



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

Diseño de un prototipo de unidad de control, recepción y transmisión de señales para un sistema de resonancia magnética (RM).

TESIS
QUE PARA OPTAR POR EL GRADO DE
MAESTRÍA EN INGENIERÍA

PRESENTA:
ING. RICARDO ALBERTO SOLÍS ESTRADA

TUTOR O TUTORES PRINCIPALES
DR. PABLO ROBERTO PÉREZ ALCÁZAR, FACULTAD DE INGENIERÍA

CDMX. ENERO 2020



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

JURADO ASIGNADO:

Presidente: Dr. Moumtadi Fátima

Secretario: Dr. De la Rosa Nieves Saúl

Vocal: Dr. Pérez Alcázar Pablo Roberto

1^{er}. Suplente: Dr. Peña Cabrera Juan Mario

2^{do}. Suplente: Dr Hautefeuille Mathieu Christian A.

Lugar donde se realizó la tesis: Facultad de Ingeniería, UNAM, México CDMX.

TUTOR DE TESIS:

Dr. Pablo Roberto Pérez Alcázar.

FIRMA.

Agradecimientos

A mis padres que siempre me han dado su apoyo incondicional, a pesar de tener diferencias de ideas. Los amo infinitamente. Gracias por todo su apoyo. Que sigan a mi lado por muchos años más.

A mis 5 hermanos que me escucharon y fueron parte de esta etapa. En especial a mis hermanos el Ing. José Genaro Solís Estrada y el Ing. Antonio de Jesús Solís Estrada por estar en todo momento apoyándome en todas mis locuras, y las que faltan por hacer. Los amo.

A mi tutor el Dr. Pablo Pérez Alcázar gracias infinitas por todo su apoyo, el espacio, su tiempo, su atención, comprensión, y su inigualable paciencia.

A María Dolores Duran Lara que a pesar de las circunstancias siempre estuvo ahí para escuchar, darme ánimos, buenos consejos y lo más impórtate, su amor incondicional, gracias LOU.

A la madre de mi hija Briselda Coyote Tlazola por darme la oportunidad de ser padre de una hermosa nena. Te amo Emma Jatzibe Solís Tlazola. Y también por tantos años de amistad y cariño que me has brindado, así como tu familia.

A mis compañeros de generación que cada uno de ellos aportó en mí, experiencia, consejos, aventuras, su apoyo y su amistad. Gracias.

Al CONACYT por el apoyo económico brindado durante mis estudios de maestría.

Índice General.

Agradecimientos.

Índice General. I

Índice de Figuras. IV

Abreviaciones. VII

Resumen. IX

1 Introducción.

1.1. Antecedentes.....1

1.2. Justificación.....4

1.3. Objetivo.....4

1.4. Metodología.....5

1.5 Organización de esta tesis.....5

2. Marco Teórico.

2.1. Radios digitales aplicados a la RMN.....7

2.2. Transmisor.....8

2.2.1. Transmisor analógico.....8

2.2.2. Transmisor digital.....9

2.3. Receptor.....11

2.3.1. Receptor analógico.....11

2.3.2. Receptor digital.....13

2.4. Resonancia Magnética y gradientes magnéticos.....	15
2.5. Hardware.....	20
2.5.1. Circuitos Programables.....	21
2.5.2. Herramientas de Diseño.....	24
2.5.3 Convertidores Analógico – Digital (CAD) y Digital – Analógico (CDA).....	25
2.6. Muestreo.....	27
2.6.1. Sobremuestreo.....	30
2.6.2. Submuestreo o muestreo paso banda.....	32
3. Diseño del Sistema.	
3.1. Consideraciones de diseño.....	40
3.2. Arquitectura del sistema.....	41
3.3. Transmisor.....	41
3.4. Receptor.....	46
3.5 Control de pulsos y gradientes magnéticos.....	47
3.6 UART y Almacenamiento.....	51
4. Implementación y Pruebas.....	55
5. Conclusiones.....	73
A. Códigos Matlab.....	75

B. Códigos VHDL	76
Bibliografía	88

Índice de Figuras.

Figura 1. Esquema de un transmisor homodino.....	8
Figura 2. Esquema de un transmisor digital.....	9
Figura 3. Receptor analógico.....	11
Figura 4. Traslado de la señal de entrada a frecuencia intermedia.....	12
Figura 5. Receptor digital.....	12
Figura 6. Traslado de banda de señal a DC.....	15
Figura 7. Representación esquemática del proceso de RM.....	17
Figura 8. Relajación longitudinal y transversal.....	17
Figura 9. Secuencia de impulsos de RF.....	18
Figura 10. Secuencia típica de spin eco.....	19
Figura 11. Partes de un FPGA.....	21
Figura 12. Dibujo esquemático de un multiplicador de 4x4 Bits.....	22
Figura 13. Cuerpo de un programa en VHDL que describe un contador.....	24
Figura 14. Señal discreta y analógica.....	28
Figura 15. Representación de la zona Nyquist.....	29
Figura 16. Magnitud y frecuencia de la señal muestreada.....	29
Figura 17. Sobremuestreo de una señal: (A) Señal original, (B) Señal muestreada dos veces su frecuencia, (C) Señal submuestreada.....	31
Figura 18. Ejemplo de una señal paso banda: (a) muestreo síncrono con el pico máximo de la forma de onda, y (b) el caso general.....	33
Figura 19. Transformada de Fourier con alias no superpuesta de una forma de onda paso banda que se muestrea a menos del doble del componente de frecuencias más alta.....	35
Figura 20. Transformada de Fourier con aliasing de una forma de onda paso banda que se muestrea en varias frecuencias.....	38

Figura 21. El aspecto donde la señal de modulación cae a la señal banda base.....	39
Figura 22. Arquitectura del sistema.....	41
Figura 23. Esquema simplificado del DDS.....	42
Figura 24. Interfaz de usuario del DDS.....	43
Figura 25. Esquemático del Clocking Wizard.....	44
Figura 26. Asistente de programación de ADD/SUB.....	45
Figura 27. Asistente de programación del Clocking Wizard.....	45
Figura 28. IP Digital Down Converter.....	46
Figura 29. Diagrama complete del Sistema de recepción.....	47
Figura 30. Diagrama de tiempo para la secuencia espín – espín.....	48
Figura 31. Secuencia espín- eco.....	49
Figura 32. Secuencia eco gradiente.....	50
Figura 33. Diagrama de flujo de control de gradientes; eco – gradiente.....	51
Figura 34. Diagrama de bloques del UART.....	52
Figura 35. Diagrama de tiempo del UART.....	53
Figura 36. Diagrama de IP FIFO.....	53
Figura 37. Conversión de un programa VHDL a bloque.....	55
Figura 38. Diagrama de IP del transmisor.....	56
Figura 39. Simulación del sistema de transmisión.....	57
Figura 40. Muestreo de una señal senoidal de 20MHz con un Fm de 40MHz.....	58
Figura 41. Señal senoidal de 20MHz con Fm de 60MHz.....	58
Figura 42. Señal senoidal de 20MHz con Fm de 150MHz	59
Figura 43. Pulsos de 4MHz modulando una señal de 20MHZ.....	59
Figura 44. Secuencia espín – espín.....	60
Figura 45. Periodo entre pulsos de 90°.....	61

Figura 46. Señal senoidal modulada de 90°	61
Figura 47. Secuencia espín – eco.....	62
Figura 48. Pulso de 180° con un tiempo de 80ms (a), tiempo de DIL de 10ms (b).....	63
Figura 49. Señal de gradientes de izquierda a derecha gradiente Gz, Gy, Gz y Gx (señal azul).....	64
Figura 50. (a) Gradiente Gy, (b) y (c) Gradiente Gz, (d) Gradiente Gx.....	65
Figura 51. Interfaz de usuario del DDC.....	66
Figura 52. Sistema de recepción completo.....	67
Figura 53. Simulación de la IP FIFO.....	69
Figura 54. Interfaz de usuario de la IP BMG.....	70
Figura 55. Diagrama de tiempo de IP BMG.....	71
Figura 56. Diagrama de tiempo de UART.....	70

Abreviaciones.

2DFT	Transformada discreta de Fourier Bidimensional
ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
AM	Modulación en Amplitud
BB	Banda Base
CA	Corriente Alterna
CAD	Convertidor Analógico Digital
CDA	Convertidor Digital Analógico
CI	Circuito Integrado
CD	Corriente Directa
CLB	Configurable Logic Block
DDC	Digital Downconverters
DDS	Síntesis Digital Directa
DSP	Digital Signal Processing
DUC	Digital Upconverters
FM	Frecuencia Modulada
FIR	Respuesta al Impulso Finito
FIFO	First In First Out
FI	Frecuencia Intermedia
FFT	Fast Fourier Transform
FPGA	Field Programable Gate Array
FID	Decaimiento Libre de la Inducción
GUI	Graphical User Interface
IP	Intelectual Property
ISE	Integrated Synthesis Environment
IDE	Integrated Design Environment

LUT	Look Up Table
MRI	Magnetic Resonance Imaging
MRS	Espectroscopia de Resonancia Magnética
MSPS	Mega Samples Per Second
NCO	Osciladores Controlados Numéricamente
PDS	Procesador Digital de Señales
PM	Modulación en Fase
PLL	Phase Locked Loop
PLD	Programmable Logic Devices
RF	Radio Frequency
RMN	Resonancia Magnética Nuclear
RM	Resonancia Magnética
RTL	Register Transfer Level
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
SFDR	Spurious Free Dynamic Range
SOC	System On a Chip
TH	Muestreo y Retención
TR	Tiempo de Repetición
TE	Tiempo de Eco
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VHDL	Very High Speed Integrated Circuit

Resumen.

En un sistema de obtención de imágenes por la técnica de resonancia magnética se tienen diversos componentes, entre ellos un sistema de gradientes magnéticos, para la selección de capas y codificación espacial; un transmisor de radiofrecuencia (RF), para la excitación de los núcleos y la producción de impulsos que conformen las diversas secuencias a utilizar para la reconstrucción de las imágenes; y un receptor de RF para la detección y análisis de la señal de resonancia magnética RM, producida por la muestra bajo observación. Hasta hace algunos años estos sistemas eran completamente analógicos y poco versátiles, por lo que se ha buscado aprovechar los grandes avances que se han tenido en los últimos años en el desarrollo de la tecnología digital, en especial los dispositivos lógicos programables (FPGA - Arreglos de compuertas programables en campo), para aplicarlos en este tipo de sistemas, combinándolos con técnicas como el muestreo paso banda y algoritmos de procesamiento digital para obtener circuitos compactos, de bajo costo y versátiles. Es por esto que en este trabajo se propone el desarrollo de un sistema que además de transmitir y recibir señales de frecuencia y ancho de banda reconfigurables, también sea capaz de producir pulsos de excitación con características ajustables, para obtener las secuencias típicas utilizadas en los equipos de adquisición de imágenes por RM, tomando como base los dispositivos FPGA.

"La verdadera felicidad radica en la finalización del trabajo utilizando tu propio cerebro y habilidades",

Soichiro Honda

1.Introducción.

1.1Antecedentes.

El descubrimiento y desarrollo de los sistemas de obtención de imagen por resonancia magnética ha sido uno de los eventos más exitosos en la historia de la imagen médica; sin embargo, este desarrollo no ha sido continuo, ya que el descubrimiento del fenómeno de resonancia, el cual fue realizado de manera simultánea e independiente por Bloch [1] y Purcell [2], se realizó en 1946 y los primeros experimentos relacionados con imágenes se realizaron en los setentas por Lauterbur y Damadian. La resonancia magnética nuclear llegó a ser una técnica de análisis químico no destructiva cuando Proctor y Yu en 1950 [3] encontraron el efecto del corrimiento químico en los espectros de resonancia magnética nuclear (RMN). Además, por esos días, Gabillard [4] ya estaba experimentando con señales de RMN provenientes de muestras situadas en campos magnéticos no homogéneos y, a partir de dichos experimentos, sugería la posibilidad de localizar núcleos resonantes utilizando campos no homogéneos generados con gradientes de campo lineales. Sin embargo, el interés por determinar las posibilidades de aplicación de la RMN al diagnóstico médico surge en 1971 con el estudio de Damadian [5] de las diferencias en los tiempos de relajación T_1 y T_2 entre diferentes tejidos y en especial entre tejido normal y tejido canceroso. En 1972, Damadian patenta un dispositivo que podía medir selectivamente las señales de RMN provenientes de muestras de tejido utilizando una técnica de un solo punto [6]. En este dispositivo, al centro del imán permanente se crea una distribución tridimensional de campo magnético, simétrico, con forma de silla de montar, en el cual solamente en la vecindad del punto silla se tiene la inhomogenidad necesaria para dar lugar a una señal de resonancia medible. El dispositivo podía producir una imagen mediante una técnica de exploración punto a punto. Con esta técnica se realizó la primera exploración de pecho humano utilizando un imán superconductor de 0.05 teslas, un tiempo de exploración de 4.5 horas y con una resolución del orden de 1 cm.

Durante 1970 y 1974, varios grupos de investigación trabajaron en el uso de diversas técnicas para la obtención de imágenes por RMN. Uno de estos grupos fue el dirigido por Lauterburg,

el cual, en 1973 [7], presentó una imagen bidimensional de un “phantom” lleno de agua. La imagen se obtuvo a partir de un número de mediciones de señales de RMN obtenidas bajo la presencia de un gradiente lineal, cuya orientación se iba modificando. Utilizando los espectros obtenidos como proyecciones unidimensionales de la imagen en direcciones perpendiculares a la dirección del gradiente aplicado, el grupo pudo reconstruir la imagen RMN con un procedimiento de proyección-reconstrucción, similar al utilizado en exploradores de tomografía computarizada. Lauterburg llamó al método Zeugmatografía, debido a que en el proceso de generación de la imagen se utiliza la combinación de un campo magnético estático y campos de RF.

