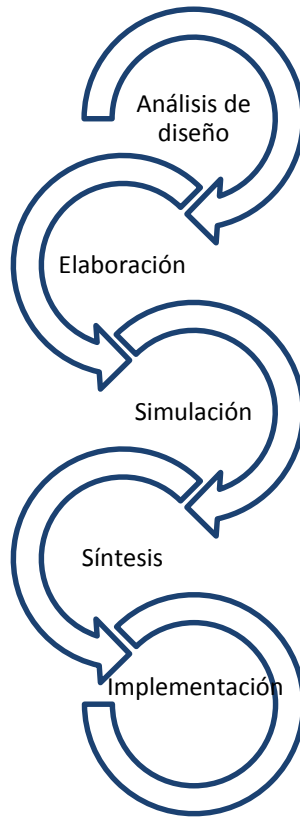


Figura 9. Metodología para el desarrollo de circuitos digitales.

El análisis de diseño es la fase inicial y consiste en el modelado a partir de los requerimientos, características del sistema y del circuito a implementar; la elaboración se realiza una vez que se han detectado y corregido los errores en el diseño, entonces se continua con la especificación del diseño en VHDL, al término de la fase de elaboración se proceder a ejecutar la simulación del diseño mediante la configuración de las señales de entrada y los tiempos de la simulación. En esta parte es recomendable la creación de bancos de pruebas para una evaluación óptima de la simulación.

La síntesis del sistema se realiza mediante la aplicación de una herramienta de síntesis, con la cual es posible convertir al diseño de abstracción de bajo nivel. Una vez que se realiza la síntesis de manera correcta, es factible la implementación del sistema en una tarjeta de desarrollo para realizar las pruebas pertinentes.

Active-HDL es un software basado en el lenguaje VHDL que permite el diseño, desarrollo y simulación de sistemas digitales en un ambiente completo. La figura 10 muestra la interfaz gráfica de Active HDL, que se compone básicamente de un explorador de diseño, un editor de HDL y una consola.

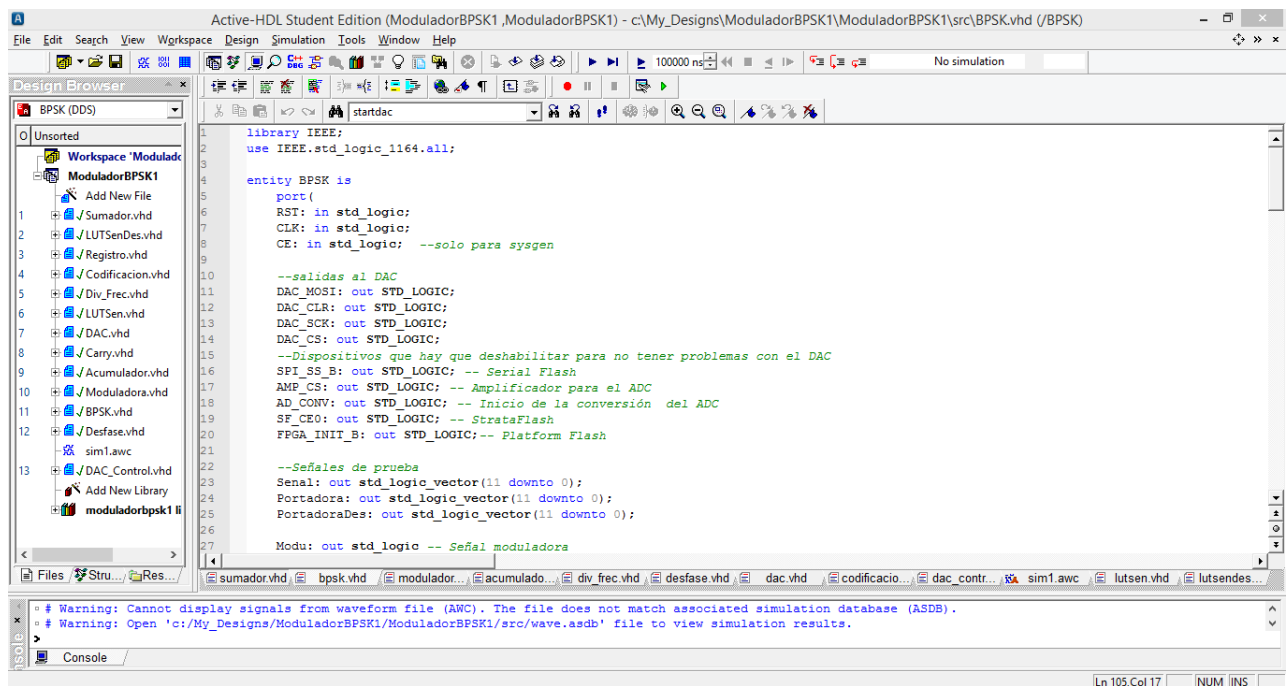


Figura 10. Interfaz de Active-HDL.

En el explorador de diseño, es posible ver el contenido de todas las entidades, también se puede agregar archivos VHDL de manera automática. Dentro del editor de HDL se desarrolla el código del diseño, y en la consola se puede visualizar un resumen de cada compilación y simulación, esto incluye las advertencias y/o errores generados durante la compilación del código.

La simulación en Active HDL, se hace mediante la herramienta de Visor de formas de onda (Waveform Viewer), esta herramienta sirve para generar un despliegue de los resultados mediante una simulación gráfica en forma de ondas, como se puede observar en la figura 11. Ésta simulación se lleva a cabo mediante la generación de un archivo (.awf) en donde se agregan las señales que se observaran en la simulación. A estas señales se les puede asignar valores iniciales. Posteriormente se le asigna el tiempo a la simulación y se procede a su ejecución (Sánchez, García, Saldívar, Montiel, Sepúlveda & Charles, 2006).

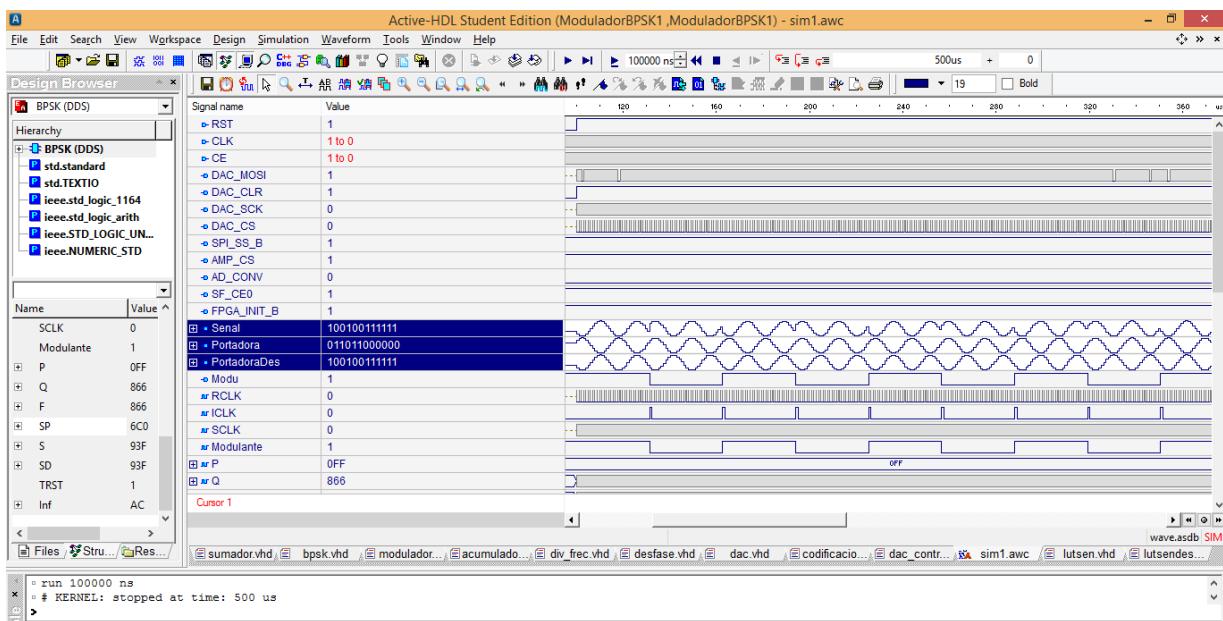


Figura 11. Visor de formas de onda de Active-HDL.

2.4.3 Xilinx ISE

ISE en su versión WebPack es un software gratis ideal para la implementación de diseños HDL en FPGA's y CPLD's, soportado por varios sistemas operativos. ISE ofrece herramientas para la síntesis, simulación e implementación de código VHDL además contiene un conjunto de controladores para la programación de tarjetas de desarrollo de Xilinx. La ventana principal de un proyecto en ISE se muestra en la figura 12.

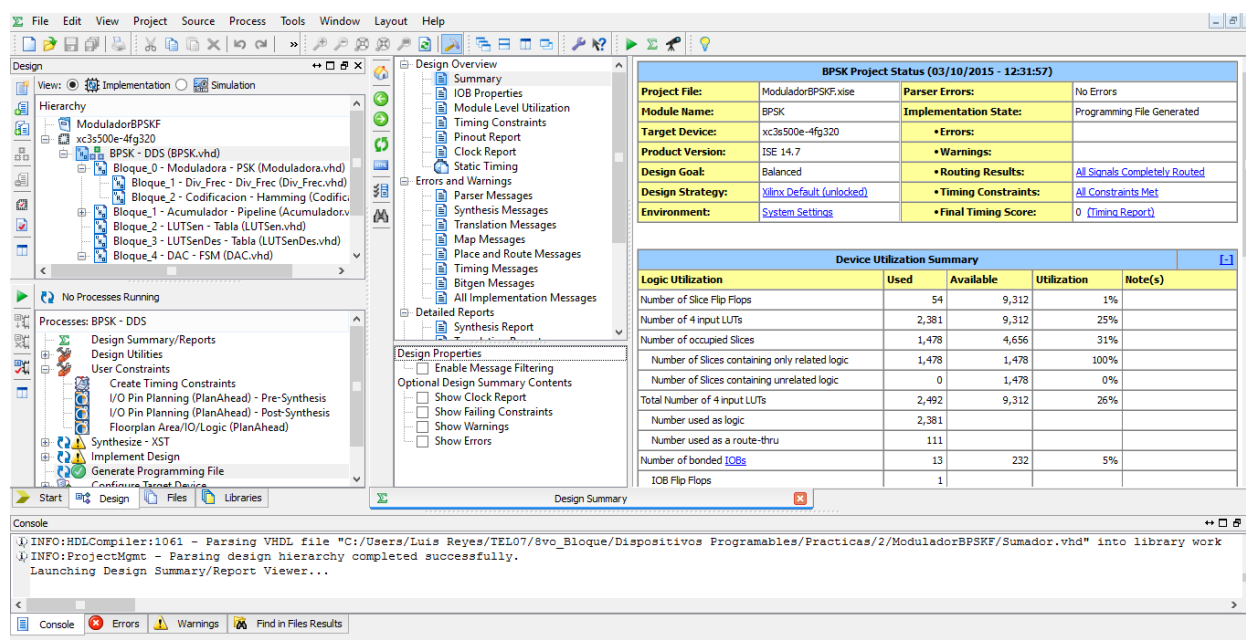


Figura 12. Interfaz de Xilinx ISE.

Las principales herramientas de ISE son: el editor de textos para la generación de código VHDL, el cual es muy similar al editor de Active-HDL; PlanAhead utilizado para el mapeo de las entradas y salidas del diseño con los puertos correspondientes a la tarjeta de desarrollo utilizada de arquitectura Xilinx, donde se especifica el puerto a utilizar, si se refiere a un puerto de entrada, salida o ambos y el nivel de voltaje utilizado. La interfaz de PlanAhead se muestra en la figura 13.

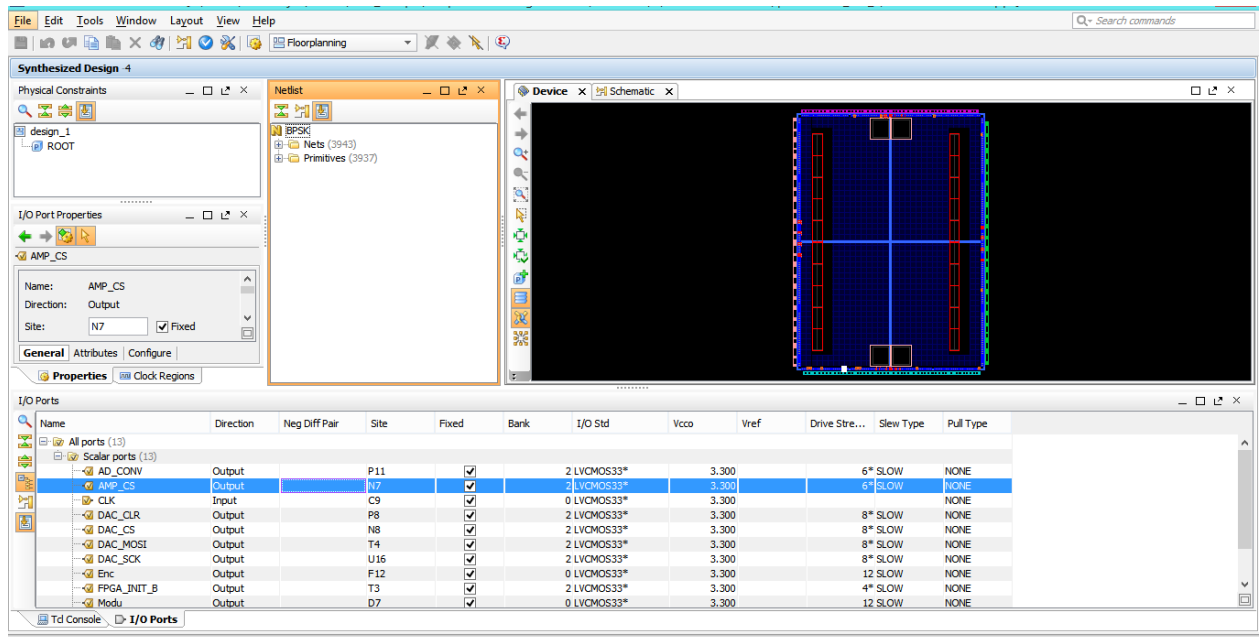


Figura 13. Interfaz de PlanAhead.

iMPACT es otra herramienta indispensable para la implementación de circuitos digitales en hardware como las tarjetas de desarrollo de Xilinx. Una vez instalados los controladores propios de la tarjeta de desarrollo a utilizar, iMPACT escanea el dispositivo y genera una cadena con los elementos programables de la tarjeta, esto permite seleccionar el elemento a programar y el archivo que se desea cargar, siendo este el último paso de acuerdo con la metodología de la figura 9. En la figura 14 se muestra la interfaz gráfica de la herramienta iMPACT.

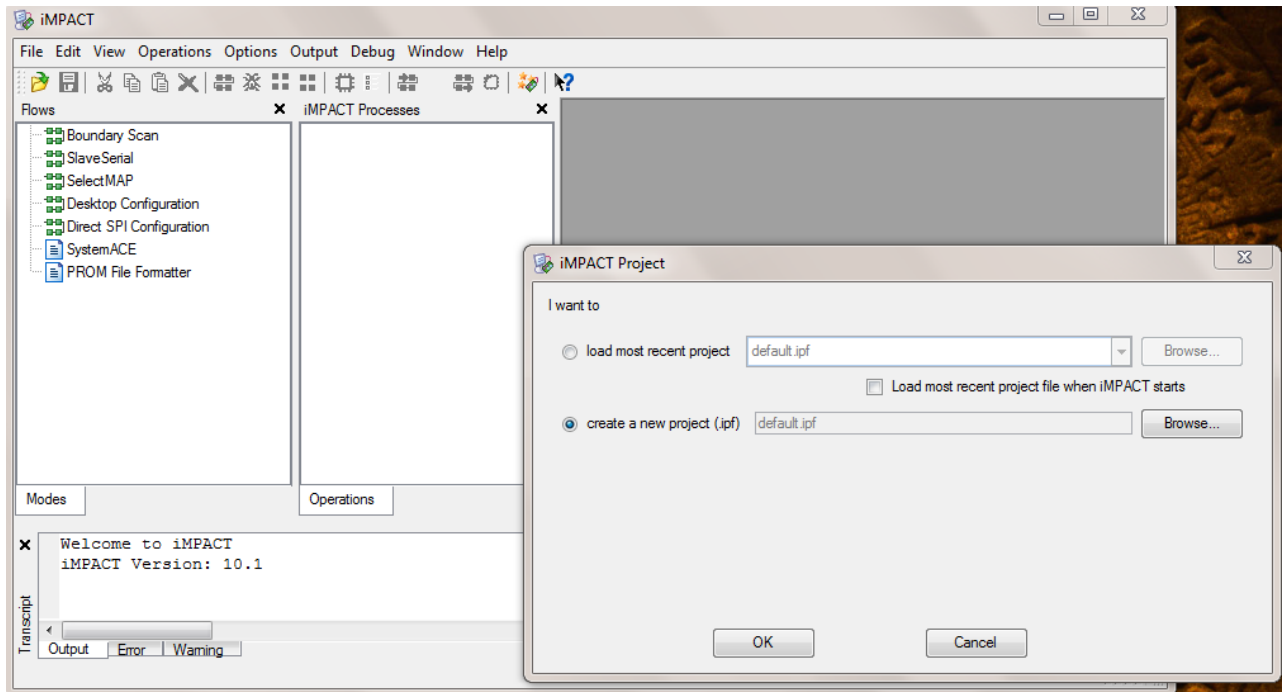


Figura 14. Interfaz de iMPACT.

2.4.4 Tarjeta de Desarrollo Spartan-3E Starter Kit

La tarjeta de desarrollo Spartan-3E Starter Kit pertenece a la familia de tarjetas Spartan de Xilinx. Esta tarjeta contiene 500 mil compuertas y un oscilador interno de 50MHz. En la figura 15 se muestra la tarjeta de desarrollo.

Las Características principales de hardware de la spartan-3E Starter Kit se enlistan a continuación (Xilinx Inc., 2011):

- Chip XC3S500E de la familia Xilinx Spartan.
- Memoria Flash de 4 Mbit para la configuración.
- CPLD XC2C64, familia CoolRunner™.
- DDR SDRAM de 64 Mbytes.
- Memoria Flash de 16 Mbyte para uso de aplicaciones.

- Memoria Flash de 16Mbits para la comunicación serial.
- Pantalla LCD de 2 líneas.
- Puerto PS/2.
- Puerto VGA.
- Puerto Ethernet 10/100 Mbits.
- Dos puertos RS-232.
- Interfaz USB para la carga de programas.
- Tres conectores de expansión con el estándar de 6 pines de Xilinx
- Convertidor Digital-Analógico SPI de cuatro canales con resolución de 12 bits.
- Convertidor Analógico-Digital SPI de dos canales con resolución de 14 bits y un pre-amplificador con ganancia programable.

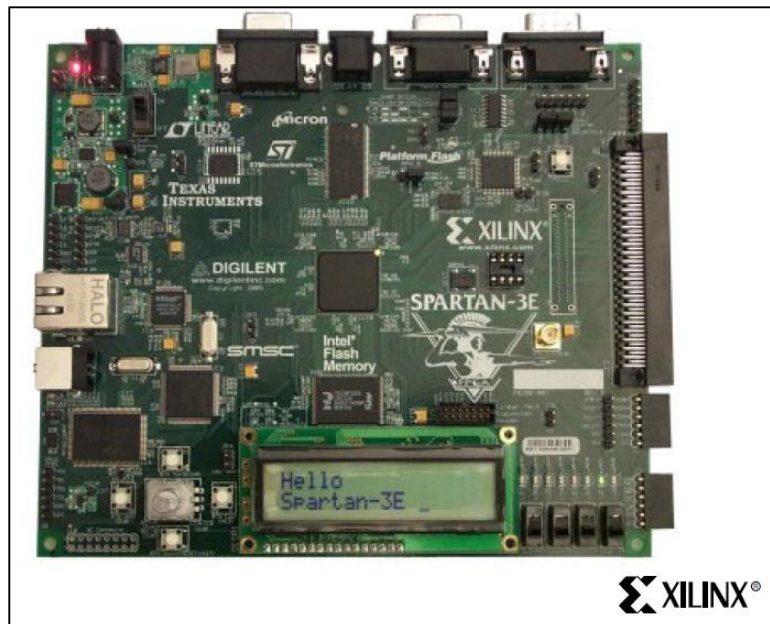


Figura 15. Tarjeta de desarrollo Spartan-3E Starter Kit (Xilinx Inc., 2011).

La tarjeta Spartan-3E Starter Kit puede funcionar con el oscilador interno de 50MHz, mediante un oscilador diferente conectado al puerto de 8 pines o mediante un oscilador externo. La ubicación de los distintos osciladores que puede tener la tarjeta se muestra en la figura 16.

Para la implementación del transmisor en la tarjeta Spartan-3E Starter Kit, los componentes a utilizar son: el oscilador interno de 50 MHz, el convertidor digital-analógico, un interruptor y un pin de entrada/salida. El uso de estos componentes se detalla en el Capítulo 4.

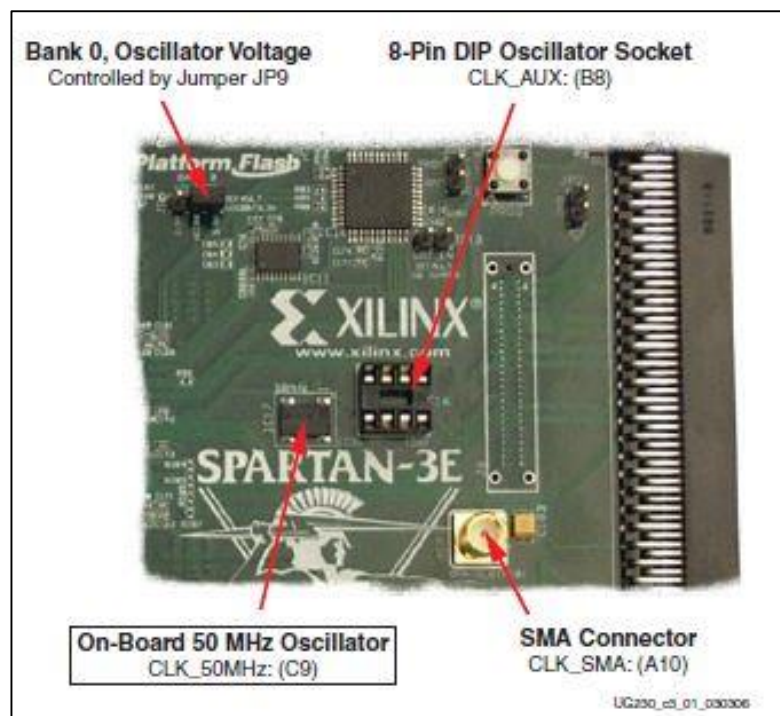


Figura 16. Ubicación y pin del oscilador interno de 50MHz de la Tarjeta Spartan-3E Starter Kit (Xilinx Inc., 2011).

2.4.4.1 DAC LTC2624

Un DAC es un circuito electrónico que permite el enlace entre señales digitales y analógicas mediante grupos de bits. El DAC devuelve una salida de forma analógica ya sea de voltaje o de corriente. El número de bits en la entrada del

convertidor es proporcional a la resolución de la salida. Las resoluciones más comunes en los DAC's son de 8 a 24 bits (Al Bustam & Shahzamal, 2013).

La tarjeta de desarrollo spartan-3E starter kit contiene un DAC de 4 canales con una interfaz de comunicación SPI (Serial Peripheral Interface). El modelo del DAC es un LTC2624 de Linear Technology con 12 bits de resolución. Este DAC se encuentra en el conector J5 de la tarjeta de desarrollo, que usa el estándar de 6 pins para periféricos de Xilinx, como se muestra en la figura 17.

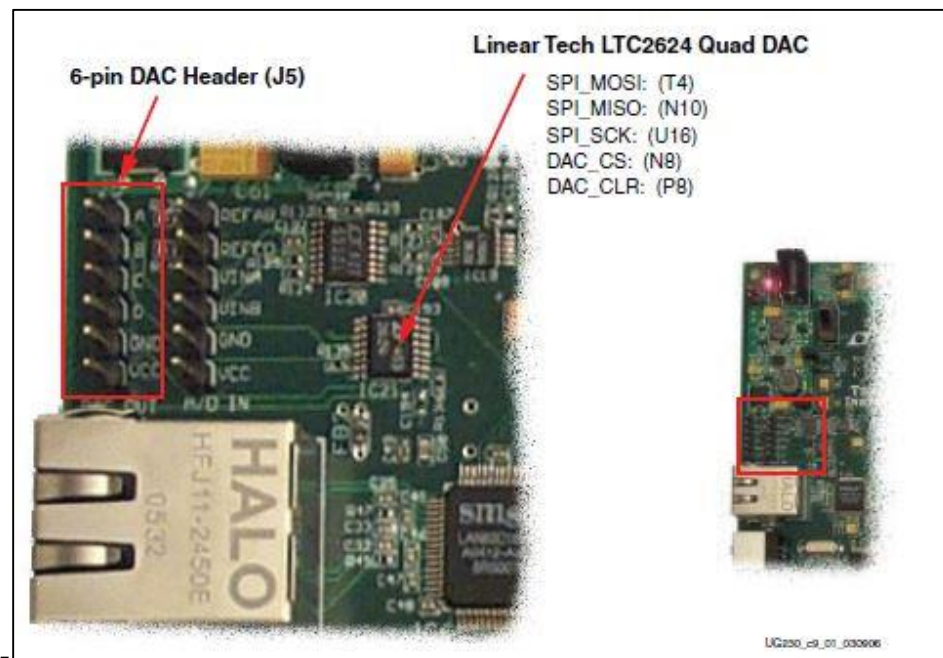


Figura 17. Ubicación del DAC y de los pines de salida para los 4 canales (Xilinx Inc., 2011).

La tarjeta de desarrollo usa una comunicación SPI para enviar los valores digitales a convertir a cada uno de los 4 canales del DAC. El bus SPI es full-duplex síncrono de cuatro señales. Las señales que interactúan con el DAC son: SPI_MOSI (Master Output, Slave Input), SPI_MISO (Master Input, Slave Output) y SPI_SLK (Reloj del sistema). La señal SPI_CS se activa en bajo para iniciar la comunicación con el DAC, es decir, funciona como el reset de la comunicación. La figura 18 muestra el esquema de conexión interno de la FPGA y el DAC LTC2624,

también se observan las señales que interactúan en la comunicación y los cuatro canales de DAC con su correspondiente salida en el conector J5 de la tarjeta de desarrollo.

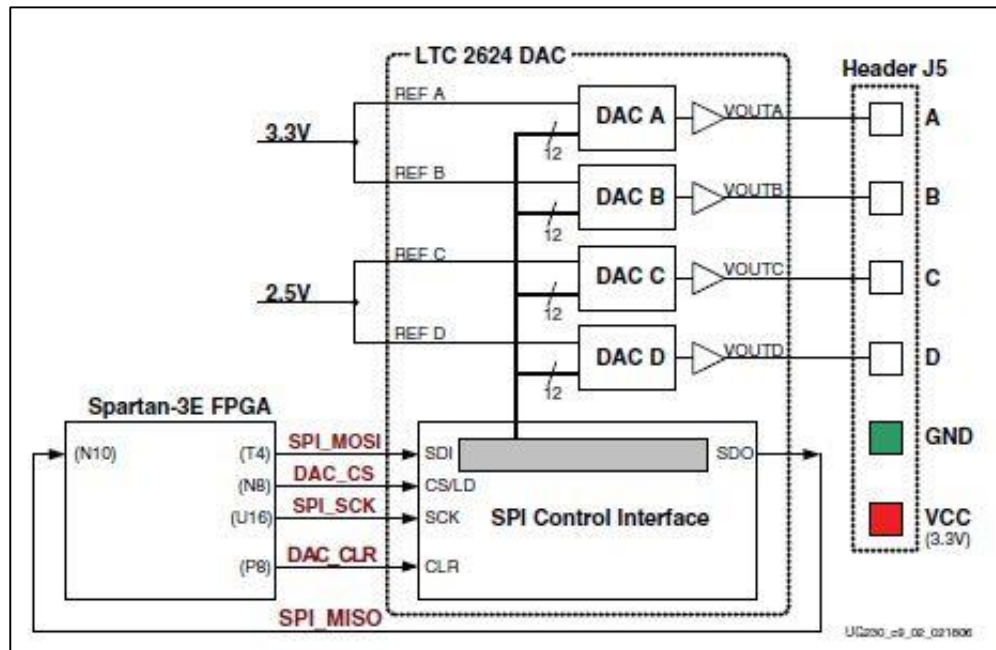


Figura 18. Esquema de Conexión de la Spartan-3E y el DAC LTC2624 (Xilinx Inc., 2011).

La figura 19 muestra la representación del protocolo de comunicación que utiliza el DAC LTC2624, este protocolo consta de 24 bits. Los primeros 4 bits del protocolo se usan para seleccionar el comando de escritura/actualización del convertidor como se puede observar en la tabla 1, el comando más utilizado es “0011” el cual actualiza inmediatamente la salida seleccionada en el DAC. Los siguientes 4 bits del protocolo son para la selección del canal del DAC a utilizar, las combinaciones para la selección del canal se puede observar en la tabla 2. Los 12 bits posteriores conforman el valor a convertir analógicamente y finalmente para completar el protocolo de 24 bits, se agregan 4 bits *don't care* (Linear Technology Corporation, 2004).

Tabla 1. Comandos del DAC LTC2624.

C3	C2	C1	C0	Acción
0	0	0	0	Escritura en el registro de n de entrada
0	0	0	1	Actualización del registro n del DAC
0	0	1	0	Escritura en el registro n de entrada y actualización de todos los registros del DAC
0	0	1	1	Escritura y actualización del registro n del DAC
1	1	1	1	Sin funcionamiento, Apagado del DAC

Tabla 2. Direcciones para la selección de canal en el DAC LTC2624.

A3	A2	A1	A0	Canal del DAC
0	0	0	0	Canal DAC A
0	0	0	1	Canal DAC B
0	0	1	0	Canal DAC C
0	0	1	1	Canal DAC D
1	1	1	1	Todos los canales

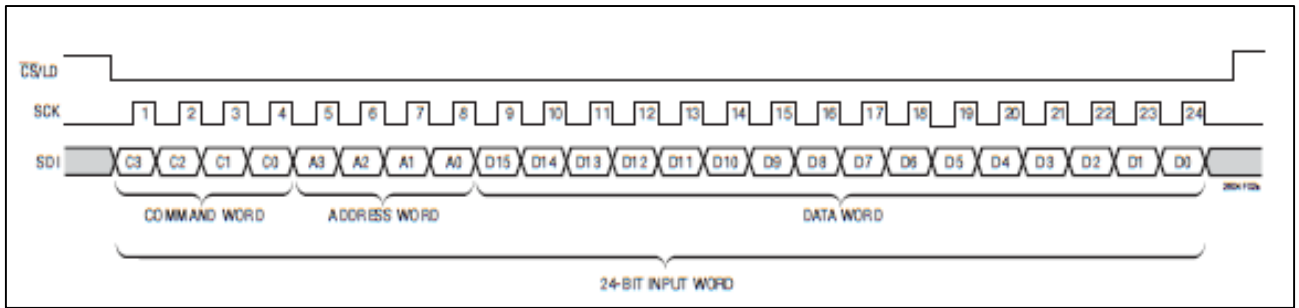


Figura 19. Protocolo de 24 bits de comunicación del DAC LTC2624 (Linear Technology Corporation, 2004).

Ya que el bus SPI se comparte con otros dispositivos como el convertidor, la memoria Flash y la PROM, es necesario deshabilitar dichos dispositivos para que no entren en conflicto con el DAC. La tabla 3 muestra los dispositivos y el valor que deben tomar para ser deshabilitados (Xilinx Inc., 2011).

Tabla 3. Dispositivos deshabilitados.

Señal	Dispositivo deshabilitado	Valor para deshabilitar
SPI_SS_B	SPI para la Memoria Flash	1
AMP_CS	Amplificador programable ADC	1
AD_CONV	Convertidor Analógico-Digital	0
SF_CE0	Flash PROM paralela	1
FPGA_INIT_B	Flash PROM	0

3. TRANSMISORES DIGITALES

3.1 Codificación en los transmisores digitales de datos

Para muchas aplicaciones los errores se pueden reducir sin alterarlos si solo se detectan sin intentar la corrección inmediata. Esto es adecuado en aplicaciones como la telemetría de datos, donde se reúne un gran número de valores para el análisis estadístico; en este análisis, si los valores erróneos se detectan, simplemente se descartan, y la pérdida es despreciable. De manera similar, en un enlace de comunicaciones de dos vías, por el hecho de que se haya detectado un error, se puede pedir una retransmisión mediante un protocolo establecido. Esta retroalimentación en un sistema de comunicaciones, brinda una ventaja grande si el sistema está sujeto a condiciones variables; cuando las condiciones de transmisión y recepción son buenas, resulta más efectivo un código de baja redundancia con una alta velocidad de datos; cuando las condiciones son malas (y se detectan una gran cantidad de errores), el transmisor puede cambiar a un código de redundancia alta y a su vez, bajar la velocidad de transmisión de datos pero aumentando la fiabilidad de la comunicación pese a las condiciones. La detección sencilla de error es funcional solo si la probabilidad de error \mathcal{E} es pequeña y la probabilidad de errores no detectados está en un adecuado bajo nivel.

La probabilidad de error (BER) en la transmisión digital es en forma equivalente a la S/N. Si para un sistema dado, se limita la potencia de la señal a algún valor máximo y los errores son, a pesar de la potencia de la señal, frecuentes de tal manera que no se establece una comunicación estable, entonces se debe buscar algún otro medio para mejorar la calidad de la transmisión.

En la teoría de la información (desarrollada por Claude Shannon y Warren Weaver) sugiere que se agreguen de forma sistemática dígitos extra al mensaje transmitido, dichos dígitos, no contienen información propia del mensaje a transmitir, sino que hacen posible que el receptor detecte o incluso, sea capaz de corregir los errores en los dígitos que llevan información. En teoría, es posible una transmisión cercana a aquella sin errores; de manera práctica, existe el inevitable dilema entre la fiabilidad de la transmisión, la eficiencia y la complejidad del equipo terminal. Considerando estos factores, se ha ideado una gran cantidad de códigos para la detección y corrección de errores para adecuarse a diferentes aplicaciones, dependiendo del sistema de comunicaciones a implementar.

Existen tres consideraciones generales en la codificación para la detección y corrección de errores: agregando los dígitos extra, a los que se les designa dígitos de comprobación o de redundancia, las palabras de código pueden ser muy diferentes entre sí. En forma analítica, la diferencia entre dos palabras binarias se mide en términos de la distancia de Hamming d , definida en forma sencilla como el número de lugares en los que las palabras tienen dígitos diferentes; así, toma d errores para cambiar de una palabra a otra. Un código que detecta o corrige K errores por palabra debe consistir en palabras de código que tengan:

$$d_{\min} = K + 1 \quad \text{Para la detección de errores}$$

$$d_{\min} = 2K + 1 \quad \text{Para la Corrección de errores.}$$

En términos generales, los códigos instrumentados con facilidad y que además son efectivos en el control de errores, requieren relativamente grande porcentaje de dígitos de comprobación. Así, el control práctico de errores tiende a ir de la mano con la reducción de velocidad de bits.

En la detección de errores se vigilan los datos recibidos y se determina si ha ocurrido un error de transmisión. Las técnicas de detección de errores no determinan cual o cuales bits están equivocados, solamente indican si hubo un error. El objetivo de la detección no es evitar que ocurran errores, sino evitar que haya errores sin detectar, posteriormente, la forma en que un sistema reacciona a los errores de transmisión varía mucho y depende de cada sistema en particular. Las técnicas más comunes para la detección de errores para los circuitos de comunicación de datos son: redundancia, codificación de cuenta exacta, paridad, suma de comprobación, comprobación de redundancia vertical, comprobación de redundancia horizontal y comprobación de redundancia cíclica.

Redundancia: Esto implica la retransmisión de un caracter dos veces, si este caracter no se recibe dos veces seguidas, hubo un error de transmisión. Si no se recibe la misma sucesión de caracteres dos veces seguidas, exactamente en el mismo orden, ha ocurrido un error de transmisión.

Codificación de cuenta exacta. Con esta codificación, la cantidad de unos en cada caracter es igual, un ejemplo de este esquema de codificación de cuenta exacta es el código ARQ, donde cada caracter tiene tres unos, y en consecuencia, si solo se cuenta la cantidad de unos recibida en cada caracter se puede determinar si ha ocurrido un error.

Paridad: La paridad es un esquema muy sencillo de detección de errores que se usa en sistemas de comunicación de datos. En la paridad se añade un solo bit, llamado bit de paridad, a cada caracter, de esta manera se obliga a que la cantidad total de unos en el mensaje, incluyendo el bit de paridad, sea un número par para la paridad par o un número impar para la paridad impar. Lo anterior se puede realizar mediante la operación binaria XOR (or exclusivo), en donde si las entradas son iguales, la salida es 0; y si las entradas son diferentes, la salida es 1. La ventaja de la paridad es su sencillez, la desventaja es que cuando se reciben varios bits erróneos, podría no detectarlos el comprobador de paridad, es decir, si cambian

la condición lógica de dos bits, la paridad sigue siendo la misma, en consecuencia, a la larga, la paridad solo detectara el 50% de los errores de transmisión.

Suma de comprobación: La suma de comprobación es una forma sencilla de detección de error. Mientras que se transmiten los datos, cada caracter se suma con el acumulado de la suma de los caracteres que se transmitieron antes, cuando se llega el final del mensaje, el sumador tendrá la suma de todos los caracteres que se transmitieron. El transmisor envía el byte menos significativo de esta suma, agregándola al final del mensaje. El receptor recibe el mensaje y suma todos los caracteres que contiene para comprobar el byte menos significativo de esta suma con el último byte del mensaje transmitido. Si estos bytes son diferentes, es seguro que hubo un error de transmisión.