Una mejora importante en la obtención de imagen por RMN fue la introducción en 1974, por Garroway, Grannell y Mansfield [8], de la técnica de excitación selectiva de rodajas, que es utilizada en la actualidad, en la cual un gradiente de campo perpendicular al plano seleccionado se aplica durante un pulso de excitación. Mediante una combinación de pulsos de excitación y pulsos de gradiente ortogonales desarrollaron una técnica de exploración por líneas.

Sin embargo, el mayor impulso a las técnicas planares llegó con la aplicación de la técnica de imagen de Fourier bidimensional (2DFT) directa, reportada en 1975 por Kumar, Welti y Ernst [9]. En esta técnica se agrega el gradiente de campo para hacer la distribución espacial en componentes de Fourier.

Las aplicaciones biomédicas de la (RMN) son dos: imágenes de resonancia magnética (MRI) y espectroscopia de resonancia magnética (MRS). Las aplicaciones de la MRS como herramienta de investigación son extremadamente diversas, abarcando estudios en células aisladas, fluidos corporales y órganos perfundidos con altas intensidades de campo magnético en un entorno experimental, basado en laboratorio, y también estudios *in vivo* que utilizan sistemas clínicos de RM. Los escáneres de resonancia magnética de cuerpo entero se han utilizado para estudiar el metabolismo de regiones bien definidas del cuerpo humano, proporcionando una "ventana metabólica" no invasiva en una amplia gama de procesos bioquímicos en el cuerpo, incluida la composición y función del ser humano, así como órganos *in vivo*. Los desarrollos clínicos de MRS han explotado muchos de los avances en RMN, en las intensidades de campo magnético que se usan actualmente (típicamente 1.5–3.0

T) y el uso de gradientes de campo magnético. La sensibilidad y la resolución espacial de la MRS es un factor limitante *in vivo*, pero la utilización paralela de la espectroscopia de RM *in vitro* de extractos de tejidos, fluidos corporales y líneas celulares a intensidades de campo magnético mucho más altas (típicamente 11.7–14.1 T) permite una interpretación más definitiva de los datos *in vivo* [10].

De lo señalado hasta ahora, en la obtención de imágenes por RMN es importante tener un imán permanente, para generar un campo magnético estático, conocido como campo principal; bobinas de gradiente que generan campos con inhomogeneidades lineales en tres direcciones mutuamente ortogonales (X, Y, Z); una bobina de RF para excitación de los sistemas de espín; y la electrónica necesaria para transmitir pulsos de RF, recibir la respuesta de dichos sistemas de espín y generar los pulsos de activación de los gradientes en los instantes adecuados. La activación de los gradientes y la generación de los pulsos de excitación se realiza de acuerdo a cierto tipo de secuencias ya establecidas, como: secuencia de recuperación de la excitación, secuencia espín-eco (secuencia de ecos de los espines), secuencia espín-eco con ecos repetidos, secuencia de recuperación de la inversión, etc [11]. Hasta hace algunos años todos los sistemas de obtención de imágenes de RMN utilizaban circuitos electrónicos analógicos para la realización de las operaciones de transmisión de pulsos de RF de excitación y la recepción de la información de RF generada por los sistemas de espín de la región del cuerpo siendo explorada. Los receptores utilizados convertían la señal de RF inducida en la bobina en datos complejos en banda base que son adecuados para la reconstrucción de la imagen. Existen diversos tipos de receptores, sin embargo, el receptor heterodino convencional utiliza un oscilador local de referencia para llevar la señal de RF a una frecuencia intermedia para la adquisición. Este proceso introducía señales, imágenes y ruido indeseado en el ancho de banda de la imagen. Para evitar estos problemas, desde principios de 1990 se ha estado trabajando en la introducción de técnicas digitales, primero para procesar las señales de frecuencia intermedia y posteriormente, con los avances logrados en el desarrollo de convertidores de analógico a digital (ADC), de gran ancho de banda y alta resolución, y circuitos integrados de RF digitales, lograr receptores digitales para muestreo directo de señales de RF; implementando así un receptor digital directo. El muestreo directo de señales de RF permite llevar las técnicas lo más cerca posible a la bobina de recepción, eliminando la necesidad de los mezcladores, oscilador local y filtros analógicos, con lo cual

se busca tener un mejor desempeño y mayor versatilidad [12]. El problema de utilizar circuitos integrados digitales de RF es que no permiten mucha flexibilidad, por lo que actualmente se proponen soluciones con base en FPGA [13]. Para incrementar el rango de aplicaciones de la recepción digital directa, además de utilizar el sobremuestreo, se ha propuesto el uso del muestreo con conversión directa a baja frecuencia o submuestreo [14], aprovechando la capacidad de los CAD específicamente diseñados para este tipo de trabajo [14, 15], los cuales tienen un ancho de banda de entrada mucho mayor que la frecuencia de muestreo utilizada. En algunos sistemas de RMN actuales se combinan los procesadores digitales de señales con los FPGAs, utilizando estos últimos para la generación de señales de control y programación de secuencias de pulsos [16].

1.2. Justificación.

En el Departamento de Ingeniería Electrónica se ha trabajado en el desarrollo de receptores digitales para sistemas que trabajan con señales en el rango de las RF, tales como sensores que trabajan con ondas acústicas superficiales y sistemas de RMN, pero en dichos sistemas no se ha incluido dentro del mismo FPGA el programador de pulsos o generador de secuencias para estos últimos. Por lo tanto, se trabajará en esta dirección, en el estudio de algunas variantes de construcción del sintetizador de señales de RF y en el estudio del comportamiento de la fase en las señales recibidas, ya que esta información es crucial en la reconstrucción de las imágenes 2DFT.

1.3. Objetivos.

Diseñar y construir un prototipo de unidad de control que, además de transmitir y recibir señales de frecuencia y ancho de banda reconfigurables, también sea capaz de producir pulsos de excitación con características ajustables, para obtener las secuencias típicas utilizadas en los equipos de adquisición de imágenes por RMN, tomando como base los dispositivos FPGA y, además de las técnicas de sobremuestreo y muestreo Nyquist, utilice el muestreo de conversión directa a baja frecuencia (submuestreo o muestreo pasobanda).

1.4. Metodología.

Para cumplir con el objetivo propuesto durante el desarrollo de este trabajo se alcanzaran las siguientes metas:

- 1) Se revisó la bibliografía relacionada con la aplicación de las técnicas digitales en sistemas de RMN.
- 2) Se diseñó y construyó un sintetizador digital que permitió generar las señales para realizar el proceso de detección de las señales producidas por la muestra bajo estudio y generar los pulsos de RF de excitación con frecuencias de hasta los 20MHz, con base en un FPGA.
- 3) Se diseñó y construyó un detector digital en cuadratura operando a una frecuencia de hasta 100MHz, tomando como base el diseño realizado en el punto 2.
- 4) Se diseñó el programador de pulsos que permitió generar las principales secuencias utilizadas en un sistema de RMN. Además de la activación de los gradientes.
- 5) Se evaluó el desempeño del sistema.
- 6) Se realizaron las modificaciones necesarias para lograr un mejor desempeño del sistema.

1.5 Organización de esta tesis.

En este primer capítulo se ha presentado el objetivo de esta tesis, así como algunos de los trabajos realizados sobre radios digitales orientados a la adquisición de datos para la obtención de imágenes médicas, con el propósito de mostrar la importancia de la misma. Se realiza una breve descripción de los temas a tratar.

En el capítulo dos se describen los fundamentos para el desarrollo de este proyecto. La teoría sobre los radios digitales, conceptos básicos de los transmisores y receptores analógicos, las diferentes técnicas de adquisición de datos para la RMN y la introducción de circuitos programables y los diferentes instrumentos para el uso de estas.

En el tercer capítulo se detalla el diseño del sistema, las consideraciones a realizar tomando en cuenta los aspectos teóricos revisados, la estructura de los sistemas que componen un radio digital y la aplicación en el FPGA.

El cuarto capítulo describe la implementación del sistema, así como como las pruebas aplicadas a ésta y la solución de los problemas que surgieron en el proceso del diseño.

En el quinto capítulo se presentan y analizan los resultados logrados, asimismo se mencionan trabajos a futuro.

2.Marco Teórico.

2.1. Radios digitales aplicados a la RMN.

El concepto de software radio fue introducido por primera vez por el ejército de los estados unidos en los años 90's. El primer proyecto fue el "Speakeasy" que abrió un nuevo campo de investigación en la industria de las telecomunicaciones, militar y civil. Mitola expuso el concepto de software radio en 1991 [17].

Los SDR (Software Defined Radio) han revolucionado los sistemas electrónicos en diversas aplicaciones incluyendo las comunicaciones, adquisición de datos y el área de procesamiento de señales. El SDR se compone de dos bloques fundamentales DDC (Digital Down converter) y DUC (Digital Up converter), estos componentes permiten remplazar los diseños de transmisores y receptores analógicos además que ofrecen significativas mejoras en rendimiento, densidad y costo [18].

El propósito de un SDR es digitalizar la señal de RF (radio frecuencia) lo más cerca posible de la antena, según las características y capacidades de los ADC (analog to digital converter). Posteriormente la señal será procesada en una computadora u otro dispositivo de procesamiento digital bajo control de un software, en la cual se remplaza las funciones realizadas antes, mediante componentes analógicos, en funciones ejecutadas por un procesador digital (modulación, control automático de ganancia, filtrado, etc). Los dispositivos más usados son los DSP's (Procesadores Digitales de Señales) y los FPGA's Este tipo de tecnología ofrece gran flexibilidad, reconfiguración, adaptabilidad y durabilidad, es decir, ofrecen una vida productiva larga [19].

Una de las aplicaciones de los SDR es en los sistemas de RMN, en los que permitan generar secuencias de radio frecuencia, así como también recibir señales emitidas por los tejidos. La RMN es una técnica no invasiva que permite por medio de pulsos electromagnéticos, en presencia de campos estáticos, obtener imágenes de tejido humano.

2.2 Transmisor.

2.2.1 Transmisor analógico.

Un transmisor, en un sistema de comunicaciones, es un conjunto de uno o más dispositivos o circuitos electrónicos que convierte la información de la fuente original en una señal que se presta más a su transmisión a través de determinado medio de transmisión. El medio de transmisión transporta las señales desde el transmisor hasta el receptor y puede ser tan sencillo como un par de conductores de cobre que propaguen las señales en forma de flujo de corriente eléctrica. Los radiotransmisores se clasifican según su ancho de banda, esquema de modulación y tipo de información [20].

Los sistemas electrónicos de comunicaciones se pueden diseñar para manejar el transmisor solo en una dirección; en ambas direcciones, solo en una a la vez; o en ambas direcciones al mismo tiempo. A estos se les llama modos de transmisión. Hay cuatro modos de transmisión posibles: símplex, semidúplex, dúplex y dúplex/dúplex [20]. Para este caso de estudio, se diseña un sistema símplex que permite transmitir la señal con la que se excitará a las moléculas del tejido humano para modificar su momento magnético.

Los transmisores de RF también se pueden clasificar en: transmisores analógicos y digitales. También se pueden clasificar por su tipo de modulación empleada:

- Transmisores con modulación en amplitud (AM y ASK). Aquí se distinguen, a su vez dos tipos: bajo nivel y alto nivel.
- Transmisores de doble banda lateral (DBL).
- Transmisores de banda lateral única (BLU).
- Transmisores con modulación en frecuencia (FM Y FSK).
- Transmisores con modulación en fase (PM y PSK).

Hay que considerar que existen varios tipos de sistemas de transmisión tomando en cuenta la modulación, pues emplean más de un tipo (tal es el caso de multiplex de frecuencia).

También existen autores que clasifican transmisores de RF por la banda de frecuencias que se emplean [21].

Por otro lado, la clasificación más interesante es aquella que se refiere a la forma en la que se genera la señal modulada:

- Transmisores homodinos. Son aquellos que modulan directamente la frecuencia de la señal portadora. Se realiza la modulación sobre la propia frecuencia de la emisión.

- Transmisores heterodinos. En ellos se modula empleando una frecuencia auxiliar intermedia y, posteriormente, mediante la mezcla con un tono de frecuencia mayor del oscilador, se alcanza la frecuencia final de emisión. La modulación se realiza sobre una frecuencia intermedia [21].

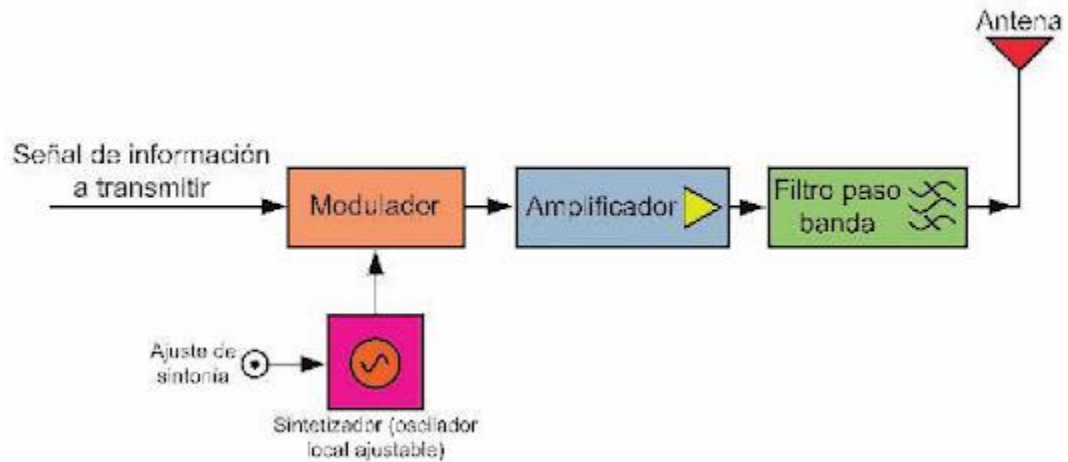


Figura 1. Esquema de un transmisor homodino [21].