La comprobación de redundancia vertical (VRC, Vertical Redundance Checking): es un esquema de detección de errores que usa la paridad para determinar si ha ocurrido un error de transmisión dentro de un caracter o palabra. En la VRC cada caracter tiene agregado un bit de paridad antes de la transmisión. Un ejemplo de este esquema de detección de errores se puede apreciar en el código ASCII.

La comprobación de redundancia horizontal (HRC, Horizontal Redundance Checking): también es un esquema de detección de errores que usa la paridad para determinar un error de transmisión, solo que no en un caracter, sino que en el mensaje completo. Con el HRC cada posición de bit tiene un bit de paridad, es decir, el bit menos significativo de cada caracter se compara mediante la operación XOR y así sucesivamente hasta llegar al bit más significativo de cada caracter. Esta secuencia de bits se transmite como si fuera el último caracter del mensaje. En el receptor, se vuelve a calcular el HRC y se compara con el último caracter del mensaje recibido. Si estos valores son iguales, se supone que no hubo error en la transmisión, de lo contrario, debe haberse presentado un error en la transmisión.

La Comprobación de redundancia cíclica. Es un esquema de detección de errores muy confiable (CRC, Cyclic Redundancy Checking): en este sistema se detectan aproximadamente el 99.95% de los errores de transmisión. Por lo general, se usa CRC con códigos de ocho bits, o de siete bits cuando no se usa paridad. En esencia, el carácter CRC es el residuo de una división. Se divide el polinomio $G(x)$ de mensaje de datos entre una función $P(x)$ generadora de polinomios, se desecha el cociente y se trunca el residuo a 16 bits, y se agrega al mensaje. Sin embargo, con la generación CRC la división no se hace con un proceso normal de división aritmética.

Los códigos convolucionales: conocidos también como códigos secuenciales o recurrentes, difieren de los códigos de bloque en que los dígitos de comprobación se intercalan en forma continua en la corriente codificada de bits en vez de agruparse en palabras. Por lo tanto el proceso de codificación/decodificación es un proceso continuo, eliminando los componentes de separación o almacenamiento que se requieren en los códigos de bloques (Wayne, 2003).

3.1.1 Códigos de Hamming

Hamming ha ofrecido un procedimiento para el diseño de códigos de bloque capaces de la corrección y detección de un error. Un Código de Hamming tiene una distancia de Hamming de 3, dado que $d \geq 2k + 1$ donde $k = 1$. Los códigos de Hamming tiene la forma de: $(n, k) = (2^m - 1, 2^m - 1 - m)$, donde m es un entero mayor o igual a 3, de manera que algunos de los códigos de Hamming permitidos son: (7,4), (15,11), (31,26) y (63,57), donde se puede observar que mientras más grande es el código, la velocidad de éste aumenta, aproximándose a 1 y por lo tanto el código se vuelve más eficiente (Couch II, 1998).

Lo que Hamming propuso es lo siguiente: Si hay q dígitos de comprobación por palabra, entonces el síndrome es una palabra de q dígitos que se puede hacer para descifrar en forma binaria y situar la posición exacta de un error sencillo. De esta manera, la matriz de comprobación de paridad se construye teniendo en cuenta de que s es igual a la columna j -ésima de H cuando hay un error único en el dígito j -ésimo del mensaje. Por lo tanto si se leen las columnas de la matriz H de izquierda a derecha, se puede observar que los dígitos son los números 1, 2, 3,..., n en binario, como se puede apreciar en la siguiente matriz H para un código de Hamming (7,4):

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Dado que las posiciones de los dígitos de comprobación deben corresponder a las columnas de H donde solo hay un 1, las palabras para un código Hamming (7,4) tiene la forma de:

$$x = [c_1 \ c_2 \ m_1 \ c_3 \ m_2 \ m_3 \ m_4]^T$$

Y las ecuaciones para los dígitos de comprobación son:

$$\begin{aligned} c_1 &= m_1 \oplus m_2 \oplus m_4 \\ c_2 &= m_1 \oplus m_3 \oplus m_4 \\ c_3 &= m_2 \oplus m_3 \oplus m_4 \end{aligned}$$

De manera que cada dígito de comprobación está comprobado por al menos tres dígitos de la palabra. Cada dígito de comprobación se calcula mediante la operación binaria XOR (Bruce, Crilly & Rutledge, 2002).

3.2 Tipos de Modulación Digitales

Ya que no es eficiente propagar las señales de información por los distintos canales de comunicación debido a los múltiples factores que alteran o limitan la señal, es necesario modular la información de la fuente mediante una señal que cumpla con las características apropiadas de acuerdo al canal de comunicación (como se especificó en el Capítulo 2), a esta señal se le conoce como señal portadora. La modulación consiste en cambiar alguna de las características de la portadora (amplitud, frecuencia o fase) mediante la señal de información, la cual puede ser analógica o digital. Dependiendo la característica de la señal portadora que se modifica y del tipo de señal de información, es la modulación que se realiza.

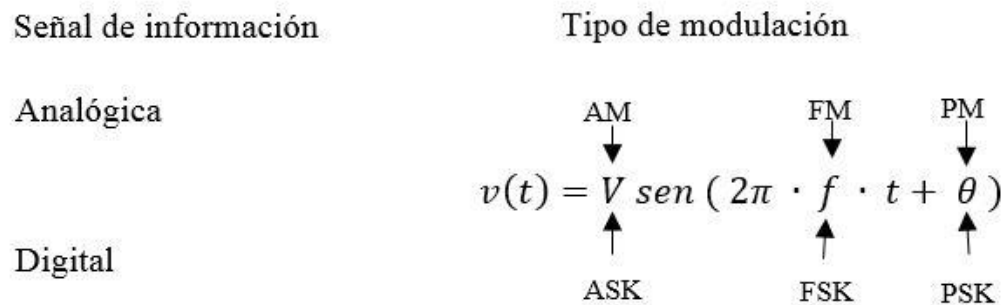


Figura 20. Tipos de Modulación (Wayne, 2003).

La figura 20 muestra de manera simplificada los tipos de modulaciones más comunes, donde se puede observar que una señal de información analógica puede modular la portadora en amplitud (AM), en frecuencia (FM) y en fase (PM); mientras que una señal de información digital puede modular a la portadora en amplitud (ASK) en frecuencia (FSK) y en fase (PSK).

La técnica de manipulación por amplitud digital o ASK (Amplitude Shift Keying), es la modulación digital más sencilla, consiste en una modulación con portadora completa y doble banda lateral. Teniendo una señal de información binaria, la ecuación de la modulación ASK es:

$$V_{am}(t) = [1 + V_m(t)] \left[\frac{A}{2} \cos(\omega_c t) \right]$$

Donde:

$V_{am}(t)$ = Voltaje de la onda de amplitud modulada (volts).

$\frac{A}{2}$ = Amplitud de la portadora (volts).

$V_m(t)$ = Señal de información binaria (volts).

$V_{am}(t)$ = Voltaje de la onda de amplitud modulada (volts).

ω_c = Frecuencia de la portadora (radianes/segundo).

En la ecuación anterior, ya que la señal moduladora $V_m(t)$ es una forma de onda binaria, en la que el '1' lógico es representado por un nivel de voltaje positivo (+1V), y el '0' lógico es representado mediante un nivel de voltaje negativo (-1V), por lo que la ecuación queda de la siguiente manera:

$$\text{Para un '0' lógico } V_{am}(t) = [1 + 1] \left[\frac{A}{2} \cos(\omega_c t) \right] = A \cos(\omega_c t)$$

$$\text{Para un '1' lógico } V_{am}(t) = [1 - 1] \left[\frac{A}{2} \cos(\omega_c t) \right] = 0$$

De manera que para la modulación ASK, se tienen dos valores, $A \cos(\omega_c t)$ o 0. En la figura 21 se puede observar el esquema de la modulación, donde se aprecia que para un '1' lógico, la señal modulada es la misma que la portadora; mientras que para un '0' lógico, la señal modulada es 0, es decir, la portadora apagada.

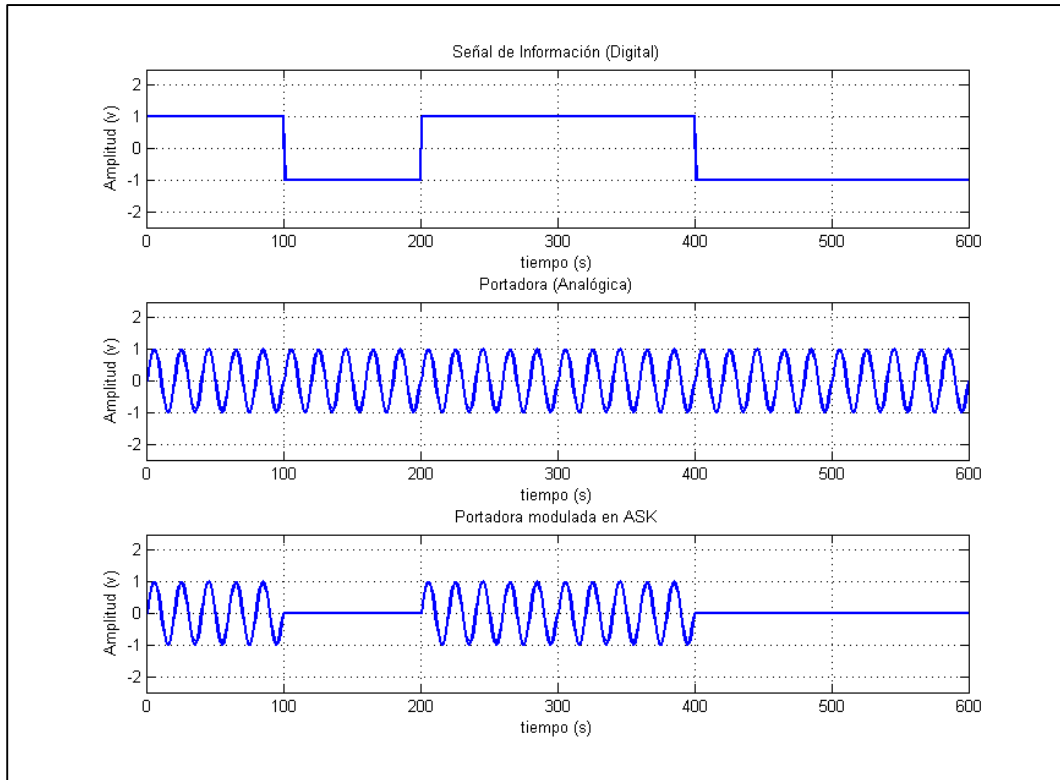


Figura 21. Modulación ASK. a) Señal de información. b) Portadora. c) Portadora Modulada.

La manipulación por desplazamiento de frecuencia (FSK, Frequency Shift Keying), es otra técnica de modulación digital, en la cual la amplitud de la señal es constante mientras que el ángulo de la señal varía. La ecuación que describe la modulación es la siguiente:

$$V_{fsk}(t) = V_c \cos[2\pi[f_c + V_m(t)\Delta f]t]$$

Donde: $V_{fsk}(t)$ = forma de onda FSK (volts).

V_c = amplitud de la portadora (volts).

f_c = frecuencia central de la portadora (Hz).

Δf = desviación máxima de frecuencia (Hz).

$V_m(t)$ = señal moduladora de entrada binaria (± 1).

En la ecuación anterior, el desplazamiento máximo de frecuencia de la portadora Δf , es proporcional a la amplitud y a la polaridad que tome la señal binaria de entrada. La señal moduladora binaria de entrada es la misma que en el caso de la modulación ASK, de manera que la ecuación queda de la siguiente manera:

Para un '0' lógico $V_{fsk}(t) = V_c \cos[2\pi[f_c - \Delta f]t]$

Para un '1' lógico $V_{fsk}(t) = V_c \cos[2\pi(f_c + \Delta f)t]$

En la figura 22 se observa el esquema correspondiente a la modulación FSK, donde se aprecia que la señal binaria de entrada, desvía la frecuencia de la portadora.

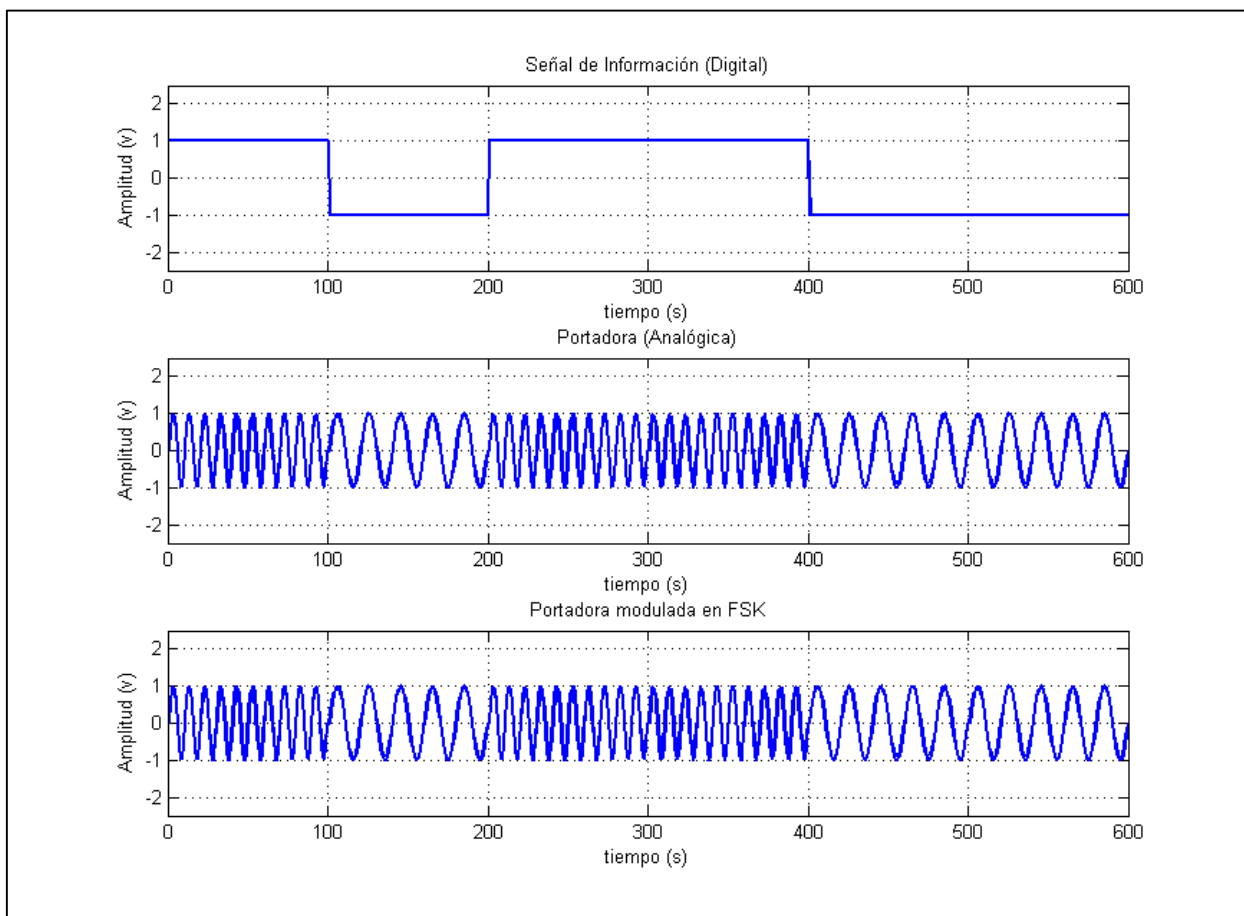


Figura 22. Modulación FSK. a) Señal de información. b) Portadora. c) Portadora Modulada.

PSK (Phase-Shift Keying) o Manipulación por Desplazamiento de Fase es una forma de modulación digital donde la señal de entrada es una señal digital y la portadora una señal analógica. En esta modulación se pueden tener varias fases de salida dependiendo del número de portadoras que se tengan y de aquí se obtienen los distintos tipos de modulación PSK que existen (BPSK, QPSK, 8PSK, 16PSK y 32PSK) (Halsall & García, 1998).

En la manipulación por desplazamiento binario de fase (BPSK) se tienen dos fases de salida posibles con una sola portadora. La señal de información digital modula en fase a la portadora de tal manera que una de las fases de salida representa un “1” lógico y la otra representa un “0” lógico como lo muestra la figura 23.

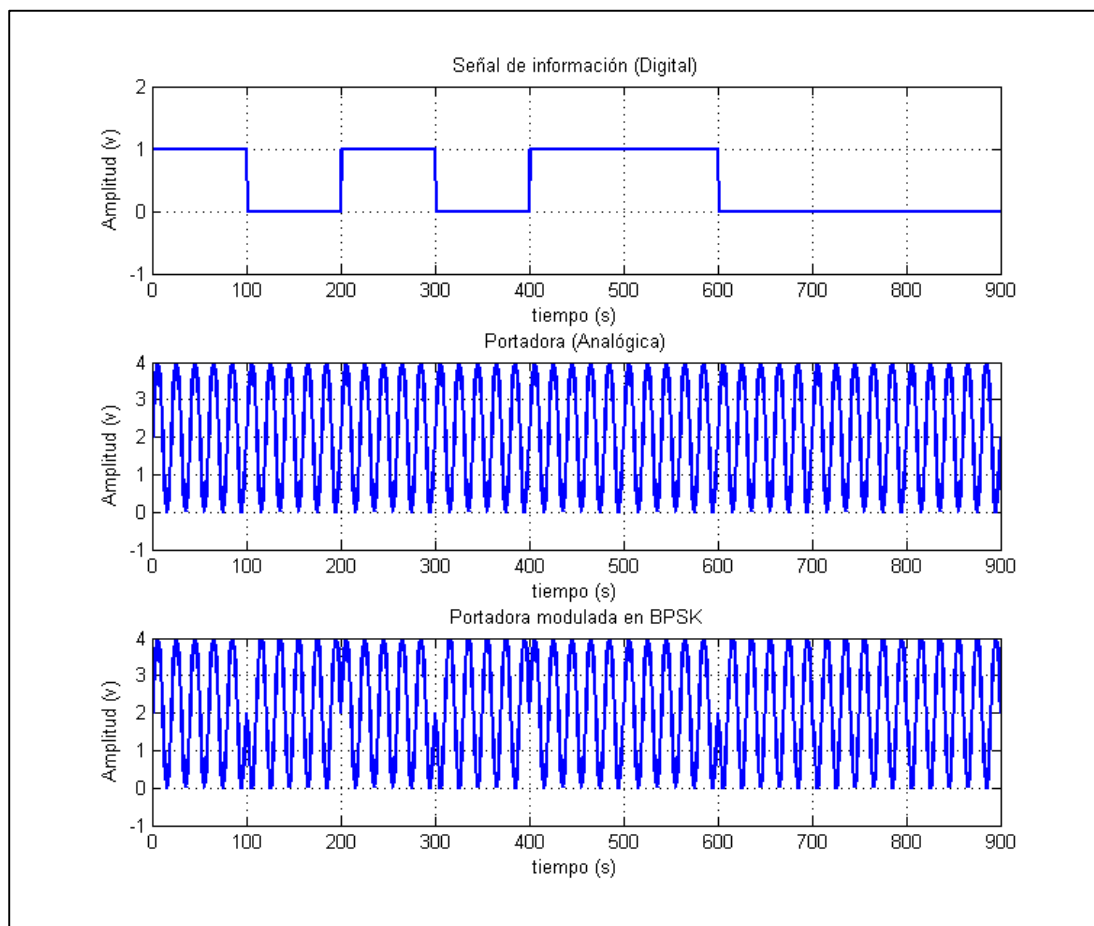


Figura 23. Modulación BPSK. a) Señal de información. b) Portadora. c) Portadora Modulada.