2.2.2 Transmisor Digital.

Las telecomunicaciones modernas se centran en los problemas relacionados con la transmisión de grandes volúmenes de información a través de largas distancias sin pérdidas debidas al ruido y las interferencias. Los componentes básicos de un sistema de telecomunicaciones digitales moderno deben ser capaces de transmitir señales de voz, datos, radio y televisión.

La transmisión digital se emplea para lograr una alta confiabilidad y porque el costo de los sistemas de conmutación digital es mucho más bajo que el costo de los sistemas analógicos. Sin embargo, para utilizar la transmisión digital, las señales analógicas que forman la mayoría de las comunicaciones de voz, radio y televisión deben someterse a un proceso de conversión de analógico a digital. (En la transmisión de datos, este paso se omite porque las señales ya están en formato digital; sin embargo, la mayoría de las comunicaciones de televisión, radio y voz utilizan el sistema analógico y deben digitalizarse). En muchos casos, la señal digitalizada se pasa a través de un codificador de origen, que emplea varias fórmulas para reducir la información binaria redundante. Después de la codificación de la fuente, la señal digitalizada se procesa en un codificador de canal, que introduce información redundante que permite detectar y corregir errores. La señal codificada se hace adecuada para la transmisión

por modulación en una onda portadora y puede formar parte de una señal más grande en un proceso conocido como multiplexación. La señal multiplexada se envía a un canal de transmisión de acceso múltiple. Después de la transmisión, el proceso anterior se invierte en el extremo receptor y se extrae la información [22].

La transmisión digital se basa principalmente en un sistema de síntesis digital directa DDS. Los DDS, o los osciladores controlados numéricamente (NCO), son componentes importantes en muchos sistemas de comunicación digital. Los sintetizadores de cuadratura como el DDS se utilizan para construir convertidores digitales ascendentes (DUC) y descendentes (DDC), demoduladores e implementar varios tipos de esquemas de modulación, incluidos PSK (cambio de fase), FSK (cambio de frecuencia) y MSK (cambio de mínimo). Un método común, utilizado por el DDS para generar digitalmente una senoide compleja o de valor real, emplea un esquema de tabla de búsqueda (Lookup Table). La tabla de búsqueda almacena muestras de una onda senoidal, Figura 2. Se utiliza un integrador digital para generar un argumento de fase adecuado que la tabla de búsqueda asigna a la forma de onda de salida deseada. Una interfaz de usuario simple acepta parámetros a nivel del sistema, como la frecuencia de salida deseada y la supresión de estímulo de las formas de onda generadas [23].

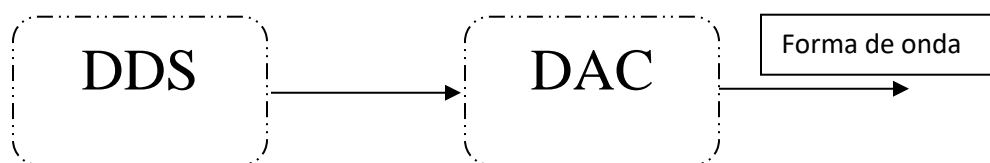


Figura 2. Esquema de un transmisor digital.

El transmisor o generador de señales de radiofrecuencia necesarias para excitar a los protones, debe generar los pulsos durante intervalos de muy corta duración, establecidos por secuencias de pulsos con una frecuencia igual a la frecuencia de precesión de los núcleos que se pretenden excitar y con una amplitud de pico a pico de varios cientos de voltios [24].

Entonces, el equipo emite una secuencia de pulsos de radiofrecuencia para cambiar la dirección del campo magnético de los protones al transmitirles energía. Solamente pulsos con la misma frecuencia que la de precesión de los protones pueden transmitirles energía. La frecuencia de precesión se calcula con la ecuación de Larmor. Al enviar pulsos de radiofrecuencia con la misma frecuencia de precesión calculada con la ecuación de Larmor estamos transfiriendo energía a los protones (fenómeno de resonancia), de tal forma que la magnetización longitudinal disminuye y aparece una magnetización transversal. Los pulsos de radiofrecuencia se diferencian por la cantidad de energía que transfieren a los protones, es

decir, por el grado de relajación provocado. Así, los pulsos se definen por la capacidad de desviar la magnetización, debido a los momentos magnéticos de los protones, desde su posición inicial en un cierto ángulo [24].

Una secuencia RM consta, generalmente, de varios pulsos de radiofrecuencia que además pueden ser diferentes entre sí. La diferencia entre unas secuencias y otras depende del tipo de pulsos de radiofrecuencia utilizados y del tiempo que hay entre ellos [24].

Como ya se mencionó anteriormente, el sistema de RMN requiere de una secuencia de pulsos que modifique el momento magnético de los núcleos de hidrogeno. Esta secuencia se logra con un trasmisor por medio de un FPGA.

2.3 Receptor.

2.3.1 Receptor analógico.

El receptor de radio heterodino convencional, que se muestra en la Figura 3, ha estado en uso durante casi un siglo. A continuación, se realiza una revisión de la estructura del receptor analógico para después realizar una comparación con un receptor digital.

Primero se amplifica la señal de RF de la antena, generalmente con una etapa de RF sintonizada, para trabajar únicamente con la banda de frecuencias de interés. Esta señal de RF amplificada se alimenta a un mezclador. La otra entrada al mezclador proviene del oscilador local cuya frecuencia está determinada por el control de sintonización de la radio. El mezclador traduce la señal de entrada deseada a FI pasa banda (frecuencia intermedia) como se muestra en la Figura 4. La etapa de FI es un amplificador paso banda que solo deja pasar la señal de interés. Las frecuencias centrales para las etapas de FI en transmisiones comerciales de AM y FM son 455KHz y 10.7MHz, respectivamente [18].

El demodulador recupera la señal de modulación original de la salida FI utilizando uno de varios esquemas diferentes. Por ejemplo, AM usa un detector de envolvente y FM usa discriminador de frecuencia. En una radio domestica típica, la salida demodulada se alimenta a un amplificador de potencia de audio que controla un altavoz [18].

El mezclador realiza una multiplicación analógica de las dos señales de entrada y genera una señal con frecuencia igual a la diferencia de dichas señales de entrada.

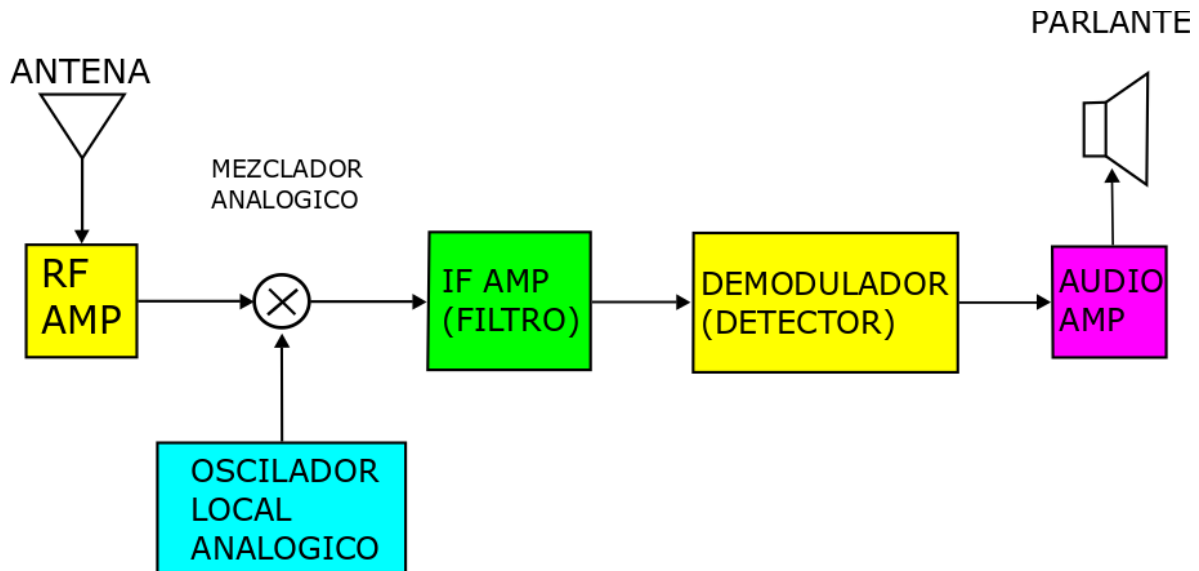


Figura 3. Receptor analógico [18].

La frecuencia del oscilador local se establece de manera que la diferencia entre ella y la señal de entrada deseada sea igual a la FI. Por ejemplo, si se desea recibir una estación de FM a 100.7 MHz y la IF es de 10.7MHz, entonces se debe sintonizar el oscilador local a una frecuencia de: $100.7 - 10.7 = 90\text{MHz}$ [18].

Esto se denomina “conversión descendente” o “traslación”, porque el mezclador desplaza una señal de frecuencia alta a una frecuencia más baja.

La etapa FI actúa como un filtro de banda estrecha que solo pasa una porción de la entrada RF traducida. El ancho de banda de la etapa de FI es igual al ancho de banda de la señal que se está intentando recibir [18].

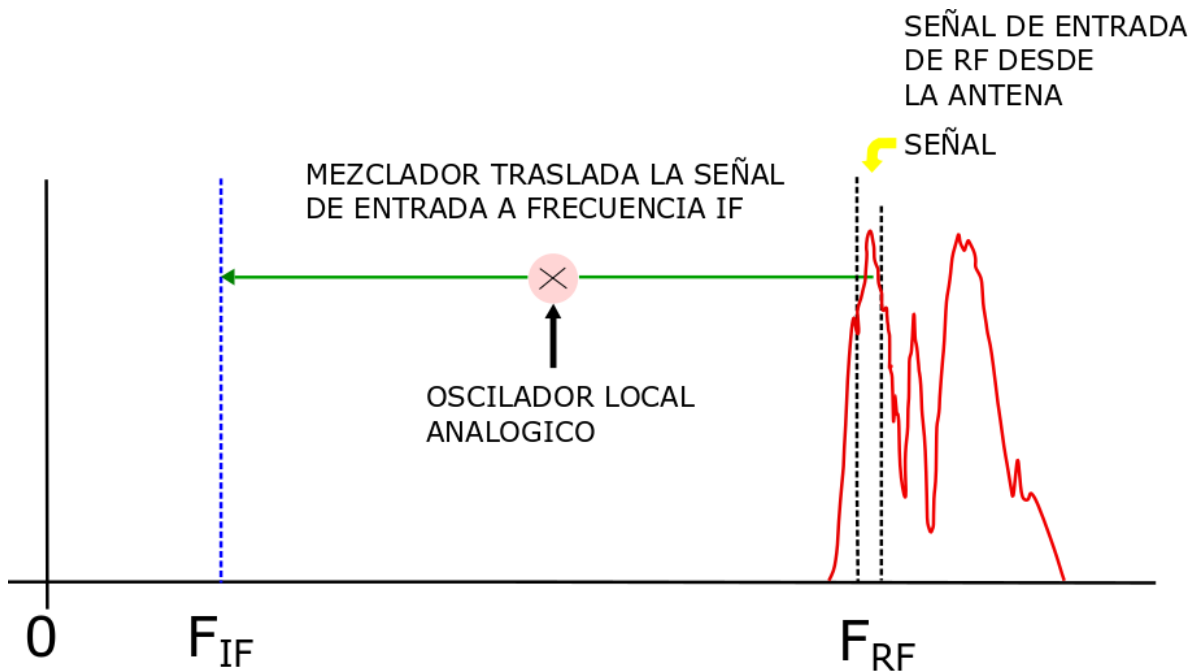


Figura 4. Traslado de la señal de entrada a frecuencia intermedia [18].

2.3.2 Receptor Digital.

La Figura 5 muestra un diagrama de bloques de un receptor de radio definido por software. El sintonizador de RF convierte las señales de RF analógicas a frecuencias de FI analógicas, al igual que las tres primeras etapas del receptor analógico.

El convertidor A/D digitaliza la señal FI y la convierte en muestras digitales. Estas muestras se envían a la siguiente etapa, que es el convertidor descendente digital (DDC) que se muestra dentro de las líneas punteadas.

El convertidor descendente digital suele ser un único chip monolítico o FPGA IP (intelectual property), y es una parte clave del sistema SDR [18].

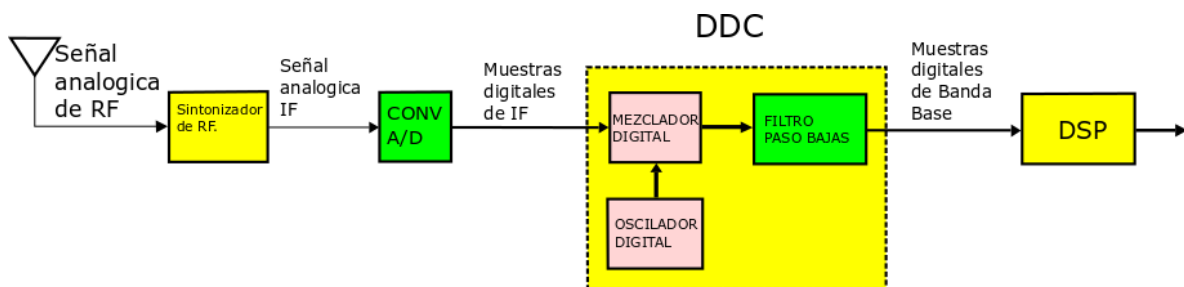


Figura 5. Receptor digital [18].