3.3 Modulación en Fase (BPSK)

En el esquema básico de BPSK que se muestra en la figura 24, en donde se tiene: el oscilador de la portadora que entra a un modulador balanceado el cual actúa como un conmutador de fase que dependiendo de la señal de entrada es la fase de salida. Esta señal de entrada primero pasa por un convertidor de nivel lógico el cual cambia la señal cuadrada con valores de 0 a +V a valores de $-V$ y +V para que multiplique a la portadora y de esta manera sea desfasada 180° o enfasada 180° .

Otra forma de analizar la modulación BPSK a la salida puede ser mediante las señales $+sen(\omega_c t)$ y $-sen(\omega_c t)$; donde la primera señal representa la señal enfasada con el oscilador de la portadora, mientras que la segunda señal representa la señal desfasada 180° respecto al oscilador de la portadora (Wayne, 2003). El esquema que visualiza lo anterior, se puede observar en la figura 25 mediante la tabla de verdad, el diagrama fasorial y el diagrama de constelación.

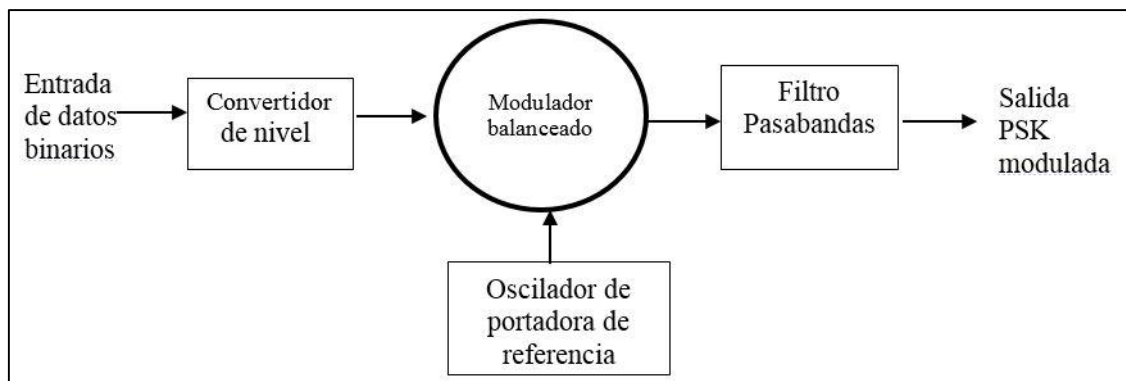


Figura 24. Diagrama de un modulador BPSK.

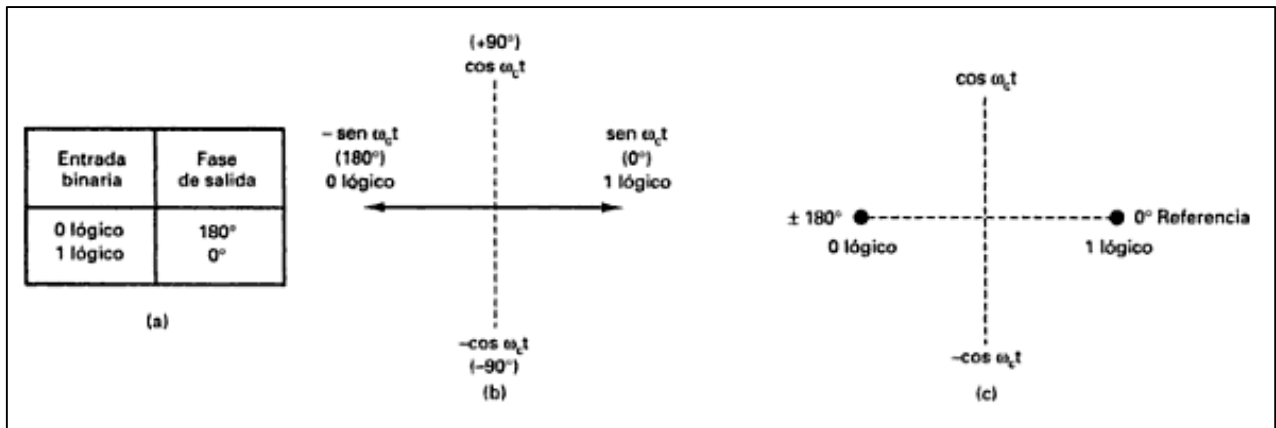


Figura 25. Fases de BPSK. a) Tabla de verdad. b) diagrama fasorial. c) Diagrama de constelación (Wayne, 2003).

Cada vez que cambia la señal de entrada, cambia la señal de salida por lo que en BPSK la tasa de cambio de salida (baudios) es igual a la tasa de cambio de entrada (bps).

La frecuencia fundamental f_a de una secuencia alternada de bits 1 y 0 es igual a la mitad de la frecuencia de bits ($f_b/2$), por lo que La ecuación de la salida de un modulador BPSK es proporcional a:

$$BPSK = \sin(2\pi f_a t) \cdot \sin(2\pi f_c t)$$

En donde: f_a = frecuencia fundamental máxima de la entrada binaria (Hz)

f_c = frecuencia de portadora de referencia (Hz)

El espectro BPSK se puede calcular mediante la frecuencia lateral superior e inferior máximas. Teniendo en cuenta que el ancho de banda máximo se da cuando hay una sucesión alternada de 1's y 0's lógicos. Si se despeja la ecuación de BPSK mediante identidades trigonométrías, se obtiene:

$$BPSK = 1/2 \cos[2\pi(fc - fa)t] - 1/2 \cos[2\pi(fc + fa)t]$$

La transformada de Fourier de esta señal nos da:

Espectro BPSK

$$= \frac{\pi}{2} [\delta(\omega - (\omega_c - \omega_a)) + \delta(\omega + (\omega_c - \omega_a)) + \delta(\omega - (\omega_c + \omega_a)) + \delta(\omega + (\omega_c + \omega_a))]$$

Por lo que las funciones δ son 1 para $\pm(fc - fa)$ y $\pm(fc + fa)$, para cualquier otro valor de frecuencia, la función δ es 0.

Así, el ancho de banda bilateral mínimo de Nyquist F_n , es

$$(fc + fa) - (fc - fa) = 2fa$$

Y como se describió anteriormente, la frecuencia fundamental de la señal es $fa = fb/2$, por lo tanto:

$$B = (2fb)/2 = fb$$

Siendo la rapidez de entrada de bits fb el ancho de banda mínimo bilateral de Nyquist.

Si tenemos una señal portadora de 10KHz y una señal de información o moduladora de 100Hz, el espectro BPSK resultante viene dado por la figura 26. En la figura se puede observar que las bandas laterales se dan en 9.9KHz y 10.1KHz por lo que el ancho de banda corresponde a 200Hz y la velocidad de transmisión es de 100bps.

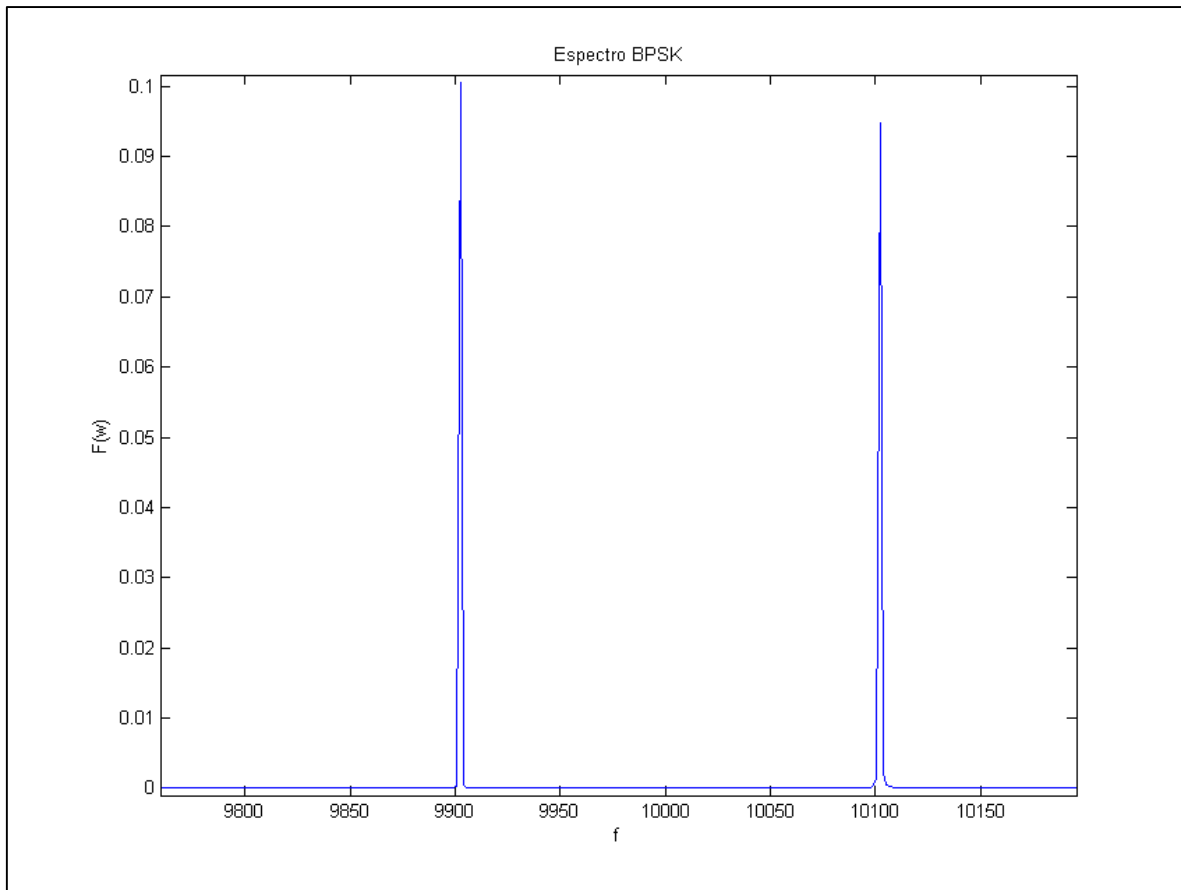


Figura 26. Espectro BPSK para una portadora de 100KHz y una moduladora de 100Hz.

4. DESARROLLO E IMPLEMENTACION

La metodología utilizada para el desarrollo del transmisor digital de datos se basó en la señalada en el Capítulo 2, quedando como lo muestra la figura 27.

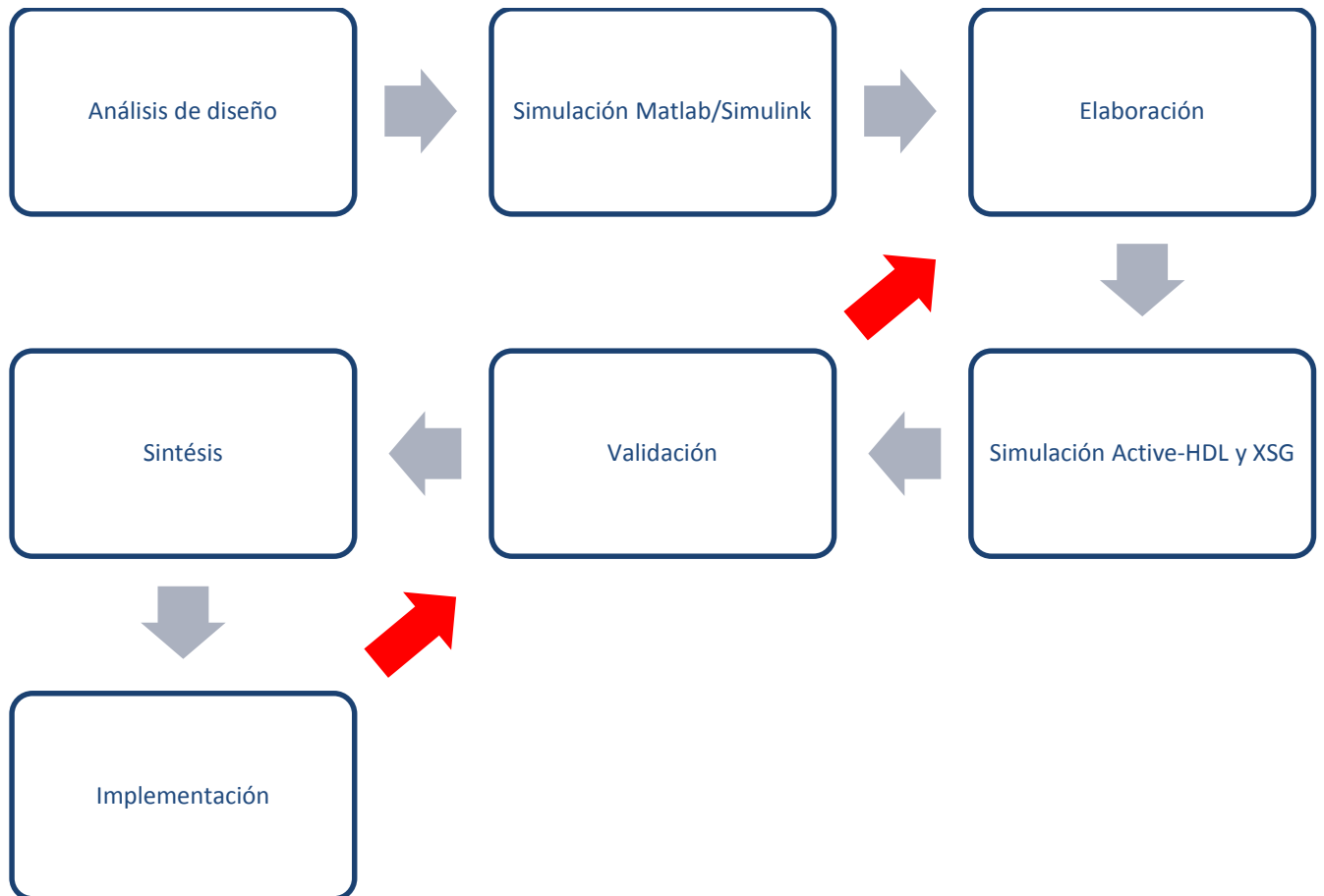


Figura 27. Metodología utilizada.

4.1 Análisis de Diseño

El diagrama a bloques que conforma el transmisor de datos digital, se desarrolla a partir de la figura 2, tomando solamente la parte del transmisor y declarando cada uno de los bloques a implementar. En la figura 28 se muestran cada uno de los bloques que conforman el transmisor de datos digital propuesto.



Figura 28. Diagrama del Transmisor de datos digital.

La fuente de información, puede ser cualquier señal binaria aleatoria, generada arbitrariamente o mediante un banco de pruebas.

El codificador de Hamming implementado es un código (7,4) como el especificado en el Capítulo 3. De manera que, si queremos transmitir la palabra “1100”, el desarrollo del código de Hamming será el que se muestra en la tabla 4. Los dígitos de comprobación se encuentran ubicados en las posiciones binarias donde solo hay un ‘1’, que corresponden a las posiciones 1, 2 y 4 en binario. Las otras posiciones se llenan con los dígitos del mensaje. Para el cálculo de los dígitos de comprobación, se utiliza la relación de la tabla 4 para la obtención de las ecuaciones. Los dígitos de mensaje que se encuentran en cada ecuación de comprobación, corresponden a las posiciones binarias que contienen un ‘1’ en la misma posición que la del dígito de comprobación a calcular.

Tabla 4. Desarrollo del código de Hamming (7,4).

Palabra	c_1	c_2	1	c_3	1	0	0
Posición (Binaria)	001	010	011	100	101	110	111
Tipo de dígito	c_1	c_2	m_1	c_3	m_2	m_3	m_4
c_1	0		1		1		0
c_2		1	1			0	0
c_3				1	1	0	0
Palabra Codificada	0	1	1	1	1	0	0

Por lo que las ecuaciones para los bits de comprobación, quedan de la siguiente manera:

$$c_1 = 1 \oplus 1 \oplus 0 = 0$$

$$c_2 = 1 \oplus 0 \oplus 0 = 1$$

$$c_3 = 1 \oplus 0 \oplus 0 = 1$$

Obteniendo la palabra codificada “0111100”, posteriormente esta palabra codificada será el conjunto de bits que entren al modulador.

Para el modulador BPSK, se tomó en cuenta las ecuaciones especificadas en el Capítulo 3, se eligió una portadora senoidal con una frecuencia en el orden de los KHz. La estructura del modulador se describe en la figura 29.

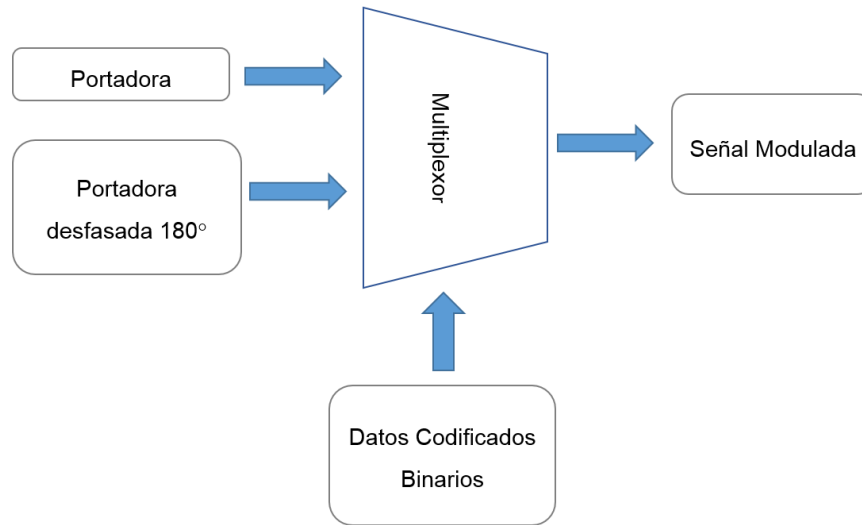


Figura 29. Diagrama del Modulador BPSK.

4.2 Simulación en Matlab/Simulink

4.2.1 Simulación del Codificador

Se realizó un modelo en Simulink para la simulación de un sistema de comunicación digital simple con un codificador de Hamming y se comparó con otro sistema sin codificación, mediante un canal binario simétrico con una probabilidad de error del 2%. De esta manera se puede comprobar la eficiencia y fiabilidad del codificador a implementar. Los bloques utilizados para la creación del modelo son: *Bernoulli Binary Generator*, *Hamming Encoder*, *Binary Symetric Channel*, *Error Rate Calculation* y *Display*. El diagrama a bloques del modelo se muestra en la figura 30 (Giordano & Levesque, 2015).