El sistema DDC está compuesto por tres partes:

- Mezclador digital.
- Oscilador local digital
- Un filtro FIR paso bajas.

El mezclador digital y el oscilador local traducen las muestras digitales de FI a banda base. El filtro paso bajo FIR (Respuesta al Impulso Finito) limita el ancho de banda de la señal y actúa como un filtro decimador paso bajo.

Las muestras en banda base digital se envían a un bloque etiquetado como DSP que realiza tareas tales como decodificación, filtrado y decimación. Comúnmente, estas tareas se han cubierto con circuitos integrados CI dedicados a aplicaciones específicas (ASIC- Application Specific Integrated Circuit) y DSP programables.

En la salida del mezclador, las señales de banda ancha y alta frecuencia de la entrada al A/D (Figura 6) se han trasladado a banda base o 0Hz mediante la detección en cuadratura I (in phase) y Q (quadrature) con un cambio de frecuencia igual a la frecuencia del oscilador local.

Esto es similar al mezclador en un receptor analógico, excepto que la mezcla se realizó hasta una frecuencia de FI. Aquí la representación compleja de la señal permite ir directamente a CD.

En efecto, el espectro de la señal de RF de banda ancha se puede “deslizar” alrededor de 0Hz, izquierda y derecha, simplemente sintonizando el oscilador local. Tenga en cuenta que las bandas laterales superior e inferior se conservan [18].

Por otra parte, el receptor aplicado en la resonancia magnética nuclear es un elemento importante, que se encarga de recibir la señal de los núcleos de hidrogeno cuando estos regresan a su momento angular inherente. Esta señal será digitalizada por un CAD y procesada por medio de un PDS. El FPGA está encargado de procesar la señal por medio de un DDC que permita filtrarla y amplificarla. Posteriormente, la señal adquirida podrá ser usada para obtener la imagen del tejido deseado por medio del espectro.

El receptor del equipo de RM capta los cambios en la magnetización, producida por los protones después de haber recibido la energía de los pulsos de radiofrecuencia. Es decir, capta la relajación longitudinal y transversal de los protones. La información obtenida directamente de la relajación de los protones se denomina matriz de datos crudos (raw data matrix) o espacio k, que no es más que una reja de puntos correspondientes a las amplitudes de las componentes en frecuencia que pueden ser manipuladas por la transformada inversa de Fourier para obtener una imagen interpretable [24].

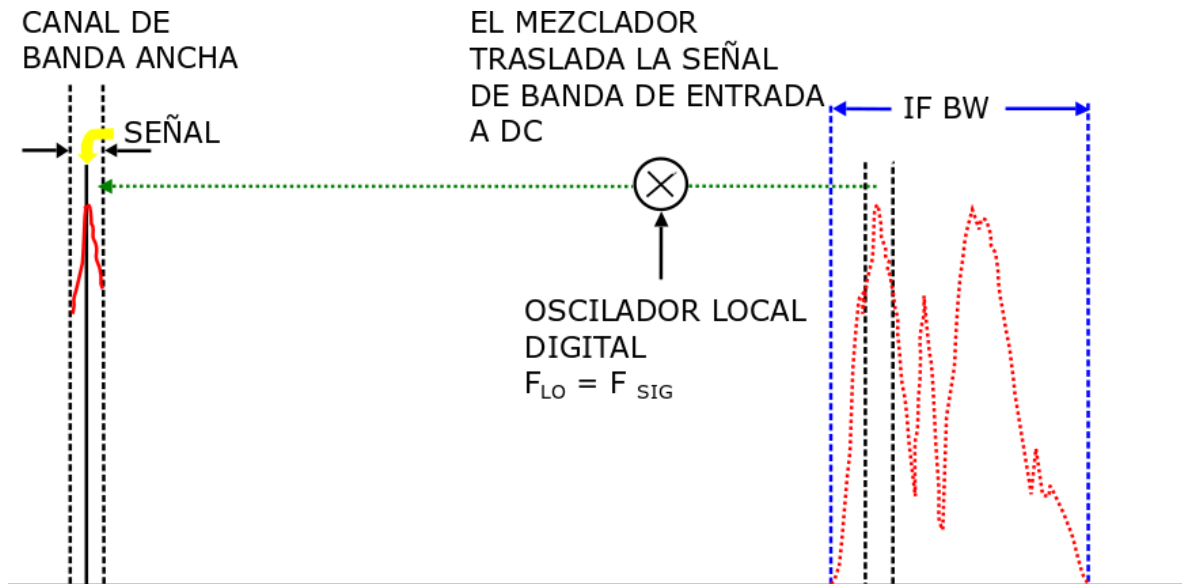


Figura 6. Traslado de banda de señal a DC [18].

Se utiliza un IP DDC de XILINX, programado en VIVADO, para diseñar el receptor digital. Un convertidor digital descendente (DDC) es el componente de contraparte del DUC y, por lo tanto, es igualmente importante como componente en los mismos sistemas de aplicación. Su función es trasladar una señal paso banda, que comprende una o más portadoras de radio o de frecuencia intermedia (RF o IF) a uno o más canales en banda base para demodulación e interpretación. Esto se logra realizando: el mezclado para cambiar el espectro de la señal de las frecuencias portadoras seleccionadas a la banda base; el diezmado para reducir la velocidad de muestreo y el filtrado para eliminar canales adyacentes, minimizar el alias y maximizar la relación señal/ruido recibida (SNR). La señal de entrada al DDC tiene una frecuencia de muestreo relativamente alta, generalmente, la frecuencia de muestreo de salida de un convertidor analógico a digital (CAD) que muestrea la señal detectada (a menudo después de la traslación en frecuencia analógica y el preprocesamiento), mientras que la salida es una tasa mucho más baja, por ejemplo, la tasa de símbolos de un sistema de comunicaciones digitales para la demodulación [25].

2.4 Resonancia magnética nuclear y gradientes magnéticos.

Las áreas principales de la resonancia magnética nuclear son: la imagenología y la espectroscopia. La espectroscopia se utiliza para la determinación de la estructura molecular y la imagenología para la visualización morfológica y la discriminación, con base en las propiedades nucleares, para ver partes de la materia que sean de interés. Por ejemplo, con la

imagenología se pueden ver estructuras que tengan como base el hidrogeno, que es el componente más importante en los tejidos del cuerpo humano [26].

El proceso de detección se basa en la relajación de los momentos magnéticos en un campo que es aplicado externamente. Se aplica un campo magnético principal obteniendo una magnitud M_0 en dirección M_z para alinear los momentos magnéticos de los núcleos. Al mismo tiempo, los momentos magnéticos precesan en dirección al campo principal M_0 . Esto se conoce como precesión de Larmor. La frecuencia de este movimiento de precesión ω viene dada por la ecuación 1, llamada ecuación de Larmor.

$$\omega = \gamma B_0 \quad (1)$$

Donde ω es la frecuencia angular de Larmor (MHz); γ es la relación giromagnética (MHz/T), que viene dada por la relación entre las propiedades magnéticas y mecánicas del núcleo; y B_0 es la fuerza del campo magnético en Tesla (T). La resonancia se producirá cuando una onda electromagnética con la frecuencia apropiada (igual a la frecuencia de Larmor o frecuencia natural) alcance los núcleos. Este pulso genera una reorientación de los momentos magnéticos (Figura 7) debido a la absorción de energía y pasarán del estado de menor energía al estado de energía más alta.

La Tabla 2.1 muestra algunas frecuencias de excitación típicas en RM para núcleos de hidrogeno, flúor, fosforo y sodio ante diferentes intensidades de campo magnético [27].

Intensidad de campo magnético (T)	Frecuencia (MHz)			
	¹ H	¹⁹ F	³¹ P	²³ Na
0.1	4.3	4	1.7	1.1
0.3	12.8	12	5.1	3.4
0.5	21.3	20	8.6	5.6
1.0	42.6	40	17.2	11.3
1.5	63.9	60	25.9	16.9
2.0	85.2	80	34.5	22.5
3.0	127.8	120	51.8	33.8
4.7	200	188	81	52.9
9.4	400	376	162	105.8
11.7	500	470	203	131.6

Tabla 2.1. Dependencia entre la intensidad de campo y la frecuencia.

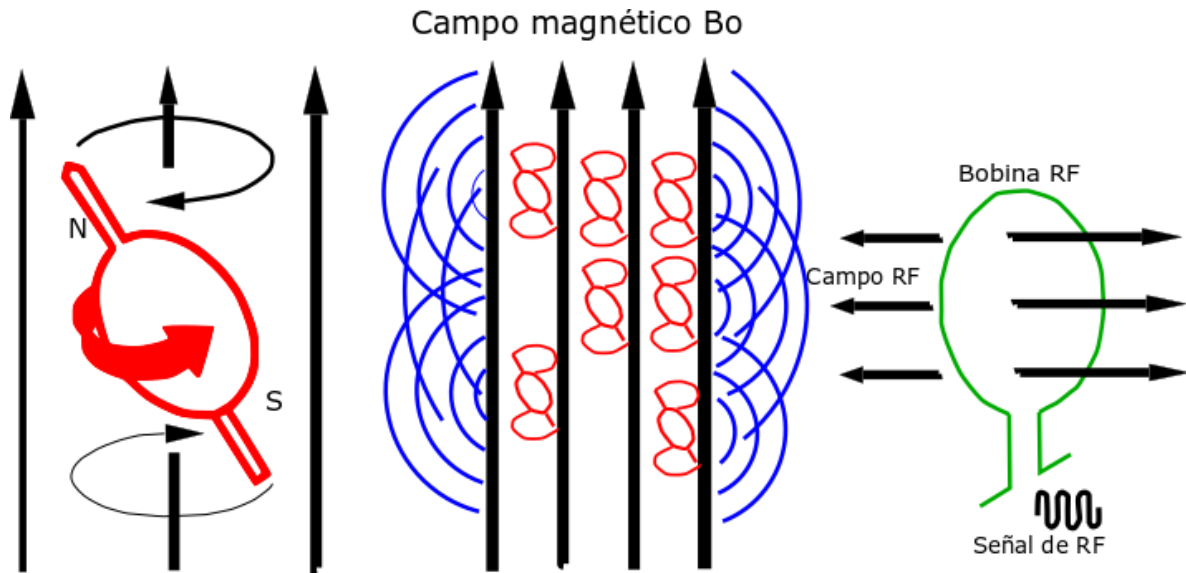


Figura 7. Representación esquemática del proceso de RM [11].

Al dejar de aplicar el pulso de RF, los momentos magnéticos de los núcleos se relajan a sus orientaciones originales. De esta manera se identifican los diferentes tiempos de relajación: longitudinal y transversal (Figura 8). Estos tiempos permiten adquirir las señales emitidas por los núcleos, ser procesadas y generar las imágenes de la sección del tejido seleccionado.

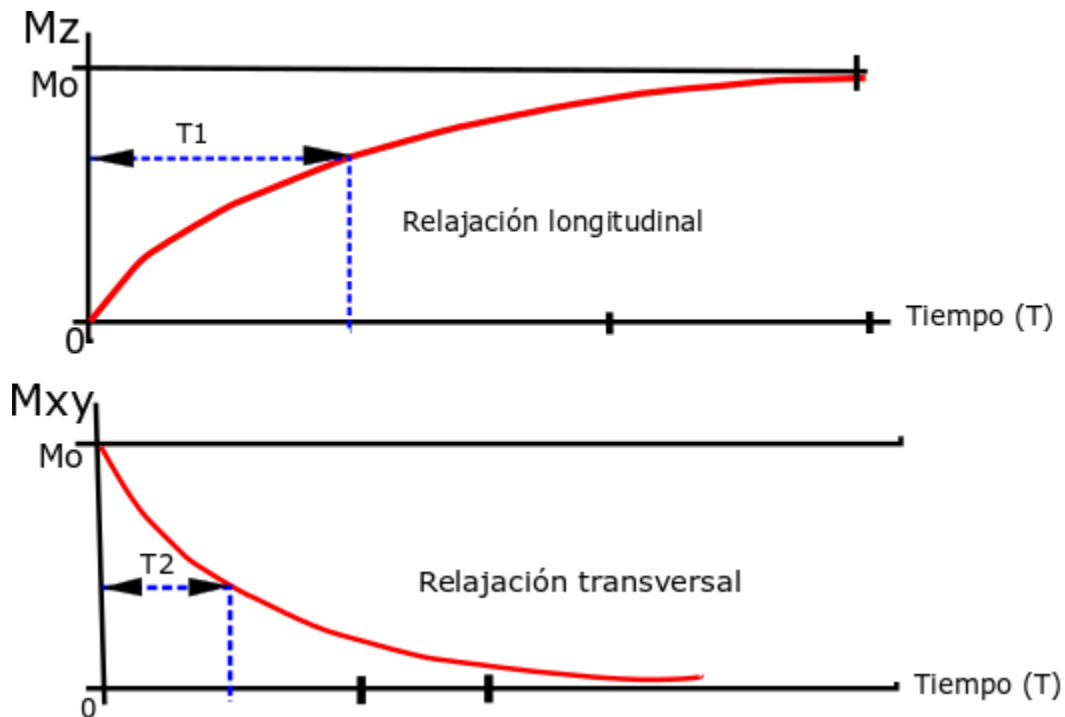


Figura 8. Relajación longitudinal y trasversal [11].