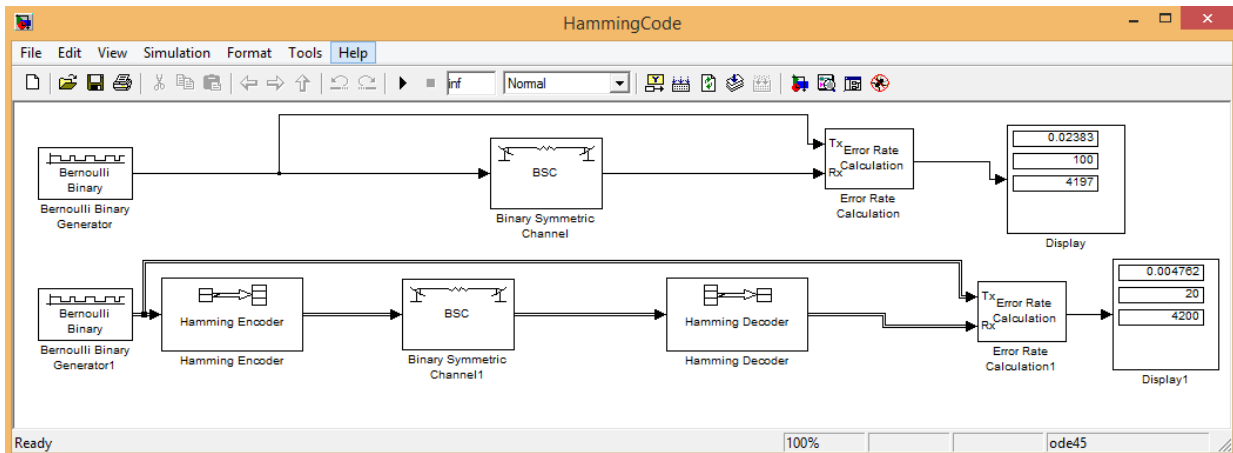


Figura 30. Diagrama del Modelo del Codificador de Hamming.

El generador binario de Bernoulli, devuelve una salida pseudoaleatoria de dígitos binarios en un arreglo de 4 bits. El bloque de Canal Simétrico Binario, es un generador de ruido que inyecta al sistema una probabilidad de error del 2%. Los bloques del Codificador y Decodificador de Hamming realizan la adición de los bits de comprobación en el transmisor y la validación de los datos en la recepción respectivamente.

El bloque de Cálculo de Tasa de Errores, compara la entra del sistema en el transmisor con la salida del receptor para encontrar errores generados. Los datos transmitidos y los errores generados son mostrados en el bloque *Display*.

4.2.2 Simulación del Modulador

Para el modelo en Simulink del modulador BPSK, se realizó la implementación en base a la figura 29. Para este modelo se usó un bloque para *sine wave* para la generación de la portadora y otro para generar la portadora desfasada 180°, un *generador binario de Bernoulli*; un bloque *switch* que en base a los datos binarios de entrada, funcione como interruptor entre la portadora y la

portadora desfasada. Para visualizar la señal de salida, se usaron los bloques de *Scope* y *Spectrum Scope*, para mostrar la señal en el dominio del tiempo y en el dominio de la frecuencia respectivamente. El modelo de Simulink correspondiente al modulador BPSK se muestra en la figura 31.

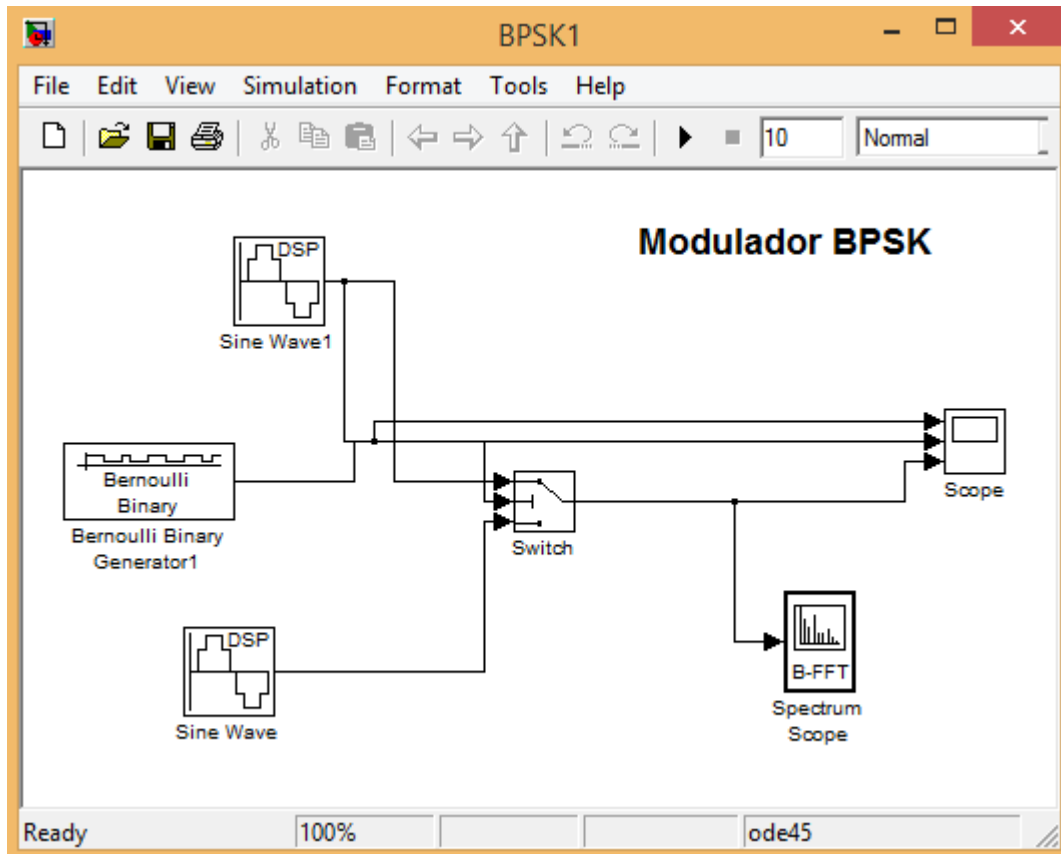


Figura 31. Diagrama del Modelo del Modulador BPSK.

4.3 Elaboración

El sistema se diseñó en base al modelo top-down para la generación del código VHDL, se utilizó el editor de textos de Active-HDL para este proceso. La entidad general del transmisor digital de datos a implementar en Hardware se muestra en la figura 32. Las señales de entrada son: el reloj maestro del sistema (CLK), la señal de RST para resetear las señales que dependen del reloj maestro.

Y la señal INF la cual contiene el grupo de los bits a transmitir sin codificar. Las señales de salida son: las cuatro señales para el DAC, las cuales se describieron en el Capítulo 2 y la señal modulante para su visualización física.

La primera parte del desarrollo, consistió en el diseño de la portadora y la portadora desfasa, para lo que se utilizó el código de Matlab de Rene Troncoso (Romero-Troncoso, 2007) para la generación de una Look Up Table de 12 bits correspondiente a una función senoidal y que se binariza en 4096 valores. De esta manera se puede implementar una aproximación de la función seno en una FPGA. La entidad de LUT de la Portadora consiste en una señal de entrada de 12 bits que se mapea a la salida con su correspondiente valor de la función senoidal. La entidad de la LUT para la portadora desfasada es similar, solo que la función mapeada corresponde a la función senoidal desfasada 180° .

EL bloque del Acumulador de fase realiza una suma sistemática de fase, acumulándola de manera que sirva para desplazarse en los valores de las LUT's de una forma cíclica. La entidad del Acumulador de fase se muestra en la figura 33. Las señales de entrada son: el reloj maestro del sistema (CLK), la señal de RST para resetear las señales que dependen del reloj maestro. Y la señal P de 12 bits la cual establece el valor de suma para el acumulador. La señal de salida 'Q' es la suma acumulada de 12 bits proveniente del registro.

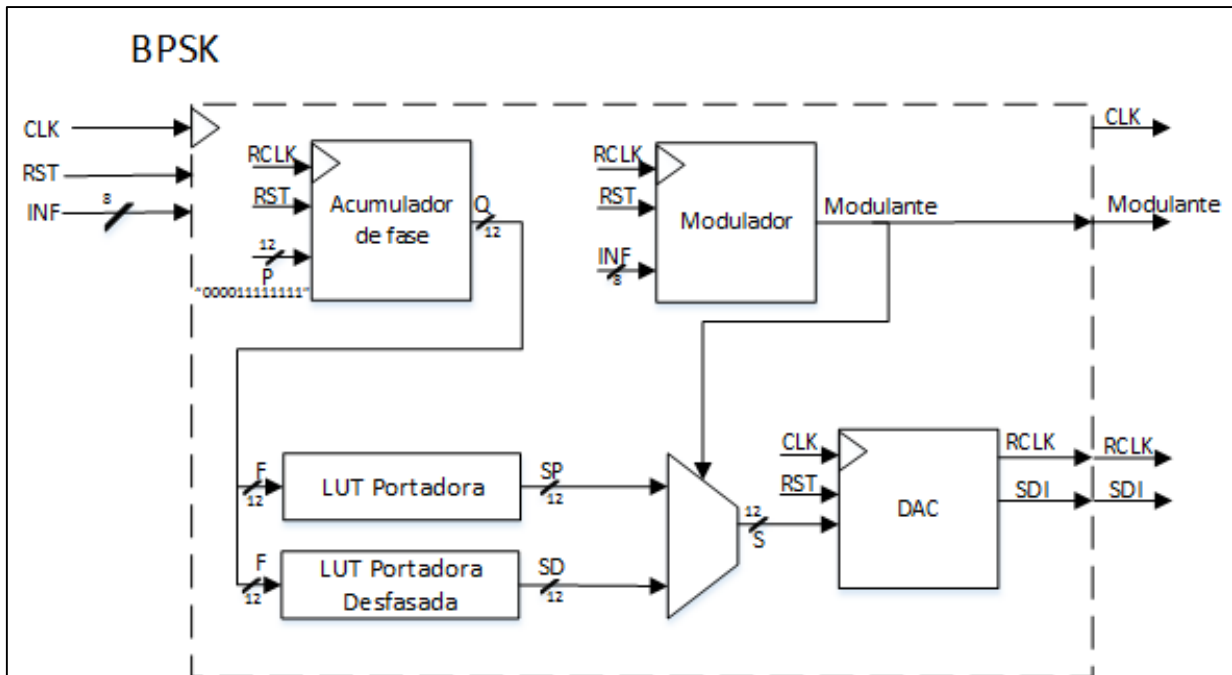


Figura 32. Entidad general del transmisor digital de datos.

El Acumulador de fase consiste en un sumador de 12 bits, una unidad de registro de 12 bits para almacenar la suma acumulada y una unidad de registro de 1 bit para almacenar el acarreo de la sumatoria.

En el sumador de 12 bits, las señales de entrada son: El valor establecido para la sumatoria R1, el valor de la sumatoria acumulada R2 y el acarreo de entrada correspondiente a la salida del registro de acarreo. Mientras que el resultado de la sumatoria se pasa a la salida del sumador. La unidad de registro de 12 bits almacena la sumatoria acumulada y se actualiza en cada ciclo de reloj con respecto a la salida del sumador. Por lo que sus señales de entrada correspondientes son: el reloj maestro, el reset y el resultado del sumador. A la salida se encuentra la señal de la sumatoria acumulada.

En el bloque de acarreo, se almacena el 13vo bit de la sumatoria y pasa como entrada en el sumador. El valor de almacenamiento se actualiza en cada ciclo de reloj igual que la unidad de registro de 12 bits.

En el bloque del Modulador consiste en un divisor de frecuencia, el codificador de Hamming y un buffer para separar los datos codificados y mostrarlos en la salida. La entidad del Modulador se muestra en la figura 34. Las señales de entrada son las mismas que las de la entidad general BPSK, y la señal de salida es la señal binaria codificada.

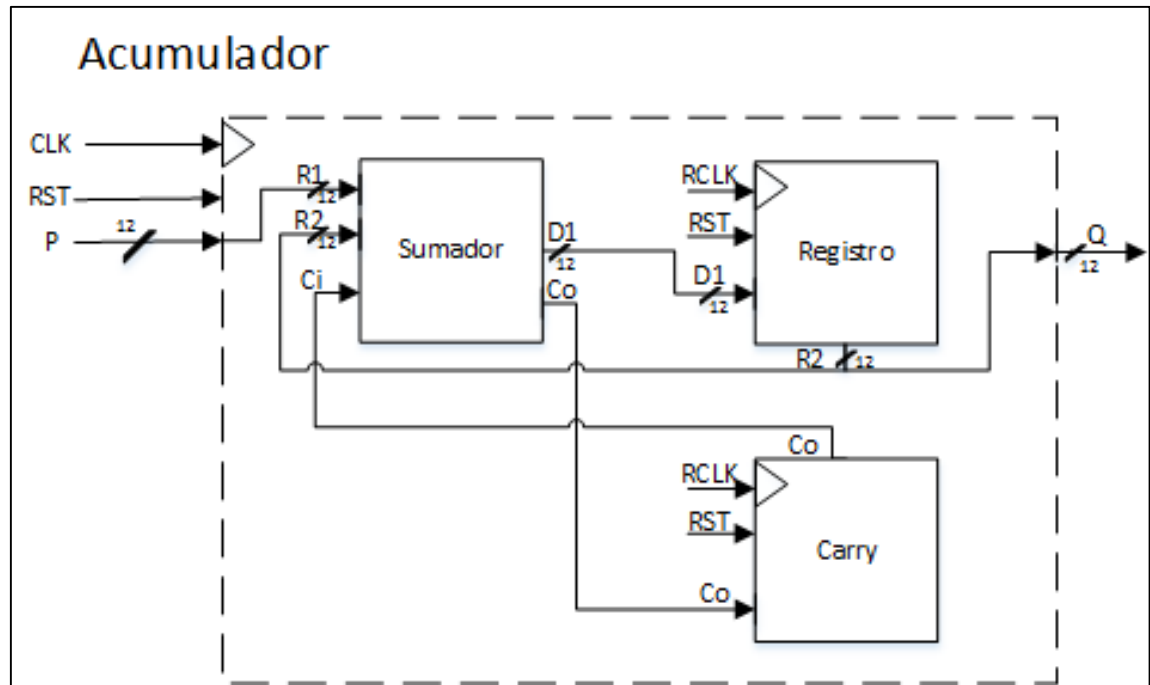


Figura 33. Entidad del Acumulador de fase.

El divisor de frecuencia se establece de acuerdo a la velocidad de bits requerida para la transmisión mediante la siguiente formula, donde M es el valor que debe ser programado para la disminución de la frecuencia de reloj:

$$M = \frac{\text{Periodo deseado}}{\text{Periodo de reloj}}$$

De esta manera la salida del divisor de frecuencia sirve para que el buffer entregue a su salida un bit a la vez proveniente del codificador. El codificador recibe en su entrada el grupo de 4 bits que codifica de acuerdo con las ecuaciones para los bits de comprobación establecidas anteriormente.

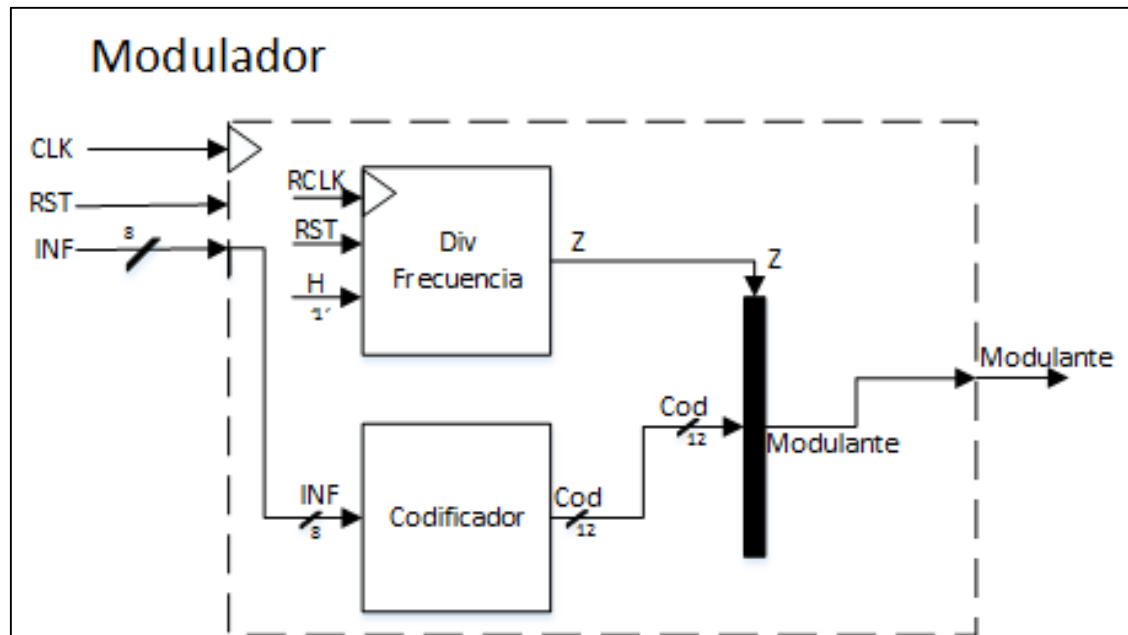


Figura 34. Entidad del Modulador.

Las salidas de las LUT's entran a un multiplexor 2 a 1 y la señal selectora proviene de la salida del bloque del modulador, de esta manera, si la señal selectora es un '0' lógico, la salida del multiplexor entrega la seña de la portadora, y si la señal selectora es un '1' lógico, la salida del multiplexor dará la señal desfasada de la portadora. La salida del multiplexor está formada por 12 bits, los cuales entran al bloque del DAC.

El bloque del DAC contiene una máquina de estados finita para la implementación del protocolo de 24 bits del DAC LTC2624 que se describió en el Capítulo 2. El esquema de la máquina de datos implementada se muestra en la figura 35.

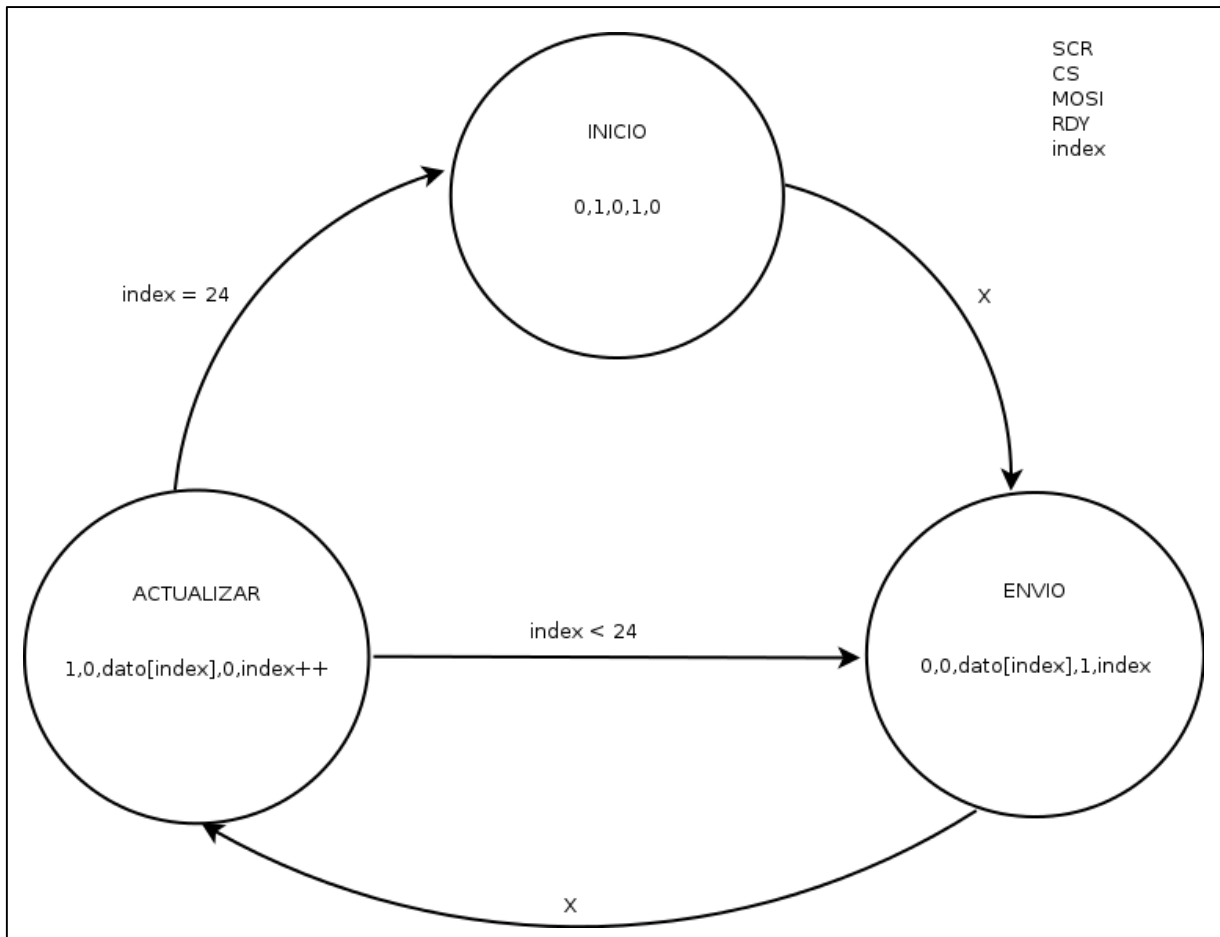


Figura 35. Esquema de la Máquina de Estados Finita para el DAC.

En este esquema se puede observar las señales que interactuarán con la máquina de estados (SCK, CS, MOSI, RDY e Index), la señal SCK funciona como el reloj para la sincronización con el DAC, la señal CS funciona como reset para la comunicación del DAC, la señal MOSI toma el valor de cada uno de los datos del protocolo de 24 bits. La señal RDY funciona para indicar que los 24 bits fueron transmitidos y la señal de index funciona para el desplazamiento de los 24 bits del protocolo. En total son 25 ciclos de reloj, los que se requieren para completar el protocolo.

4.4 Simulación en Active-HDL y XSG

Teniendo el desarrollo en VHDL de las entidades descritas anteriormente, se realizó una simulación en Active-HDL mediante el Visor de Formas de Onda, para la validación del transmisor digital.

La figura 36 muestra el botón para la ejecución del Visor de Formas de Onda, dentro de Active-HDL. También se muestra el cuadro para la configuración de las señales de entrada, donde se les puede asignar una señal de reloj, un valor binario fijo o una tecla para el cambio de estado (estado bajo o '0' lógico y estado alto o '1' lógico).

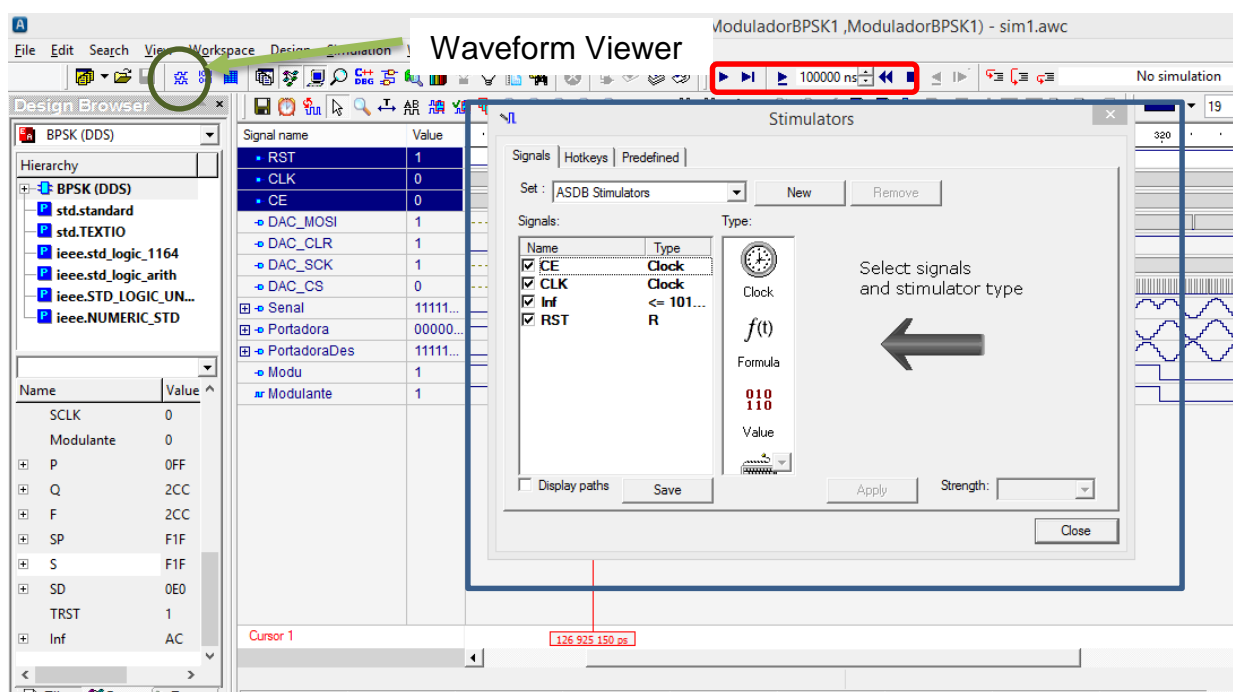


Figura 36. Simulación en Active-HDL.

Una vez que se ejecutó el Visor de Formas de Onda, se seleccionan las señales que se desean visualizar en la simulación, y se arrastran a la pantalla del Visor. Estas señales pueden ser de entrada, salida o intermedias. Ya que se tienen las señales en el Visor, es posible asignar a las señales de entrada, algún valor, como se mencionó anteriormente. Esto se realiza dando click derecho a la señal deseada y seleccionando la opción de *stimulators*, se abrirá la ventana para la configuración de dichas señales (como se muestra en la figura 36 dentro del cuadro azul). Para la ejecución de la simulación se tienen los botones seleccionados dentro del cuadro rojo en la figura 36. Estos botones sirven para indicar: el tiempo de ejecución a realizar, avanzar, retroceder y parar la simulación.

La simulación en XSG nos ayuda a tener otro método para la validación del código VHDL generado, este método genera un entorno similar a lo que sería la implementación en hardware ya que se simula también la capacidad de la tarjeta de desarrollo utilizada, por lo que si un diseño sobrepasa la capacidad de la tarjeta, la simulación no funcionará. Para la creación de este modelo se siguieron los pasos descritos en la sección 2.4.1.1 sobre el uso del bloque *Black Box*. Los bloques adicionales utilizados son: *constant*, *step*, *scope* y *clock probe* (como se muestra en la figura 37).

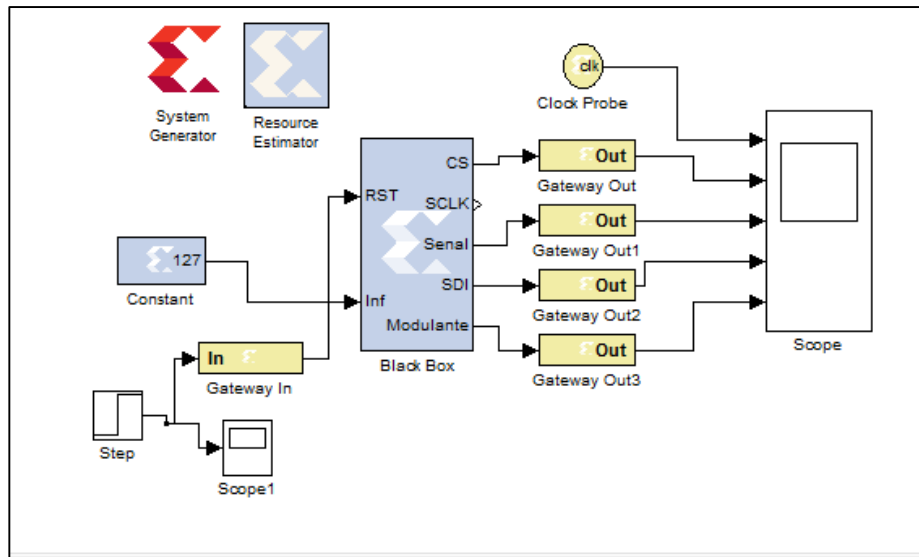


Figura 37. Modelo de la simulación del transmisor en XSG.

El bloque *constant* sirve para ingresar un valor binario a la entrada del transmisor, también se puede usar un generador binario de Bernoulli con su correspondiente bloque *Gateway In*. El bloque *step* es una función escalón unitario que se usa como señal para el reinicio del sistema. El bloque de *clock probe* devuelve a la salida la señal de reloj del sistema y el bloque *scope* visualiza las señales de salida del bloque *Black Box*. Los bloques se conectan como se muestra en la figura 37. En el bloque de *System Generator*, se selecciona la tarjeta Spartan-3E Starter Kit con un reloj de 50MHz, para simular su capacidad.

4.5 Síntesis

Para la síntesis del código VHDL se usa Xilinx ISE, por lo que se exportó el código generado en Active-HDL y se importó en Xilinx ISE. Un paso previo a la síntesis es la asignación de los pines de la FPGA con su correspondiente señal del transmisor, usando la herramienta PlanAhead y la tabla 5 que indica la señal y el pin utilizado de la FPGA. La figura 38 muestra la asignación de pines realizada.

Una vez que se tiene la asignación de los pines, se realiza la síntesis para la abstracción a bajo nivel del código VHDL, esto da como resultado la generación del archivo de programación (.bit) que se utiliza para la implementación en el hardware. El archivo (.bit) es creado con el mismo nombre del archivo VHDL de más alto nivel y en el mismo directorio en el que se encuentra el proyecto creado en Xilinx ISE.

Tabla 5. Tabla para la asignación de pines en la FPGA.

Señal	Tipo de señal (Entrada o Salida)	Numero de Pin en la FPGA
AD_CONV	Salida	P11
AMP_CS	Salida	N7
FPGA_INIT_B	Salida	T3
SF_CE0	Salida	D13
SPI_SS_B	Salida	U3
CLK	Entrada	C9
RST	Entrada	L3
Modu	Salida	D7
Enc	Salida	F12
DAC_CLR	Salida	P8
DAC_CS	Salida	N8
DAC_MOSI	Salida	T4
DAC_SCK	Salida	U16

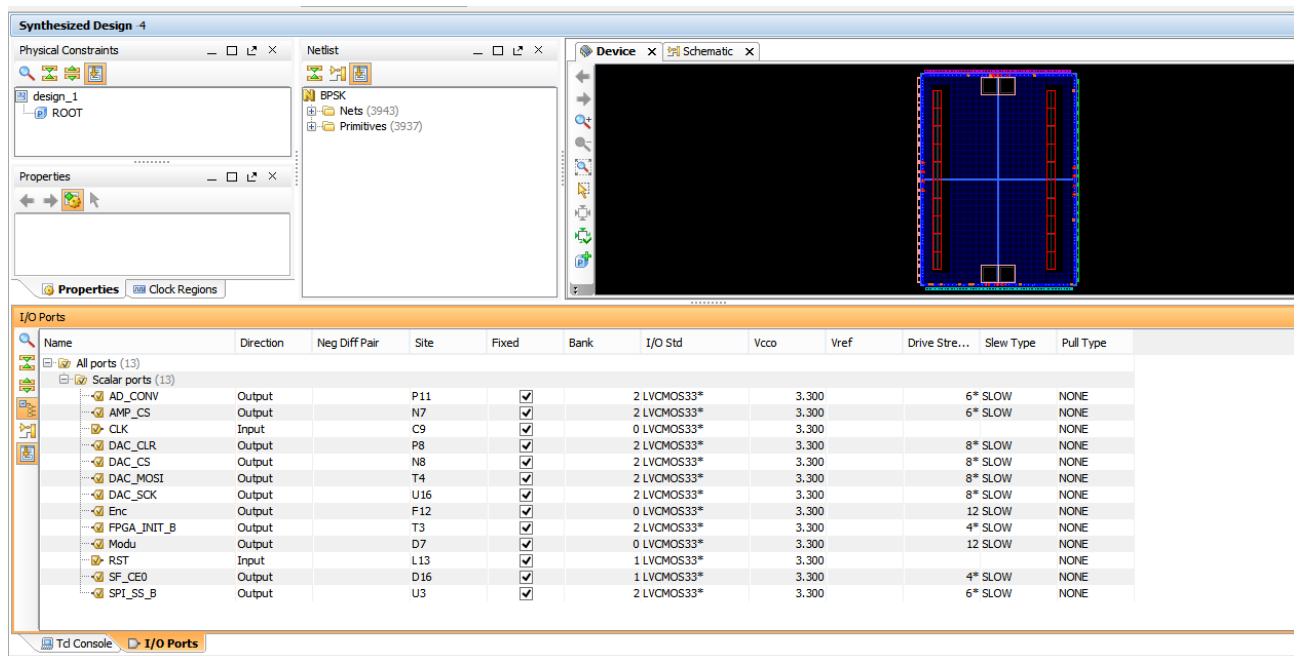


Figura 38. Asignación de las señales de entrada y salida.

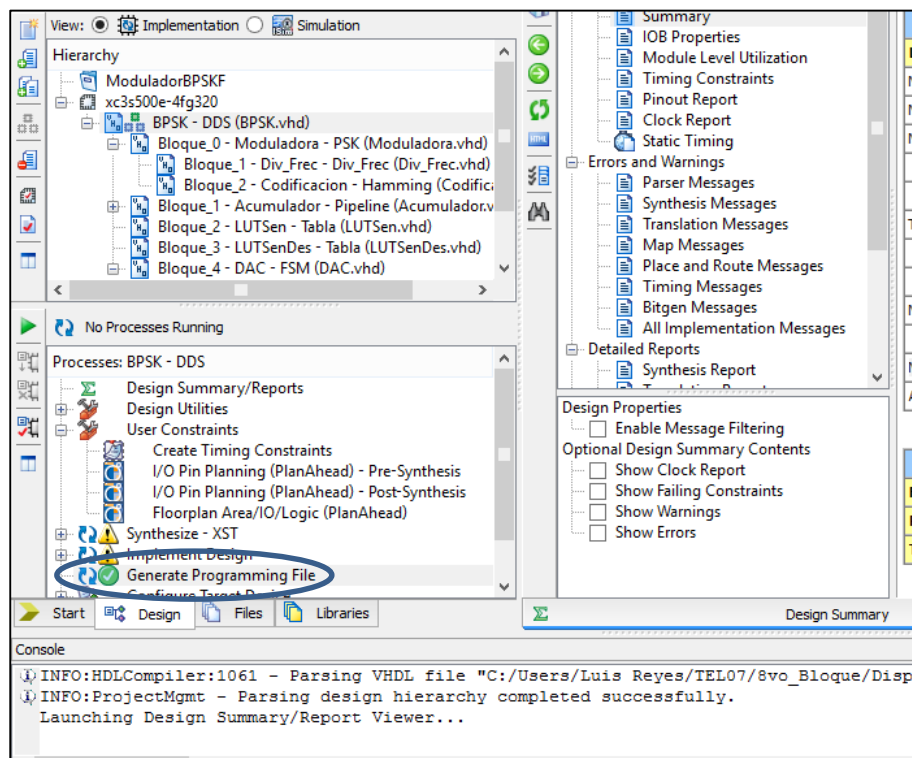


Figura 39. Síntesis mediante Xilinx ISE.

4.6 Implementación

Para la programación del (.bit) en la tarjeta de desarrollo, se utilizó la herramienta de iMPACT, como mencionó en el Capítulo 2. Una vez que se tiene conectada correctamente la tarjeta mediante el puerto USB de programación, iMPACT escanea la tarjeta y genera la cadena de elementos programables. Con click derecho en el Chip XC3S500E de la cadena generada, se selecciona la opción *add a new configuration file* para abrir la ventana de búsqueda del archivo (.bit) como se muestra en la figura 40.

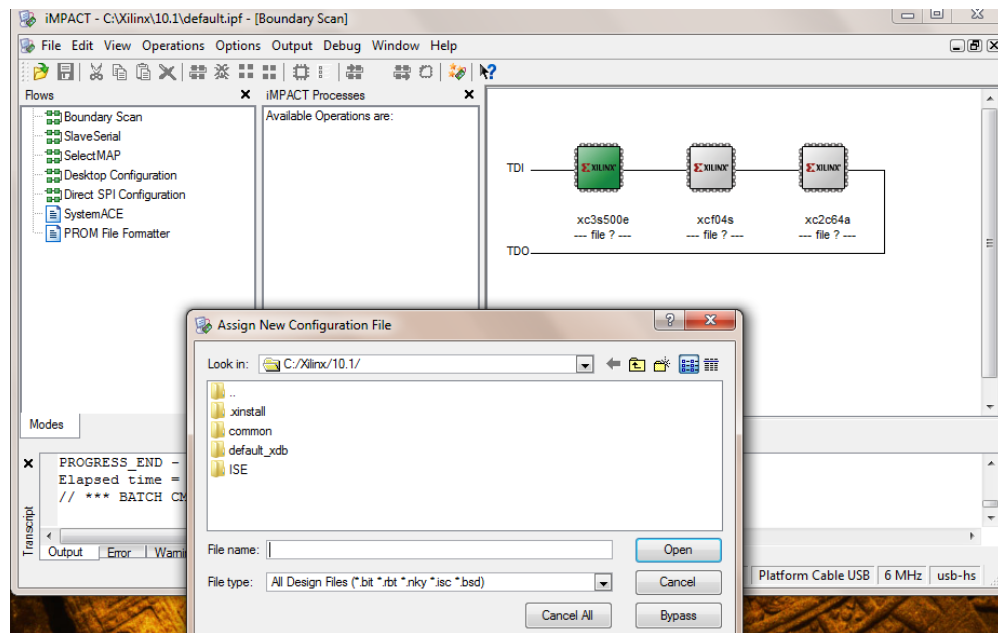


Figura 40. Selección del archivo (.bit).