El tiempo requerido para que la señal emitida desde los núcleos de hidrogeno caiga a cero viene dado por la constante de tiempo T_2 . En ausencia de cualquier campo de gradiente lineal, la señal emitida por los núcleos de hidrogeno es la señal de decaimiento libre de la inducción FID (free induction decay) que durará poco tiempo, generalmente menos que el tiempo previsto por la constante de tiempo T_2 . La interacción aleatoria de los spines o la relajación de los spines, así como las inhomogeneidades estáticas en un campo magnético, son responsables de la alineación rápida de la señal FID. Después de la aplicación de la señal de RF de 90° , la magnetización M_0 que estaba previamente en dirección longitudinal se alinea con el plano transversal dando lugar a la magnetización transversal neta como se observa en la Figura 8. Esto conduce a un desfase del giro que se corrige mediante la aplicación de un pulso de 180 grados. La aplicación de un pulso de 180 grados elimina las inhomogeneidades, así como la relajación spin-spin, lo que conduce a la amplitud máxima que se puede alcanzar durante una descomposición (Figura 9). El intervalo de tiempo entre la aplicación de dos pulsos de RF de 90 grados se denomina tiempo de repetición (TR) [28].

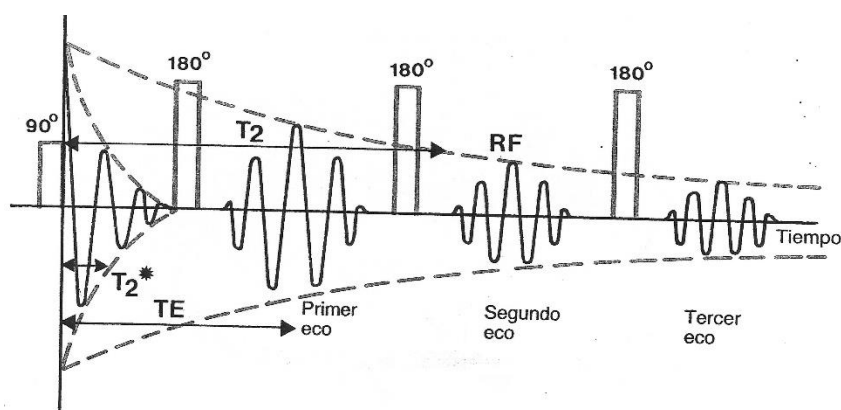


Figura 9. Secuencia de impulsos de RF [11].

Cada tipo de tejido tiene valores únicos de constante de tiempo T_1 y T_2 , modificando los tiempos de adquisición de las secuencias se puede obtener imágenes potenciadas en T_1 y T_2 lográndose de esta manera, el contraste o brillantes de una imagen [24]. Aquellos que emiten señales con tiempos más prolongados aparecerán más brillantes en comparación con aquellos que emiten señales con tiempos más cortos. En 1972, Paul Lauterbur estableció una nueva técnica denominada “zeugmatografía de RMN” que utiliza un pequeño campo magnético lineal sobre el campo magnético uniforme continuo para obtener la decodificación de la posición espacial en la frecuencia de la señal de RM [28].

La técnica propuesta por Lauterbur sigue siendo la idea central para la formación de imágenes en resonancia magnética. Esta requiere la aplicación de tres campos magnéticos lineales (G_x ,

Gy, Gz), junto con el campo magnético principal alrededor del paciente, lo que hace que varíe la frecuencia de resonancia con la posición espacial y el tiempo. Este método se está utilizando para formar imágenes desde el inicio del desarrollo de la MRI y ha demostrado ser muy exitoso y eficiente [28].

Las secuencias representan la combinación de las señales de RF con las señales de gradiente, aplicadas en un intervalo específico, para formar imágenes de alto contraste de tejidos y eliminar los artefactos lo más posible. Dos secuencias populares, utilizadas en la formación de imágenes de alto contraste, son la espín eco y la gradiente eco. Espín eco consiste en la aplicación de series de pulsos de 90 grados y pulsos de 180 grados en intervalos de TE/2 y la lectura de la señal en TE. Después del intervalo TR, los pulsos se repiten sobre la base de una serie de filas que se utilizarán en la formación de una imagen. Espín eco utiliza dos pulsos de RF con pulsos diferentes, de 90 grados y 180 grados para la generación del espín eco y el reenfoque de giro; mientras que, la técnica de gradiente eco utiliza un solo impulso de RF, lo cual hace que el proceso de adquisición de la imagen sea más rápido en comparación con el espín eco [28].

Aunque las técnicas de eco de gradiente son más rápidas que el eco de giro, tienen algunas dificultades, como la susceptibilidad magnética de los tejidos y las imperfecciones del hardware [28].

La Figura 10 muestra la secuencia de eco de spin típica donde Gx representa el gradiente de selección de segmento, Gy representa el gradiente de codificación de fase, Gz representa el gradiente de codificación de frecuencia. [28]

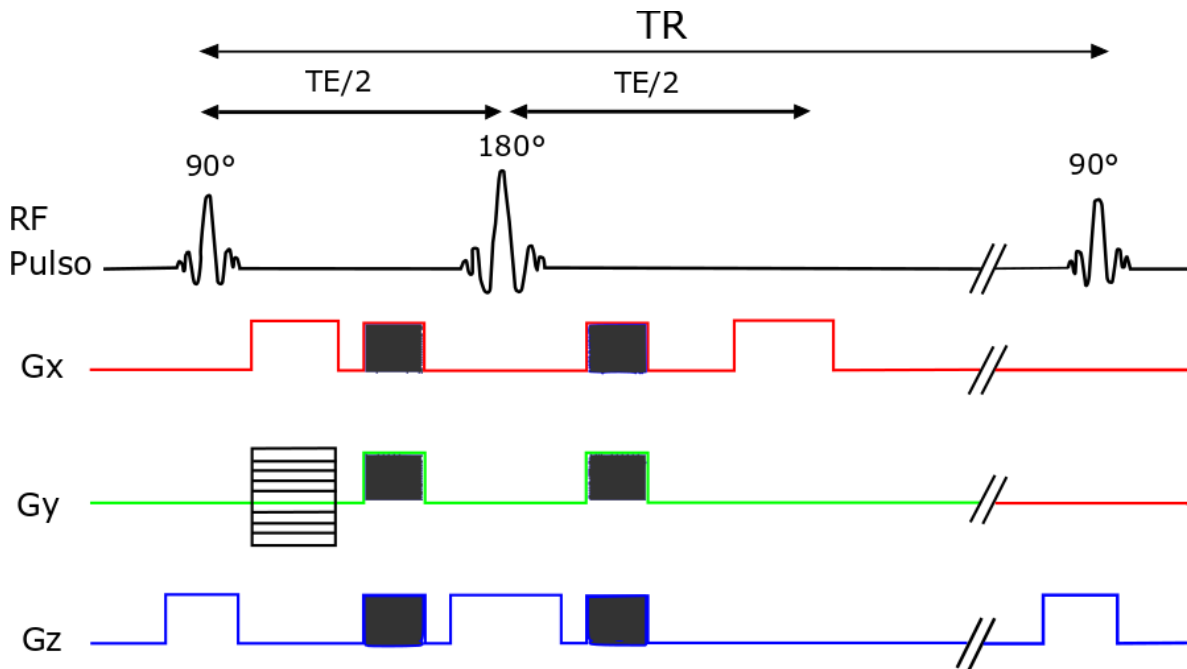


Figura 10. Secuencia típica de espín eco [28].

2.5 Hardware

2.5.1 Circuitos Programables.

En la actualidad el nivel de integración alcanzado con el desarrollo de la microelectrónica ha hecho posible desarrollar sistemas completos dentro de un solo circuito integrado SOC (System On Chip), con lo cual se han mejorado de manera notoria características como velocidad, confiabilidad, consumo de potencia y sobre todo el área de diseño.

Gracias a la escala pequeña y la alta interacción que se tiene en la actualidad en la industria, militar como de consumo, ha minimizado el tamaño de sus desarrollos; por ejemplo, la telefonía celular, computadoras personales, calculadoras de bolsillo, agendas electrónicas, relojes digitales, sistemas de audio, sistemas de telecomunicaciones, entre otras. Estas no son más que aplicaciones típicas que muestran la evolución de los circuitos integrados, también conocidos como chips [29].

Los dispositivos lógicos programables (PLD, por sus siglas en inglés) favorecen la interacción de aplicaciones y desarrollos lógicos mediante el empaquetamiento de soluciones en un circuito integrado. Los PLD de alto nivel de integración se crearon con el objeto de tener mayor cantidad de dispositivos en un circuito (SOC). Se caracteriza por la reducción del espacio y costo, además de ofrecer una mejora sustancial en el diseño de sistemas complejos, dado que incrementan la velocidad y la frecuencia de operación. Los sistemas PLD se clasifican por su arquitectura en: PROM, PLA, PAL, GAL, CPLD, FPGA [29].

Los arreglos de compuertas programables en campo (FPGAs) son chips de silicio reprogramables. A diferencia de los procesadores que se encuentran en una PC, al programar un FPGA el chip se vuelve a configurar para implementar su funcionalidad, en lugar de ejecutar una aplicación de software. Ross Freeman, el cofundador de Xilinx, inventó el FPGA en 1985 [30].

Al utilizar bloques de lógica pre-construidos y recursos para ruteo programables, se pueden configurar estos chips para implementar funcionalidades personalizadas en hardware sin tener que utilizar una tablilla de prototipos o un caudín.

Además, los FPGAS está compuesto por un numero finito de recursos predefinidos con interconexiones programables para implementar un circuito digital reconfigurable y bloques de E/S para permitir que el circuito acceda al exterior [31].

Sólo deberá desarrollar tareas de cómputo digital en software y compilarlas en un archivo de configuración o *bitstream* que contenga información de cómo deben conectarse los componentes. Además, los FPGAs son completamente reconfigurables y al instante toman una nueva “personalidad” cuando se compila una configuración de circuito diferente. Anteriormente, sólo los ingenieros con un profundo entendimiento de diseño de hardware digital podían trabajar con la tecnología FPGA. Sin embargo, el aumento de herramientas de

diseño de alto nivel está cambiando las reglas de programación de FPGAs, con nuevas tecnologías que convierten los diagramas a bloques, gráficos, o hasta el código ANSI C a circuitos de hardware digital. Estas herramientas pueden realizar la programación en VHDL ya que se pueden utilizar las IP (XILINX), SYSTEM GENERATOR de MATLAB, incluso LabVIEW. Esto con el fin de llevar al usuario a tener una experiencia amigable con el sistema y aprovechar la tecnología FPGA.

Cabe mencionar que para utilizar este tipo de software hay que tomar en cuenta que se debe de tener un amplio conocimiento sobre el FPGA en uso, así como sus limitantes. Ya que cada FPGA tiene sistemas embebidos para tareas específicas, esto quiere decir que no es lo mismo utilizar un FPGA BASYS 3 a un ARTY A7 o incluso un Nexys Video de la familia XILINX. La adopción de chips FPGA en la industria ha sido impulsada por el hecho de que los FPGAs combinan lo mejor de los ASICs y de los sistemas basados en procesadores. A diferencia de los procesadores, los FPGAs llevan a cabo diferentes operaciones de manera paralela, por lo que éstas no necesitan competir por los mismos recursos. Cada tarea de procesos independientes se asigna a una sección dedicada del chip, y puede ejecutarse de manera autónoma sin ser afectada por otros bloques de lógica. Como resultado, el rendimiento de una parte de la aplicación no se ve afectado cuando se agregan otros procesos [32].

Los cinco beneficios principales de la tecnología FPGA son: rendimiento, tiempo en llegar al mercado, precio, fiabilidad y mantenimiento a largo plazo [32].

Estos dispositivos tienen recursos que incluyen cantidades de bloques lógicos configurables y bloques lógicos de función fija, como multiplicadores y bloques de RAM. Los bloques lógicos configurables (CLB por sus siglas en inglés) son la unidad lógica básica de un FPGA, también llamadas divisiones o celdas lógicas como se puede observar en la Figura 11. Estos bloques se componen básicamente de flip-flops y tablas de búsqueda (LUT por sus siglas en inglés) [31].

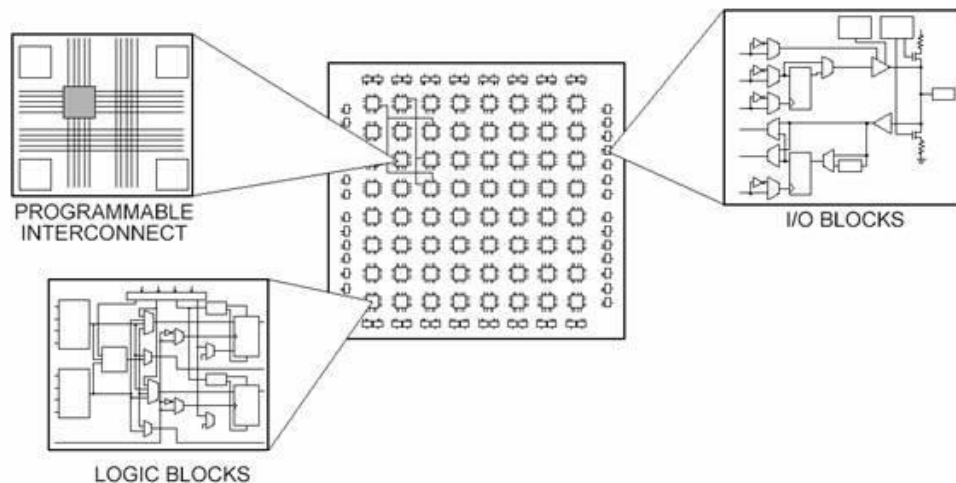


Figura 11. Partes de un FPGA [31].

Los flip-flops son registros de desplazamiento binario utilizados para sincronizar la lógica y guardar estados lógicos entre ciclos de reloj dentro de un circuito FPGA. En cada borde de reloj, un flip-flop bloquea el valor 1 o 0 (verdadero o falso) en su entrada y mantiene ese valor constante hasta el siguiente borde de reloj. La Mayor parte de la lógica en un CLB se implementa utilizando cantidades muy pequeñas de RAM en forma de LUT. Es fácil suponer que el número de compuertas del sistema en un FPGA se refiere al número de compuertas NAND y compuertas NOR en un chip en particular. Pero, en realidad, toda la lógica combinatoria (AND, OR, NAND, XOR, etc.) se implementa como tablas de verdad dentro de la memoria LUT. Una tabla de verdad es una lista predefinida de salidas para cada combinación de entradas [31].

Para los multiplicadores, los FPGA tienen circuitos preconstruidos, que permiten ahorrar el uso de LUT y flip-flop en aplicaciones de procesamiento de señales y matemáticas, como se puede ver en la Figura 12, la muestra un multiplicador de 4x4 preconstruido con compuertas lógicas [31].

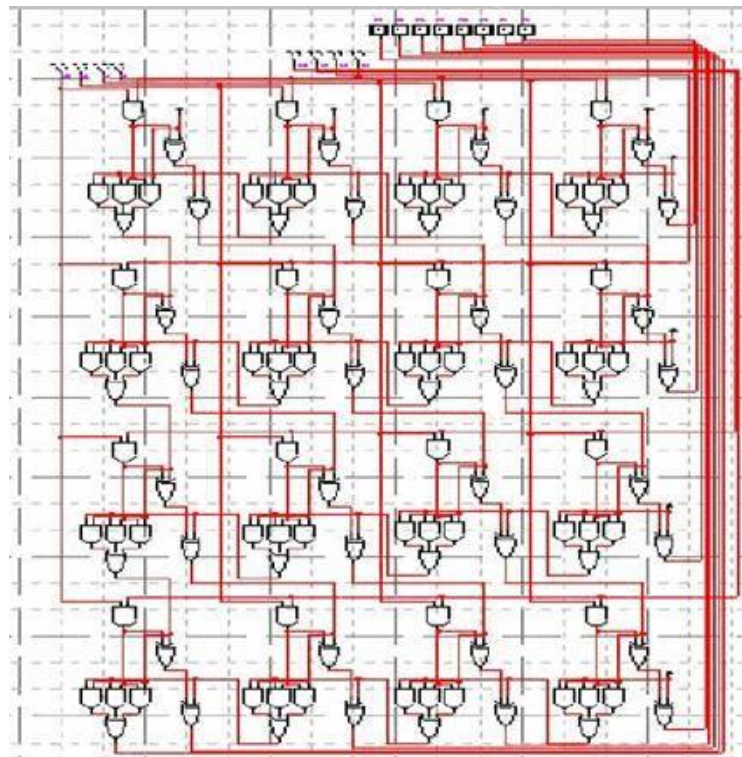


Figura 12. Dibujo esquemático de un multiplicador de 4X4 Bits [31].

Muchos algoritmos de procesamiento de señales implican mantener el total acumulado de números que se multiplican y, como resultado, los FPGA de mayor rendimiento como los FPGA XILINX Arty A7, tienen circuitos de multiplicación y acumulación (MAC) preconstruidos. Estos bloques de procesamiento preconstruidos, son conocidos como cortes DSP48 [31].

Por otro lado, los recursos de memoria es otro punto a considerar en los FPGAS. La RAM definida por el usuario, integrada en todo el chip FPGA, es útil para almacenar conjuntos de datos o pasar valores entre tareas paralelas. Dependiendo de la familia FPGA, puede configurar la RAM incorporada en bloques de 16 o 36KB [31].

La ejecución paralela inherente de los FPGA permite que diferentes piezas de lógica de hardware sean controladas por diferentes relojes. El paso de datos entre la lógica que se ejecuta a diferentes velocidades puede ser complicado y la memoria interna a menudo se usa para suavizar la transferencia utilizando buffers de memoria primero en entrar, primero en salir (FIFO por sus siglas en inglés) [31].

Durante los primeros 20 años de desarrollo de los FPGA, los lenguajes de descripción de hardware (HDL) como VHDL y verilog evolucionaron a los idiomas principales para diseñar los algoritmos que se ejecutan en el chip FPGA. Estos lenguajes de bajo nivel integran algunos de los beneficios que ofrecen otros lenguajes textuales con la consideración de que en un FPGA está describiendo un circuito. La sintaxis híbrida resultante requiere que las señales se asignen o se conecten a puertos de E/S externos a señales internas, que finalmente están conectadas a las funciones que alojan los algoritmos. Estas funciones se ejecutan secuencialmente y pueden hacer referencia a otras funciones dentro de los FPGA [31].

Para verificar la lógica creada por un programador de FPGA, es una práctica común escribir bancos de prueba (test bench) en HDL para ejecutar y evaluar el diseño del sistema y así confirmar entradas y salidas. El banco de pruebas y el código FPGA se ejecutan en un entorno de simulación que modela el comportamiento de temporización de hardware del chip FPGA y muestra todas las señales de entrada y salida al diseñador para la validación de la prueba. El proceso de crear el banco de pruebas HDL y ejecutar la simulación a menudo requiere más tiempo que crear el diseño FPGA HDL original [31].

Una vez que ha creado o desarrollado un diseño FPGA usando HDL y lo haya verificado, debe alimentarlo a una herramienta de compilación que tome la lógica basada en texto y, a través de varios pasos complejos, sintetice su HDL en un archivo de configuración o flujo de bits que contenga información sobre cómo deben conectarse los componentes entre sí, esto se puede ver en la Figura 13 que muestra el cuerpo de un contador en VHDL con sus respectivas entradas y salidas. Como parte de este proceso manual de varios pasos, a menudo se requiere que especifique una asignación de nombres de señal a los pines en el chip FPGA que se está utilizando [31].