El último paso para concluir la implementación, es indicar la programación del elemento, dando click derecho sobre él, seleccionando la opción *Program*; lo cual dará como resultado el mensaje azul de la figura 41.

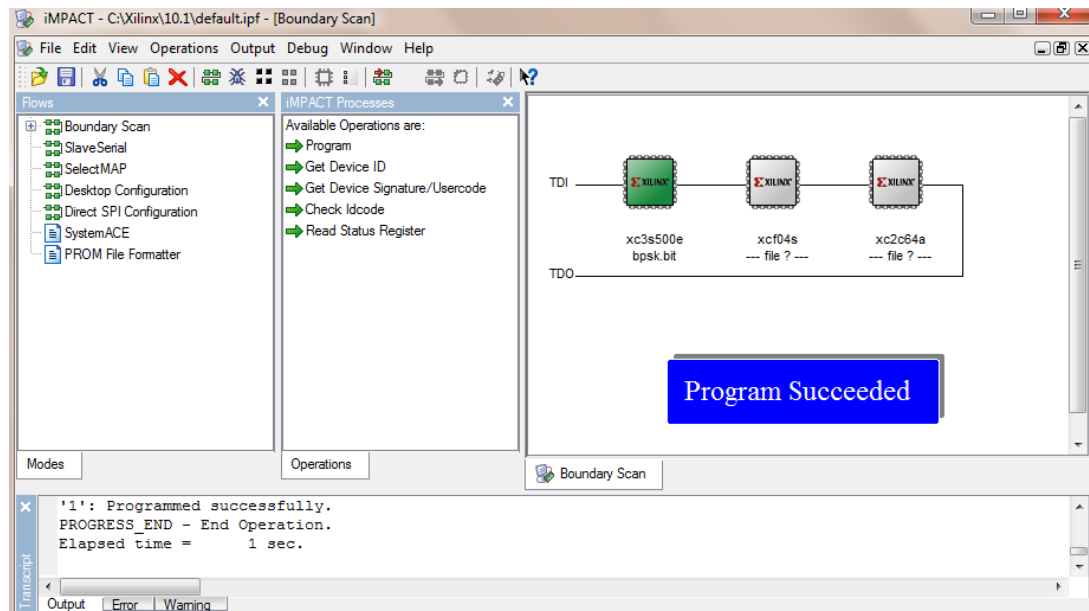


Figura 41. Programación de la tarjeta de desarrollo.

La conexión de la tarjeta de desarrollo y el osciloscopio digital se llevó a cabo utilizando dos puntas para osciloscopio. La primera punta conectada a la salida del canal A del DAC, usando la tierra que contiene el conector de 6 pines del DAC. Esta punta contendrá la señal modulada.

La segunda punta servirá para visualizar la señal de información, por lo que se conecta en el pin D7 de la tarjeta como se especificó en la tabla 5. La tierra para esta punta puede ser la misma que para la primera punta.

La figura 42 muestra la conexión física de la tarjeta de desarrollo y el osciloscopio, donde se puede observar las dos puntas del osciloscopio y el cable de alimentación de la tarjeta.

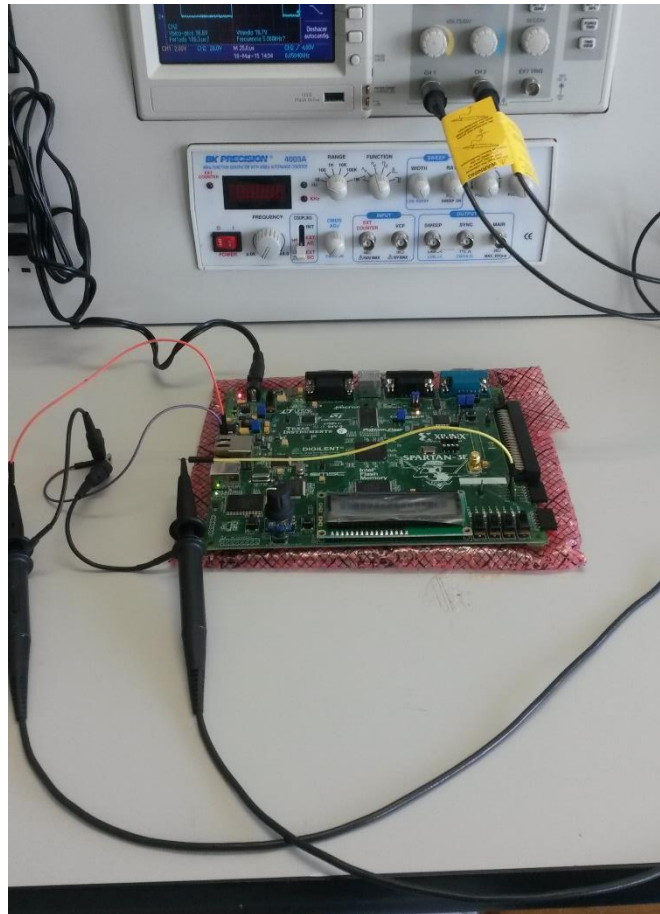


Figura 42. Conexión de la tarjeta de desarrollo y el osciloscopio digital.

5. ANÁLISIS DE RESULTADOS

5.1 Resultados de la Simulación

Los resultados de la simulación del codificador en Simulink se muestran en la figura 43. Donde se observa que para el sistema sin codificación, de 4197 bits enviados, 100 de ellos tuvieron error. Dando una probabilidad de error del 2.38%. Mientras para el sistema con codificación de Hamming se obtuvo una probabilidad de error del 0.47% debido a que de 4200 bits enviados, solamente 20 de ellos se encontraban alterados.

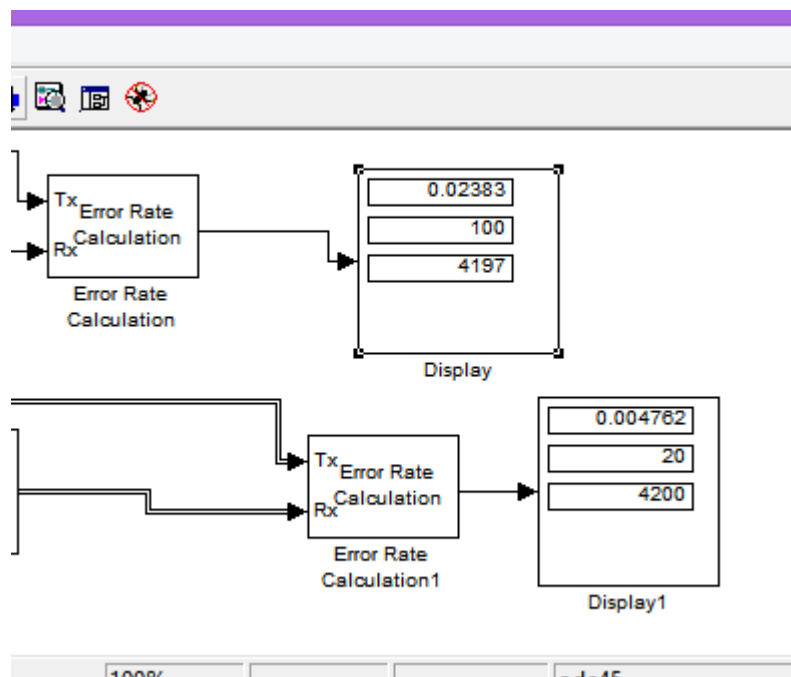


Figura 43. Resultados de la simulación de la codificación de Hamming en Simulink.

Respecto a la simulación del modulador BPSK en Simulink, los resultados se pueden observar en la figura 44, donde se muestra la señal de información en la parte superior, la portadora se encuentra en medio y la señal modulada en la parte inferior. La portadora es una señal senoidal de 6Hz mientras que la señal de información tiene una frecuencia de 3Hz, es decir, la mitad de la frecuencia de la portadora.

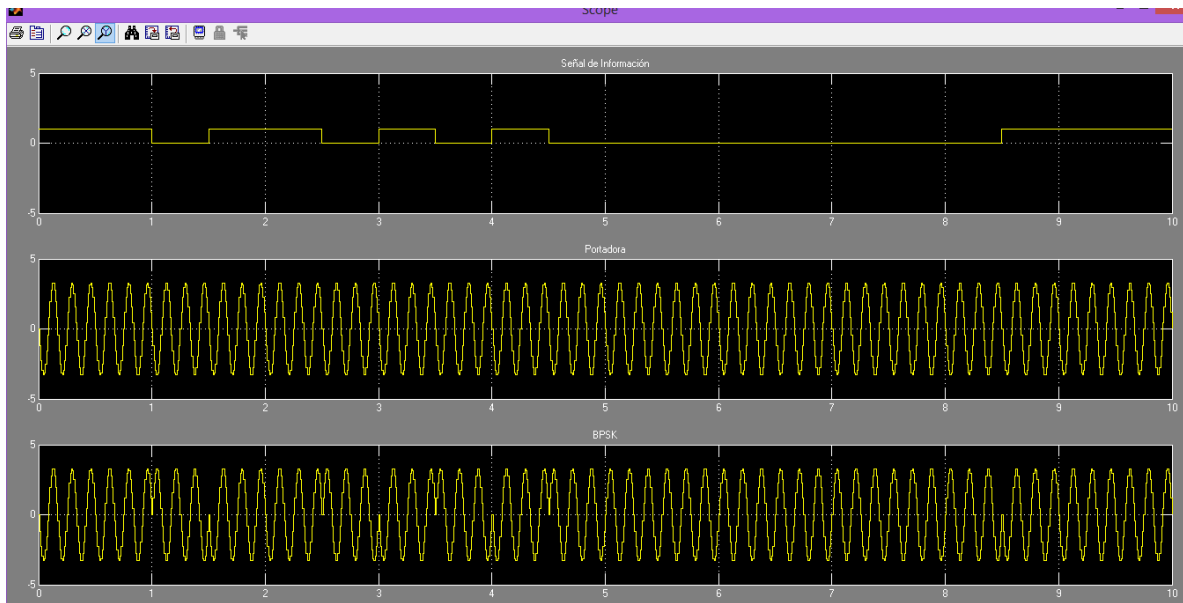


Figura 44. Resultados de la modulación BPSK en Simulink. a) Señal de información. b) Portadora. c) Señal BPSK.

En la figura anterior, se puede apreciar como la señal se desfasa 180° cada vez que hay un cambio en la señal binaria. Si los cambios en la señal binaria son consecutivos, el cambio en la portadora se da cada 3 periodos. El ancho de banda de esta señal se puede observar en la figura 45, donde se presentan las bandas laterales, partiendo de la frecuencia central de la portadora y desplazándose casi 3Hz a la izquierda y a la derecha. Esto debido a que la señal de información no es una señal de 1's y 0's consecutiva, por lo que esta señal no alcanza el ancho de banda máximo que se describió en el Capítulo 3.

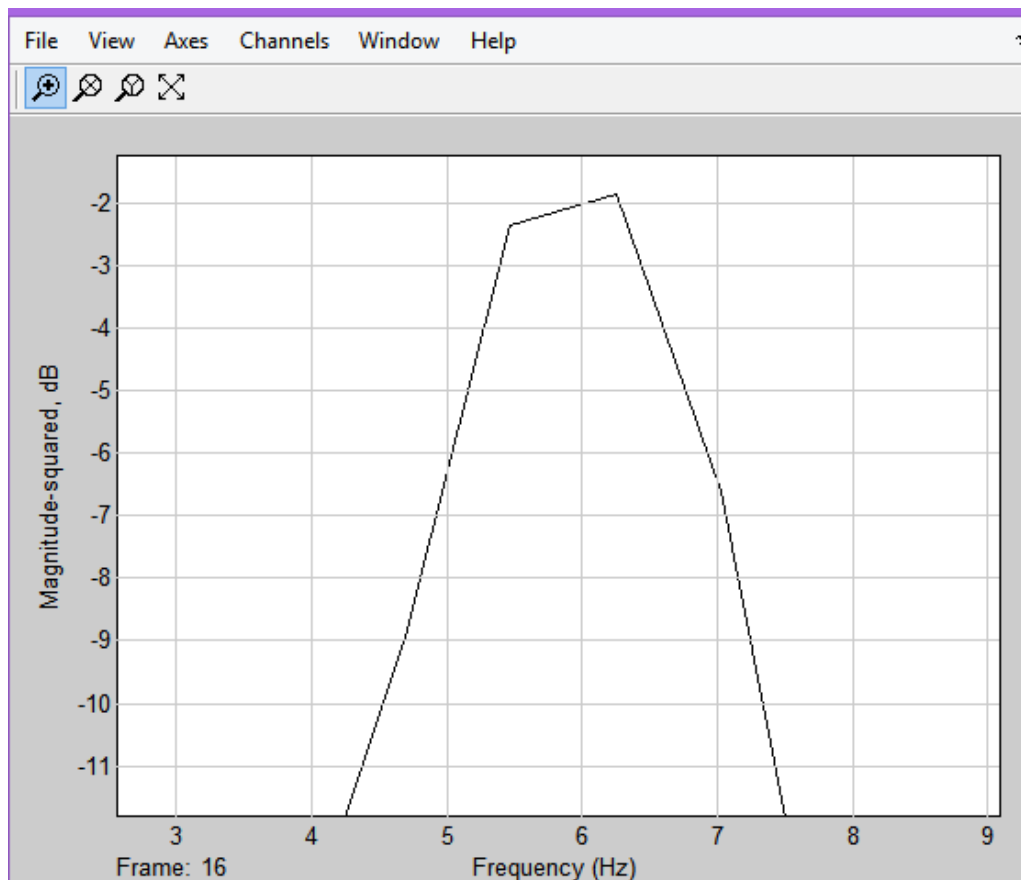


Figura 45. Ancho de banda de la señal BPSK simulada.

Los resultados de la simulación en Active-HDL se muestran en la figura 46. Se observa la señal generada por LUT de la portadora y de la portadora desfasada. La señal modulada realiza los cambios de fase de acuerdo a la señal de información o modulante. La frecuencia de la portadora está limitada ya que con un reloj maestro de 50MHz, el DAC puede trabajar hasta 25MHz. Ya que el protocolo para la comunicación consta de 25 estados, tenemos que la frecuencia de actualización del DAC es de 1MHz. Por lo que cada cambio en la salida del DAC se da a 1ns. Si definimos una portadora senoidal en 16 valores discretos junto con la frecuencia de actualización del DAC, obtenemos una frecuencia de 62.5KHz

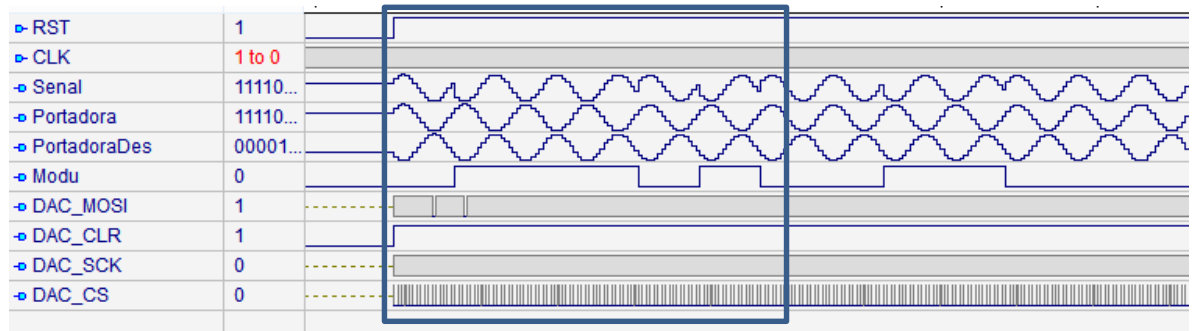


Figura 46. Simulación en Active-HDL.

Los resultados de la simulación utilizando XSG y Simulink se observan en la imagen 47. Al comparar estos resultados con los de la simulación en Active-HDL se obtienen resultados idénticos cuando la señal binaria tiene la misma secuencia en las dos simulaciones. Se aprecia los cambios de fase que realiza la portadora.

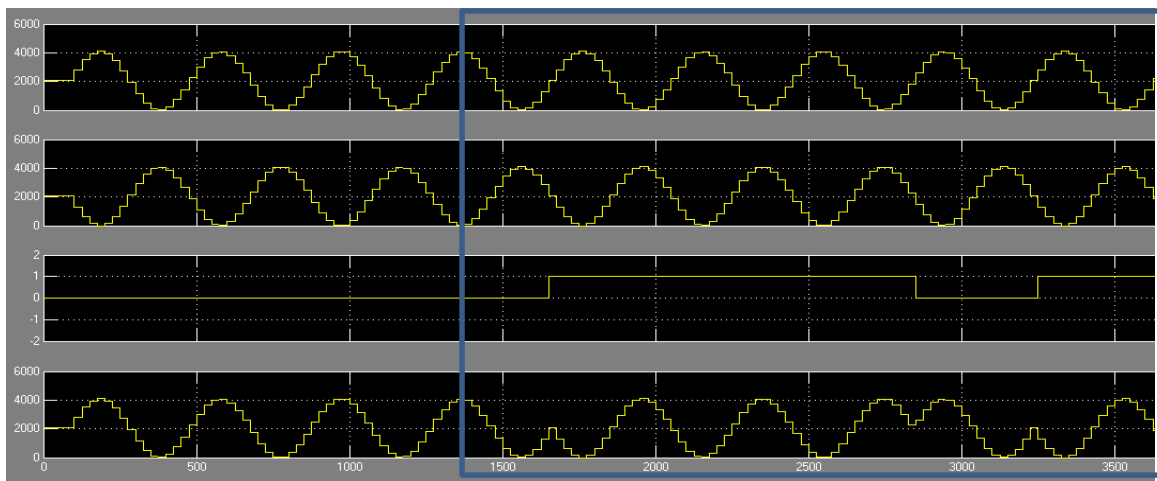


Figura 47. Simulación en XSG y Simulink.

5.2 Resultados de la Implementación

En cuanto a la implementación en la tarjeta de desarrollo del transmisor digital incluyendo la codificación y la modulación, la figura 48 muestra el despliegue de la señal de información y la señal modulada en el osciloscopio. Mientras que en la computadora se muestra la forma esperada para estas señales. Se puede apreciar que tanto la señal de información como la señal BPSK coinciden en ambos despliegues.