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity signed_adder is
6   port
7   (
8     aclr : in   std_logic;
9     clk  : in   std_logic;
10    a    : in   std_logic_vector;
11    b    : in   std_logic_vector;
12    q    : out  std_logic_vector
13  );
14 end signed_adder;
15
16 architecture signed_adder_arch of signed_adder is
17   signal q_s : signed(a'high+1 downto 0); -- extra bit wide
18
19 begin -- architecture
20   assert(a'length >= b'length)
21     report "Port A must be the longer vector if different sizes!"
22     severity FAILURE;
23   q <= std_logic_vector(q_s);
24
25   adding_proc:
26   process (aclr, clk)
27   begin
28     if (aclr = '1') then
29       q_s <= (others => '0');
30     elsif rising_edge(clk) then
31       q_s <= ('0'&signed(a)) + ('0'&signed(b));
32     end if; -- clk'd
33   end process;
34
35 end signed_adder_arch;

```

Figura 13. Cuerpo de un programa en VHDL que describe un contador [31].

2.5.2 Herramientas de diseño.

VHDL (Very High Speed Integrated Circuit hardware description language) es un lenguaje orientado a la descripción o modelado de sistemas digitales, es decir, se trata de un lenguaje mediante el cual se puede describir, analizar y evaluar el comportamiento de un sistema electrónico digital. Este lenguaje fue estandarizado por la IEEE en 1987 y después se realizó una actualización en 1993.

VHDL es un lenguaje poderoso que permite la integración de sistemas digitales sencillos, elaborados o ambos en un dispositivo lógico programable, sea de baja capacidad de integración como un GAL, o de mayor capacidad como los CPLD y FPGA [29].

Xilinx posee diversas herramientas que pueden ser usadas para la programación de FPGAs. Entre algunas de estas herramientas se encuentran ISE DESIGN y VIVADO, que permiten programar los FPGAs tanto en VHDL como en verilog. Para poder hacer simulaciones en el FPGA, Xilinx cuenta con ChipScope, que permite cargar los diseños al sistema y analizarlos.

Otra herramienta es System Generator que permite realizar una simulación de un FPGA en el ambiente Simulink de MATLAB por medio de bloques. MATLAB permite realizar co-simulaciones de un sistema FPGA y realizar la transformación a VHDL.

También se pueden encontrar en el ambiente de programación herramientas que facilitan al programador el desarrollo de proyectos, las llamadas IP (propiedad intelectual).

2.5.3 Convertidores Analógico – Digital (ADC) y Digital – Analógico (DAC)

El new Collegiate Dictionary de webster define una señal como “Una cantidad o impulso físico detectable (o medible) (como voltaje, corriente o intensidad del campo magnético) mediante la cual se pueden transmitir mensajes o información” [33].

Por su propia naturaleza, las señales son analógicas, ya sea CD, CA, niveles digitales o pulsos. Sin embargo, es habitual diferenciar entre señales analógicas y digitales de la siguiente manera: las variables analógicas (o del mundo real) en la naturaleza incluyen todas las cantidades físicas medibles [33].

Algunas señales resultan en respuesta a otras señales. Un buen ejemplo es la señal de retorno de un sistema de imágenes de ultrasonido o radar, que resulta de una señal transmitida conocida. Por otro lado, las señales digitales corresponden a cantidades físicas que han sido condicionadas y formateadas en dígitos.

Estas señales digitales pueden o no estar relacionadas con variables analógicas del mundo real. Los ejemplos incluyen los datos transmitidos a través de redes de área local (LAN) u otras redes de alta velocidad. En el caso específico del procesamiento digital de señales (DSP), la señal analógica se convierte en forma binaria mediante un dispositivo conocido como un convertidor analógico a digital (CAD). La salida del CAD es una representación binaria de la señal analógica, la cual es manipulada aritméticamente por el procesador digital de señales. Después del procesamiento, la información obtenida de la señal puede volver a convertirse en forma analógica utilizando un convertidor digital a analógico (CDA). Otro concepto clave incorporado en la definición de la señal es que hay algún tipo de información contenida en la señal. Esto nos lleva a la razón clave para procesar señales analógicas del mundo real: la extracción de información [33].

Esta información normalmente existe en forma de amplitud de señal (absoluta o relativa), frecuencia o contenido espectral, fase o relaciones de temporización con respecto a otras señales. Una vez que la información deseada se extrae de la señal, se puede utilizar de varias maneras.

En algunos casos, puede ser conveniente reformatear la información contenida en una señal. Este sería el caso en la transmisión de una señal de voz a través de un sistema telefónico de

acceso múltiple por división de frecuencia (FDMA por sus siglas en inglés). En este caso, se utilizan técnicas analógicas para “apilar” canales de voz en el espectro de frecuencia para la transmisión a través de un relé de microondas, cable coaxial o fibra. En el caso de un enlace de transmisión digital, la información de voz analógica se convierte primero en digital utilizando un CAD. La información digital que representan los canales de voz individuales se multiplexa en el tiempo (acceso múltiple por división de tiempo, o TDMA por sus siglas en inglés) y se transmite a través de un enlace de transmisión digital en serie (como en el sistema T-Carrier) [33].

Otro propósito del procesamiento de la señal es comprimir el contenido en frecuencias de la señal (sin perder información importante), luego formatearla y transmitirla a velocidades de datos más bajas, logrando así una reducción en el ancho de banda del canal requerido. Analogic Devices ofrece una familia de MicroConverters que incluye circuitos de acondicionamiento analógico de precisión: CAD, CDA, microcontrolador y memoria FLASH en un solo chip o tarjeta.

En algunos casos, la señal que contiene la información está enterrada en ruido y el objetivo principal es la recuperación de la señal. Las técnicas como filtrado, autocorrelación, convolución, etc., se utilizan a menudo para realizar esta tarea tanto en el dominio analógico como en el digital [33].

Hay diversos tipos de convertidores según su aplicación y contienen diferentes parámetros que los caracterizan, los cuales pueden ser: la velocidad de conversión, la resolución, los rangos de entrada, etc. Por ejemplo, una mayor cantidad de bit, implica mayor precisión, pero también mayor complejidad. Un incremento en un solo bits permite disponer del doble de precisión (mayor resolución), pero hace más difícil el diseño del circuito, además, la conversión podría volverse más lenta [34].

Por otro lado, es necesario tomar en cuenta los alcances del sistema a desarrollar y su aplicación. Para este caso de estudio se seleccionaron dos tarjetas con las características adecuadas para la conversión analógica - digital como digital – analógica, a continuación, se da una breve explicación de estos dispositivos y sus más características más relevantes.

El AD6644 es un convertidor analógico digital (CAD) monolítico de 14 bits de alta velocidad y alto rendimiento. Todas las funciones necesarias, incluyendo muestreo, retención (TH) y referencia, se realizan en el chip para proporcionar una solución de conversión completa. El dispositivo proporciona salidas digitales compatibles con circuitos CMOS [35].

El AD6644 alcanza un rango dinámico sin espurias (SFDR), de 100dB a través de la banda Nyquist. Este rendimiento innovador alivia la carga de los receptores digitales multimodo (radios software), que normalmente están limitados por el CAD. Con el sobremuestreo, los armónicos se pueden colocar fuera del ancho de banda de análisis. El sobremuestreo también facilita el uso de receptores de decimación, lo que permite reducir el ruido de fondo en el

ancho de banda de análisis. Al remplazar los filtros analógicos tradicionales con componentes digitales predecibles, los receptores modernos pueden construirse utilizando menos componentes de RF, lo que resulta en menores costos de fabricación, mayores rendimientos y una confiabilidad mejorada [35].

El AD9755 es un convertidor digital analógico que tiene dos puertos de entrada, de alta velocidad multiplexados, con un canal único conectado al CDA CMOS, de 14 bits. El AD9755 ofrece un rendimiento excepcional de CA y CC al tiempo que admite tasas de actualización de hasta 300 MSPS. También se ha optimizado para aplicaciones de velocidad ultra alta de hasta 300 MSPS en las que las velocidades de datos superan las posibles en un solo puerto de interfaz de datos CDA. El PLL del CDA puede tener el doble de velocidad del reloj aplicado externamente y es capaz de intercalar los datos de los dos canales de entrada. La velocidad de datos de salida resultante es el doble que la de los dos canales de entrada.

El CDA utiliza una arquitectura de fuente de corriente segmentada, combinada con una técnica de conmutación patentada, para reducir la falla de energía y maximizar la precisión dinámica. Las salidas de corriente diferencial admiten aplicaciones en las que se utiliza terminal referenciada a tierra o terminal diferencial. Cada una de las salidas diferenciales puede proporcionar una corriente en el rango de 2mA a 20mA [36].

2.6 Muestreo.

Una señal porta información mediante la variación en el tiempo, t , que puede asumir valores en un rango continuo o valores discretos, nt_s , donde t_s es un valor fijo conocido como intervalo de muestreo y n es un valor entero. De lo indicado antes, además, se pueden considerar dos tipos básicos de señales: continuas y discretas (Figura 14). En el caso de las señales continuas la variable independiente es continua (t) [37]. Por otra parte, las señales discretas solo están definidas en tiempos discretos y, en consecuencia, para estas señales la variable independiente toma solamente un conjunto discreto de valores. La amplitud también puede tomar valores continuos o estar cuantizada en un número finito de niveles discretos entre sus extremos [38].

Para distinguir las señales discretas de las continuas, se usa el símbolo ‘ t ’ para denotar la variable independiente continua y ‘ n ’ para indicar la variable independiente discreta. También se puede considerar que, para representar señales de forma gráfica, una señal continua se representa como $x(t)$ y una señal discreta como $x[n]$ [37].

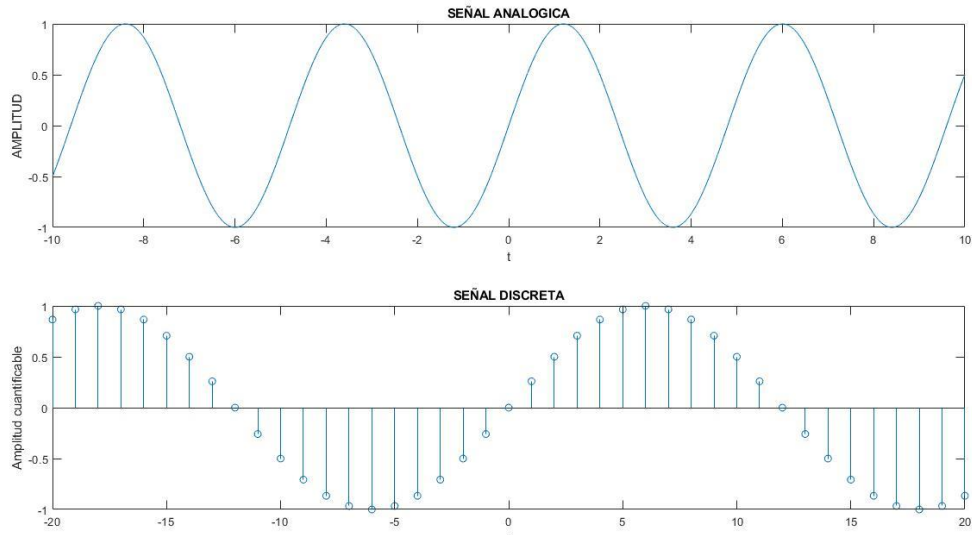


Figura 14. Señal discreta y analógica.

En el procesamiento de señales, los CDA y CAD permiten modificar o tratar a la señal, ya sea en tiempo continuo o discreto, para extraer la información requerida por el usuario. Para este procesamiento de la señal en recepción, pueden aplicarse diferentes técnicas de muestreo. Para el caso de recepción de señales estas técnicas pueden ser sobremuestreo y submuestreo que se explicará más adelante. Estas técnicas permiten que el sistema sea más eficiente, o reducir el ruido de cuantización o permitir realizar un filtrado que remueva el ruido suficiente de la señal a tratar [35]. Antes, se ha mencionado que los SDR son utilizados en los sistemas RMN. Los SDR requieren de CDA y CAD que permitan convertir las señales de recepción y transmisión para adquirir los datos necesarios para su estudio.