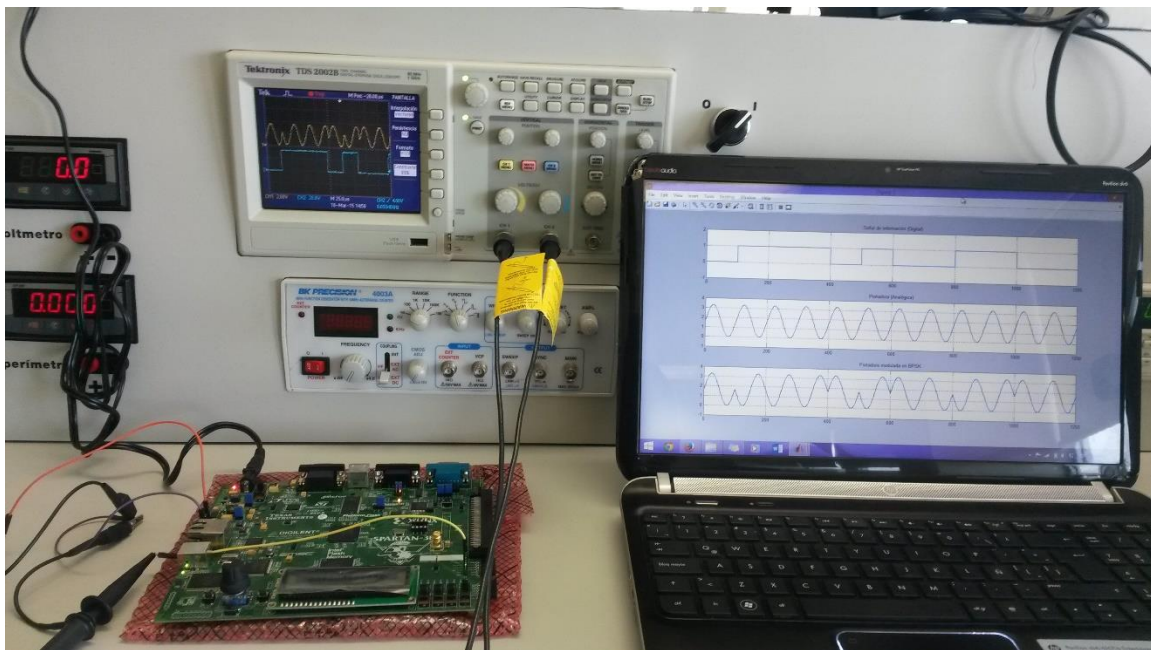


Figura 48 . Implementación en la Tarjeta de desarrollo.

La figura 49 muestra la señal portadora y la señal de información en el osciloscopio por separados y la figura 50 muestra la señal modulada respecto a la señal de información codificada.

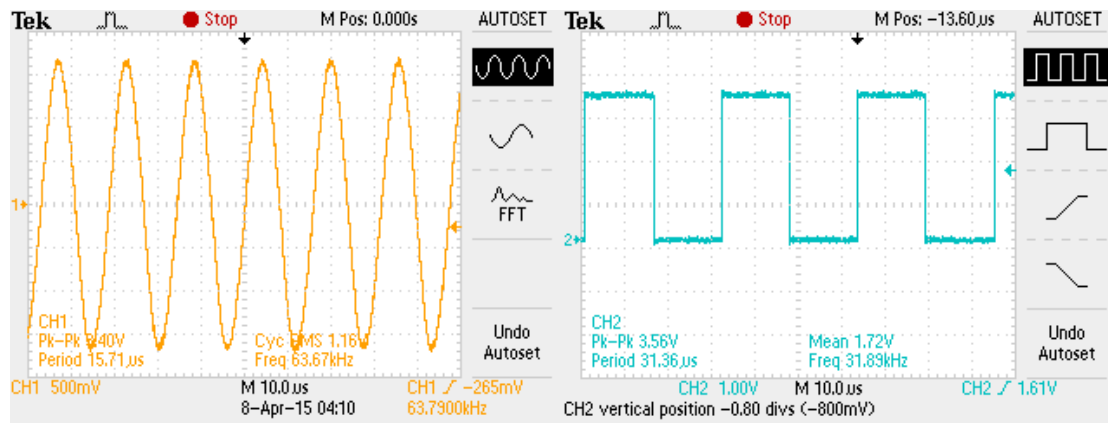


Figura 49. Visualización en el osciloscopio digital. a) Portadora. b) Señal de información.

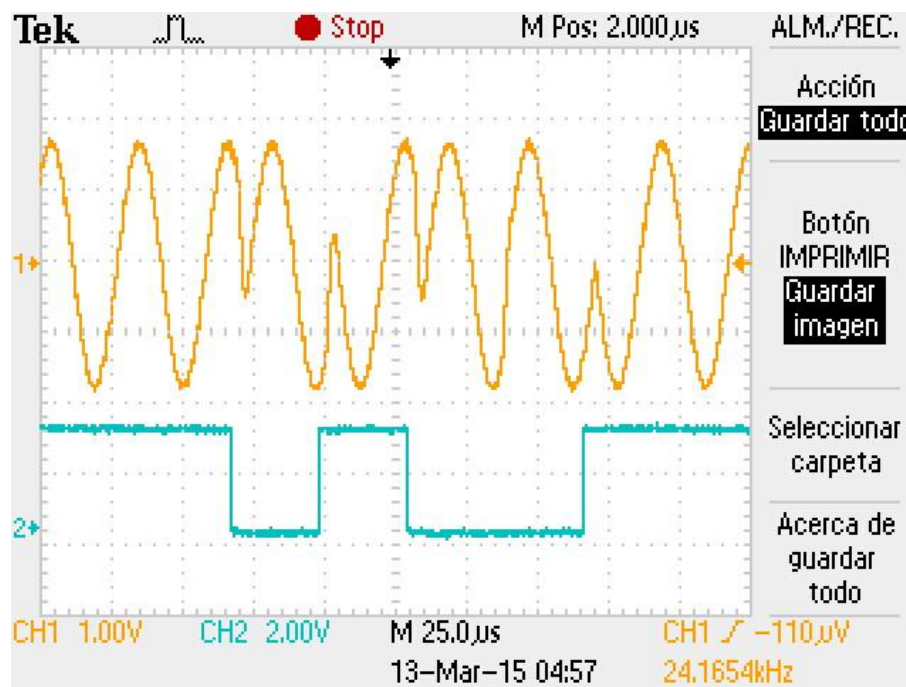


Figura 50. Señal Modulada en el osciloscopio.

Para la comprobación del ancho de banda máximo analizado en el Capítulo 3, se usó una señal alternada de 1's y 0's sin codificación como en la figura 51. Con una señal modulante de la mitad de frecuencia que la portadora.

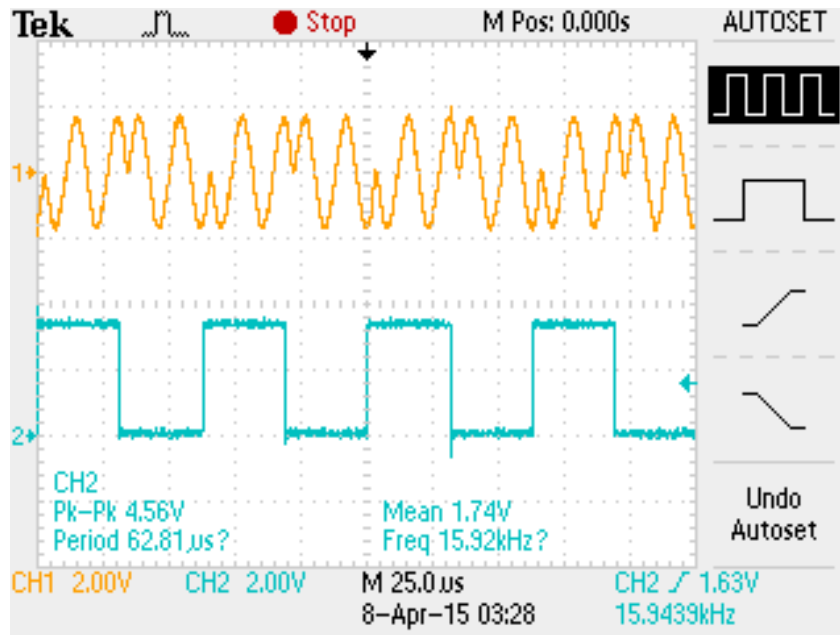


Figura 51. Señal binaria alternada y señal modulada.

En la figura 52 se visualiza a través del osciloscopio el ancho de banda máximo para la modulación implementada, mientras que en la figura 53 se muestran las bandas laterales, las cuales están desplazadas respecto a la frecuencia fundamental de la portadora. Este desplazamiento es igual a la frecuencia de la señal de información. Por lo que si la frecuencia de la portadora es de 62.5KHz y la frecuencia de la señal de información es de 31.25KHz, las bandas laterales estarán en 31.25KHz y en 93.75KHz.

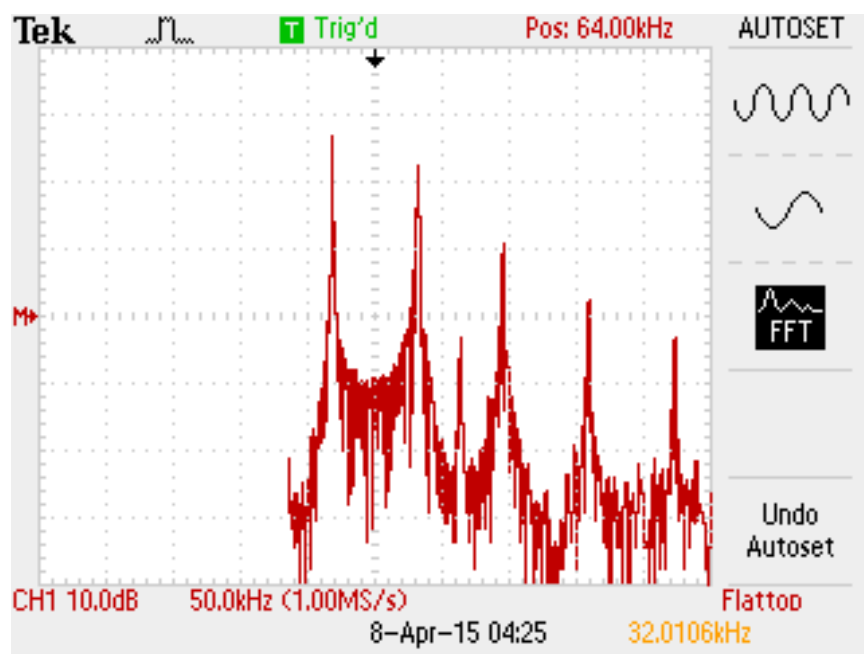


Figura 52. Espectro de la señal BPSK.

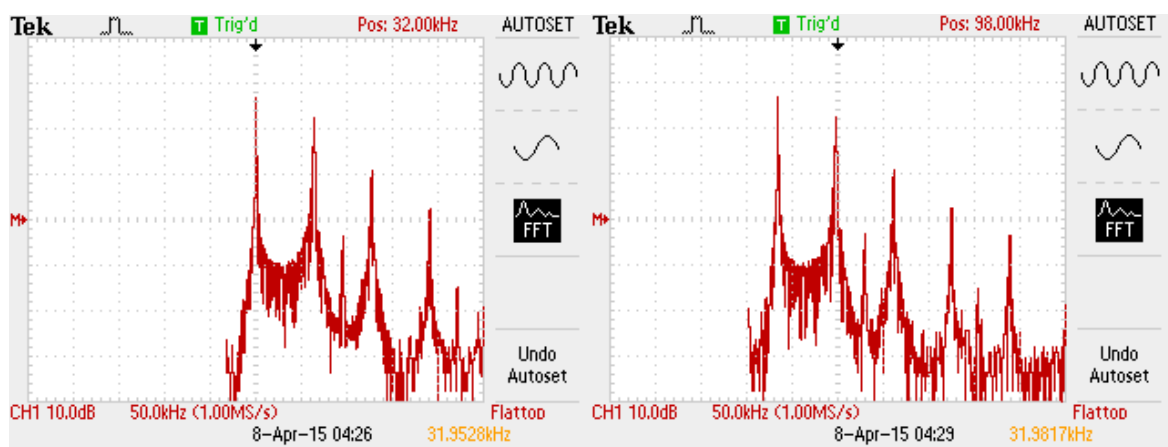


Figura 53. Bandas laterales del espectro BPSK.

Los recursos utilizados de la tarjeta de desarrollo para la implementación del transmisor digital se muestran en la tabla 6. Se puede apreciar que el uso de los recursos de la tarjeta son menores al 70%, por lo que se es capaz de agregar módulos adicionales.

Tabla 6. Utilización de recursos de la tarjeta de desarrollo.

Recursos	Usados	Porcentaje Utilizado
Flip-Flops	54	1%
LUT de 4 entradas	2494	26%
Segmentos ocupados	1478	31%
Buffer de entrada/salida	13	5%
Buffers de Reloj	2	8%

5.3 Conclusiones

En ésta tesis se presentó el desarrollo e implementación de un transmisor digital de datos utilizando un esquema de codificación para el control de errores mediante códigos de Hamming y un esquema de modulación BPSK. Se llevó a cabo un análisis de los sistemas de comunicaciones digitales, de los lenguajes descriptivos de hardware y de la tecnología FPGA para la definición de la metodología que abarcara el diseño completo del transmisor. La validación del diseño del transmisor se hizo por medio de simulaciones que comprobaran el funcionamiento antes de llevarlo a la etapa de implementación para optimizar los tiempos de desarrollo. La implementación se realizó en una tarjeta de desarrollo Spartan-3E Starter Kit que contiene un DAC para la salida de la señal modulada.

Por medio de éste trabajo se pudo comprobar que el desarrollo de sistemas de comunicaciones mediante tecnología FPGA es factible y que se pueden explotar los beneficios que ofrece, en especial la flexibilidad de diseño y el tiempo de desarrollo, gracias al concepto modular con el que se realiza la programación. En cuanto al funcionamiento del sistema en la tarjeta de desarrollo, los resultados son los esperados y la respuesta de la tarjeta comprueba que el paralelismo puro de las FPGA's da una ventaja en el desarrollo de sistemas digitales complejos.

Con la ayuda de herramientas como XSG y Matlab, el diseño y validación de prototipos para comunicaciones, se puede llevar a cabo de una manera flexible y rápida. Por lo que con el dominio de estas herramientas se pueden crear elementos complejos para los sistemas de comunicaciones con la finalidad de ser implementados en dispositivos programables.

El aprendizaje que tuve durante el desarrollo de ésta tesis abarca una gran cantidad de conocimientos, desde la oportunidad de involucrarme más con los dispositivos programables y la programación en VHDL, hasta reforzar mis conocimientos de acerca de las comunicaciones digitales, que como ingeniero en telecomunicaciones es una parte fundamental en mi formación.

5.4 Trabajo Futuro

Respecto al trabajo futuro, existen varios puntos que se pueden desarrollar en base a éste trabajo. El primero, sería incluir un circuito externo para poder propagar la señal modulada mediante una antena adecuada. Lo segundo sería, el desarrollo del receptor BPSK para completar el sistema de comunicaciones, añadiendo filtros digitales a la implementación en la FPGA. Utilizar una tarjeta de desarrollo con más recursos para implementar un esquema de multiplexación y combinarlo con la modulación para lograr un sistema de comunicaciones adecuado para una aplicación real.

REFERENCIAS

Aceves, M. A. & Ramos, J. M. 2012. Fundamentos de Sistemas Embebidos (Ed.). Asociación Mexicana de Mecatrónica A.C. México.

Bruce C., Crilly P. & Rutledge J. 2002. Sistemas de comunicación. (4ta Ed.) McGraw-Hill.

Chye, Y. H., Ain, M. F., & Zawawi, N. M. 2009. Design of BPSK transmitter using FPGA with DAC. In Communications (MICC), 2009 IEEE 9th Malaysia International Conference on (pp. 451-456). IEEE.

Couch II, L. W. 1998. Sistemas de comunicación digitales y analógicos (7ma Ed). Prentice Hall.

Digilent Inc. 2014.
<http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,897&Prod=NEXYS3&CFID=5644420&CFTOKEN=24ce718231f10800-1F4F5181-5056-0201-020930A9E23F6C12>.

Elamary, G., Chester, G., & Neasham, J. 2009. An analysis of wireless inductive coupling for High Data Rate biomedical telemetry using a new VHDL n-PSK modulator. In Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference on (pp. 211-214). IEEE.

Garcia, J., & Cumplido, R. 2005. On the design of an FPGA-Based OFDM modulator for IEEE 802.16-2004. In Reconfigurable Computing and FPGAs, 2005. ReConFig 2005. International Conference on (pp. 4-pp). IEEE.

García, I. J. M., & Pérez, I. E. J. 2006. Dispositivos lógicos programables (PLDs): diseño práctico de aplicaciones. RA-MA. p 3.

Giordano, A. A., & Levesque, A. H. 2015. Modeling of Digital Communication Systems Using SIMULINK. John Wiley & Sons.

Halsall, F., & García, R. E. 1998. Comunicación de datos, redes de computadores y sistemas abiertos. Addison-Wesley Iberoamericana.

Linear, Technology. 2004. LTC2604/LTC2614/LTC2624 datasheet.

Mandal, S., & Sarpeshkar, R. 2008. Power-efficient impedance-modulation wireless data links for biomedical implants. Biomedical Circuits and Systems, IEEE Transactions on, 2(4), 301-315.

Maxinez, D. G., & Jara, J. A. 2002. VHDL: el arte de programar sistemas digitales. Grupo Editorial Patria. p 15.

Moore, H. 2007. Matlab para ingenieros. L. M. C. Castillo (Ed.). Pearson Educación. p 3.

Rakesh, P. C. 2012. FPGA Design And Implementation Of Matrix Multiplier Architectures For Image And Signal Processing Applications. Pioneer Journal. <http://pioneerjournal.in/conferences/tech-knowledge/14th-national-conference/3783-fpga-design-and-implementation-of-matrix-multiplier-architectures-for-image-and-signal-processing-applications.html>.

Romero-Troncoso, R. D. J. 2007. Electrónica digital y lógica programable (Ed.). Universidad de Guanajuato, Mexico.

Sánchez, I. M. M., García, M. Á. P., Saldívar, P., Montiel, O., Sepúlveda, R., & Charles, R. H. 2006. VHDL—Modelado de circuitos lógicos digitales en Active HDL 5.1. Encuentro de Investigación en Ingeniería Eléctrica 2006. (pp. 169-174). ENINVIE.

Sghaier, A., Areibi, S., & Dony, B. 2008. A pipelined implementation of OFDM transmission on reconfigurable platforms. In Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on (pp. 000801-000804). IEEE.

Sundace Multiprocessor Technology. 2014
http://www.sundance.com/prod_info.php?board=SMT6041.

Tsutsui, A., Miyazaki, T., Yamada, K., & Ohta, N. 1995. Special purpose FPGA for high-speed digital telecommunication systems. In Computer Design: VLSI in Computers and Processors, 1995. ICCD'95. Proceedings. 1995 IEEE International Conference on (pp. 486-491). IEEE.

Van de Beek, R. C. H., Ciacci, M., Al-Kadi, G., Kompan, P., & Stark, M. (2012, June). A 13.56 Mbps PSK receiver for 13.56 MHz RFID applications. In Radio Frequency Integrated Circuits Symposium (RFIC), 2012 IEEE (pp. 239-242). IEEE.

Wayne, T. 2003. Sistemas de comunicaciones electrónicas (4ta Ed.). Pearson Educación.

Xilinx, Inc. 2011. Spartan-3E FPGA Starter Kit Board User Guide.