Primero se revisará el teorema fundamental del muestreo para los sistemas SDR. El teorema de Nyquist establece que: cualquier señal puede representarse mediante muestras discretas si la frecuencia de muestreo es al menos el doble del ancho de banda de la señal.

Para visualizar lo que sucede en el muestreo, imagine que está utilizando un papel de computadora transparente con "plegado en abanico". Use el borde horizontal del papel como el eje de frecuencia y escálelo de manera que los pliegues del papel se alineen con los múltiplos enteros de la mitad de la frecuencia de muestreo f_s . Cada hoja de papel ahora representa lo que llamaremos una "Zona Nyquist" tal como se presenta en la Figura 15 [18].

En esta figura se muestra la señal solo en una zona Nyquist ya que se ha aplicado un filtro pasa bajas para solo tener la señal en esa zona sin pérdida de información. Esto con el fin de evitar copias de la señal y así evitar la destrucción de esta por generación de imágenes espectrales.

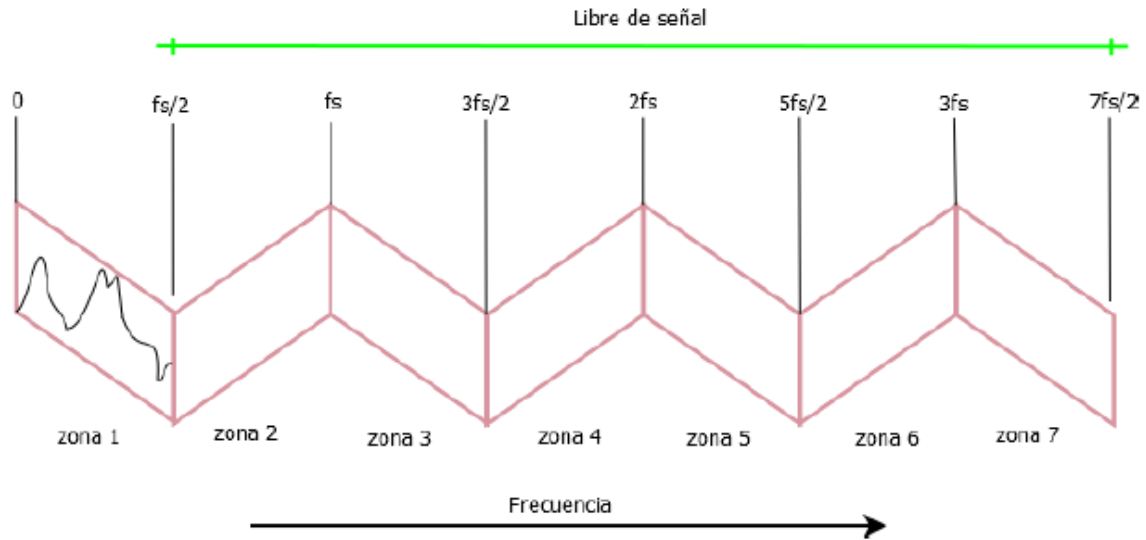


Figura 15. Representación de la zona Nyquist [18].

Utilice el eje vertical del papel plegado en abanico para observar la magnitud de la señal y trazar el espectro de frecuencias de la señal a muestrear, como se muestra en la Figura 16. Para ver los efectos del muestreo, coloque el papel plegado en abanico en una pila. El espectro resultante se puede ver sosteniendo la pila transparente en una luz y mirando a través de ella. Puede ver que las señales en todas las hojas o zonas están "dobladas" o en "alias" una encima de la otra, y ya no se pueden separar. Una vez que este plegamiento o alias ocurre durante el muestreo, los datos muestreados resultantes se corrompen y nunca se pueden recuperar. El término "aliasing" es apropiado porque después del muestreo, una señal de una de las zonas más altas ahora parece estar en una frecuencia diferente [18].

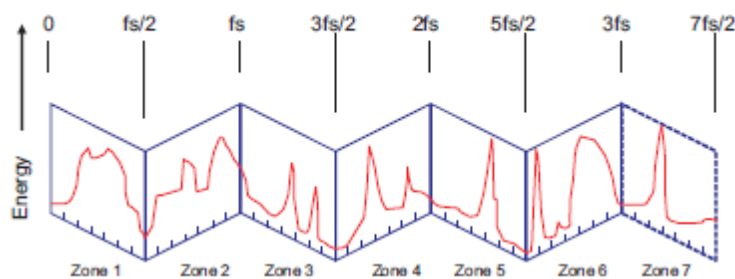


Figura 16. Magnitud y frecuencia de la señal muestreada [18].

Una señal de banda base tiene componentes de frecuencia que comienzan en $f = 0$ y se extienden hasta cierta frecuencia máxima. Para evitar la destrucción de datos al muestrear

una señal en banda base, se debe asegurar de que toda la energía de la señal caiga solamente en la primera banda de Nyquist, como se muestra en la Figura 15. Hay dos formas de hacer esto:

1. Insertando un filtro paso bajo, para eliminar todas las señales por encima de $f_s / 2$, ó
2. Aumentando la frecuencia de muestreo para que todas las señales presentes caigan por debajo de $f_s / 2$. Se debe tener en cuenta que $f_s / 2$ también se conoce como la "frecuencia de plegado".

2.6.1 Sobremuestreo.

Un convertidor analógico - digital (CAD) es una interfaz activa entre las cadenas de señales analógicas y digitales en un sistema embebido. Un CAD convierte señales analógicas en señales digitales en sistemas electrónicos. La característica clave de un CAD es la precisión (resolución) que ofrece. Cuanto mayor sea la precisión deseada, mayor será el costo del CAD. Una mayor precisión del CAD se logra mediante el diseño de hardware para cuantificar la amplitud de la señal analógica en la señal digital con una mayor longitud de palabra. Los CAD prácticos tienen una longitud de palabra finita. Para lograr un equilibrio efectivo entre el costo y la precisión del sistema, se logra una mayor precisión de conversión mediante el sobremuestreo del CAD de baja resolución.

La idea básica de los convertidores A/D con sobremuestreo consiste en aumentar la tasa de muestreo de la señal hasta el punto en que pueda emplear un cuantificador de baja resolución. El sobremuestreo permite reducir el rango dinámico de valores de la señal entre muestras sucesivas y reducir los requisitos de resolución del cuantificador. Además, reduce el ruido de cuantización y el SNR (Signal Noise Ratio) [36]. Por otro lado, esta técnica permite separar las réplicas espectrales de la señal y facilitar las condiciones de filtrado; por ejemplo, si se lleva a cabo un sobre muestreo por un factor de 10, entonces también se tendrá una tasa de datos 10 veces mayor para almacenar y procesar [39].

La técnica de sobremuestreo es útil cuando se necesita capturar bordes rápidos, transitorios y eventos que solo ocurren una vez; es un método para mejorar la resolución de un convertidor [39]. El principio del sobremuestreo es tomar un gran número de muestras y calcular un valor medio [40]. Como se puede observar en la Figura 17 (A) se muestra una señal senoidal con cierto tiempo y nivel de voltaje. Según la teoría de Nyquist se puede muestrear la señal a dos veces su frecuencia de la señal original sin pérdidas de información como se puede analizar en la Figura 17 (B). Si se sobremuestra se tiene mayor números de muestras teniendo una mejor resolución de la señal original como se puede ver en la Figura 17 (C).

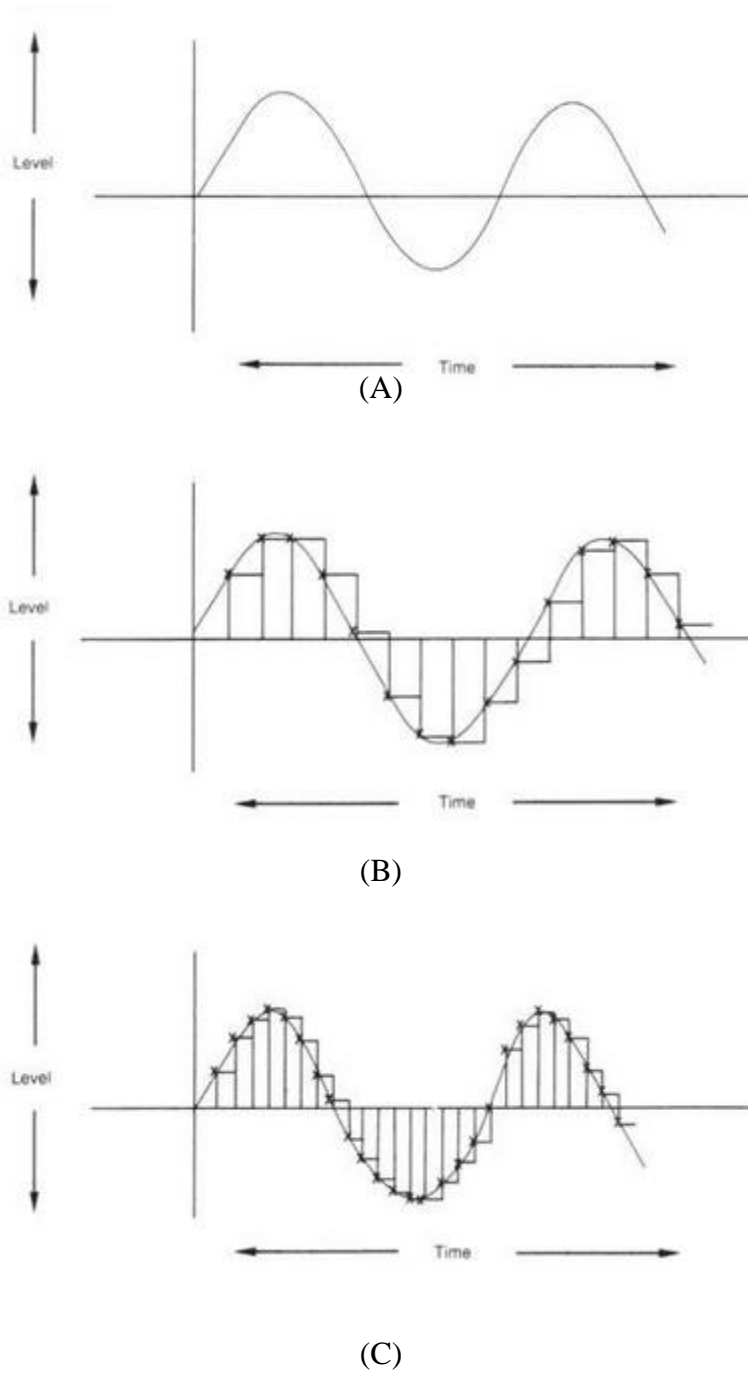


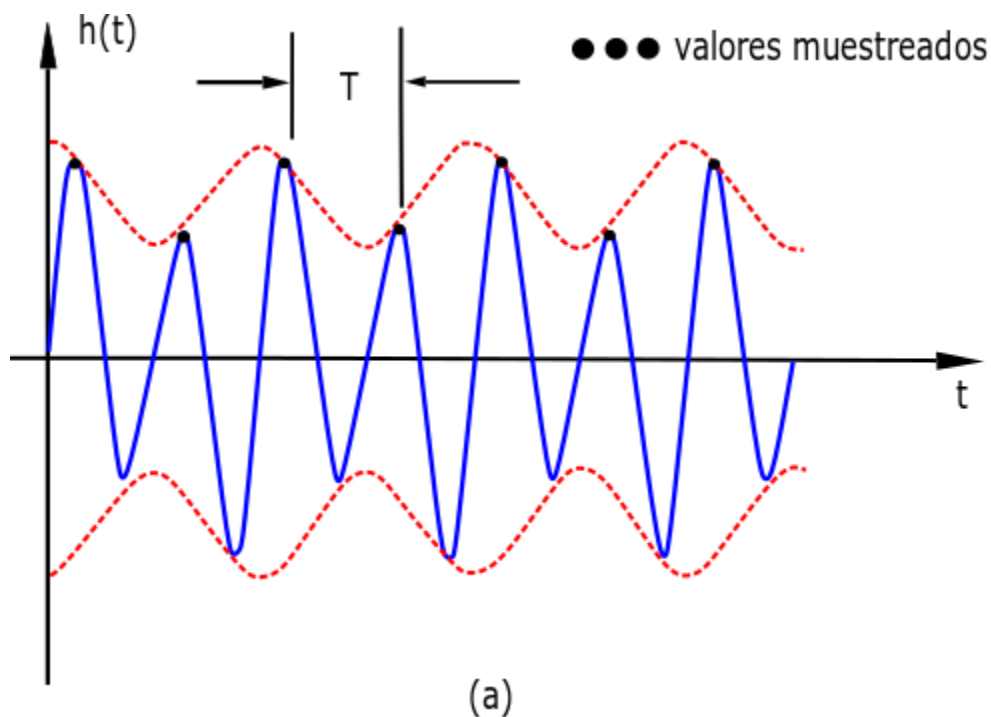
Figura 17. Sobremuestreo de una señal: (A) Señal original, (B) Señal muestreada dos veces su frecuencia, (C) Señal Submuestreada [41].

Dicho de otra manera, un sobremuestreador con un factor de muestreo de L , o expansor por L , inserta $L-1$ muestras nulas entre muestras consecutivas, haciendo que el espectro de salida sea una versión comprimida del espectro de la entrada; esta compresión de las réplicas centradas en los múltiplos de 2π hace que aparezcan $L - 1$ copias nuevas en el intervalo $[-\pi, \pi]$ [39].

2.6.2 Submuestreo o muestro paso banda.

La FFT se utiliza a menudo en aplicaciones de procesamiento de señales paso banda. El muestreo eficiente de las señales paso banda es de suma importancia cuando se utiliza la FFT. Por esta razón, desarrollamos el teorema del muestreo paso banda, como un caso especial de los criterios de Nyquist para el muestreo de formas de onda en banda base.

Para ilustrar el concepto de muestreo de una señal paso de banda, considere la forma de onda en el dominio del tiempo que se muestra en la Figura 18. La forma de onda mostrada por la línea continua en la Figura 18 (a) es una señal paso banda, modulada en amplitud. La línea de puntos representa la modulación, o contenido de información, de la señal. Tenga en cuenta que la forma de onda de modulación se muestrea dos veces por período, pero la frecuencia de la portadora se muestrea una sola vez por período. Como se muestra, las muestras caracterizan completamente la modulación, o la información, la forma de onda, aunque la frecuencia de muestreo da como resultado un alias de la señal paso banda. La forma de onda de la Figura 18 (a) fue muestreada en sincronismo con el pico de la forma de onda portadora para claridad en la presentación. Esto no es un requisito para el muestreo de paso banda, como se ilustra en la Figura 18 (b) [42].



Aquí mostramos la misma frecuencia de muestreo que antes, pero con un ligero retraso. La forma de onda de línea discontinua representada por las muestras es la señal de modulación. Se supone que las formas de onda paso banda tienen un espectro distinto de cero solo en el intervalo de frecuencia $f_l < |f| < f_h$ 'donde f_h y f_l son las frecuencias más alta y más baja que forman parte del espectro de la señal paso banda, respectivamente.

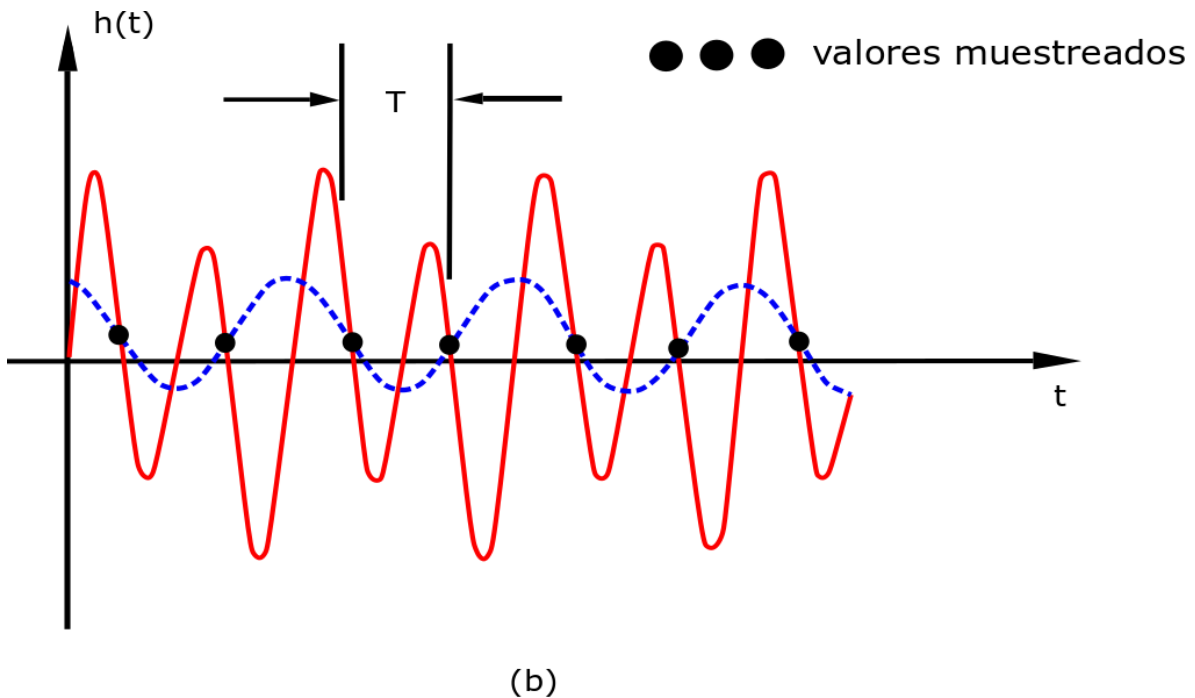


Figura 18. Ejemplo de una señal paso banda: (a) muestreo síncrono con el pico máximo de la forma de onda, y (b) el caso general [42].

El ancho de banda de transmisión de una señal paso banda se define como $B_T = f_h - f_l$. Al utilizar el criterio de Nyquist, se tomará una muestra de la señal paso banda a una frecuencia de $2f_h$ muestras por segundo, para garantizar que no se produzca alias o solapamiento durante el muestreo. Sin embargo, el proceso de muestreo produce imágenes del espectro (alias) espaciadas en armónicos de la frecuencia de muestreo. El aliasing se puede usar ventajosamente cuando se muestrean señales paso banda, seleccionando una tasa de muestreo inferior a $2f_h$ ($B_T \ll f_l$) si asociamos la señal paso banda con una de las imágenes de aliasing [42].

El teorema de muestreo paso banda establece que una señal paso banda puede reproducirse a partir de los valores de sus muestras si la frecuencia de muestreo f_s satisface la relación

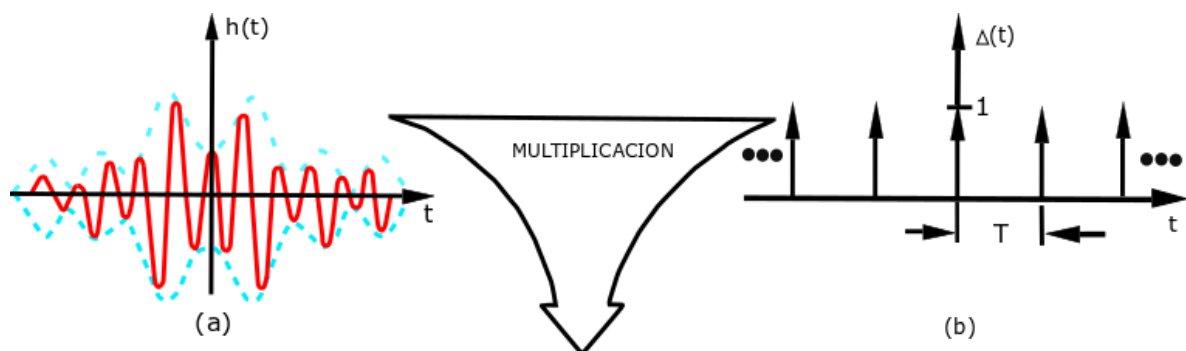
$$\frac{2f_h}{n} \leq f_s \leq \frac{2f_l}{(n-1)} \quad (3)$$

$$2 \leq n \leq f_h(f_h - f_l) \quad (4)$$

y n es un valor entero. La condición de la ecuación 3 asegura que no se produce la superposición del espectro y solo produce frecuencias de muestreo aceptables para $f_s < 2f_h$. Tener en cuenta que si $n = n'$ es el número entero más grande que no exceda $\frac{f_h}{(f_h - f_l)}$, entonces la frecuencia de muestreo crítica (más baja) para una señal paso banda viene dada por la ecuación 3 como $f_s' = \frac{2f_h}{n'}$. También observe que si elegimos $n = \frac{f_h}{(f_h - f_l)}$, entonces la ecuación 3 requiere $f_s \geq 2(f_h - f_l) = 2B_T$ [42].

Se ilustra el concepto de muestreo eficiente de señales paso banda en la Figura 19 por medio del teorema de convolución. En la Figura 19 se muestran una forma de onda paso banda en el dominio del tiempo y el espectro de frecuencias paso banda correspondiente Figura 19 (a) y Figura 19 (c), respectivamente. Observe en la Figura 19 (c) que la frecuencia central del espectro paso banda es $8f_0$ y el ancho de banda de transmisión B_T es $2f_0$. Si se elige $f_s = 6f_0$, que satisface las restricciones del teorema de muestreo paso banda de la ecuación 3, para $n = 3$. La función de muestreo en el dominio del tiempo se muestra en la Figura 19 (b) y la correspondiente función de muestreo en el dominio de la frecuencia se muestra en la Figura 19 (d) [42].

La multiplicación de la forma de onda paso banda en el dominio del tiempo de la Figura 19 (a) y la función de muestreo de la Figura 19 (b) da como resultado la forma de onda muestreada ilustrada en la Figura 19 (e). Recordando el teorema de convolución, que indica que la multiplicación en el dominio del tiempo implica la convolución en el dominio de la frecuencia; entonces, la transformada de Fourier de la función de muestreo en el tiempo de la Figura 19 (b) se convoluciona con el espectro de señal paso banda que se muestra en la Figura 19 (c), dando como resultado la función en frecuencia con aliasing es ilustrada en la Figura 19 (f). Se observa que en la Figura 19 (f) la función de frecuencia muestreada centrada en la frecuencia $\pm 2f_0$ es idéntica a la función de frecuencia paso banda original centrada en la frecuencia $\pm 8f_0$. Aunque la función centrada en $\pm 2f_0$ se debe al aliasing, no se ha perdido información debido a la superposición del espectro [42].



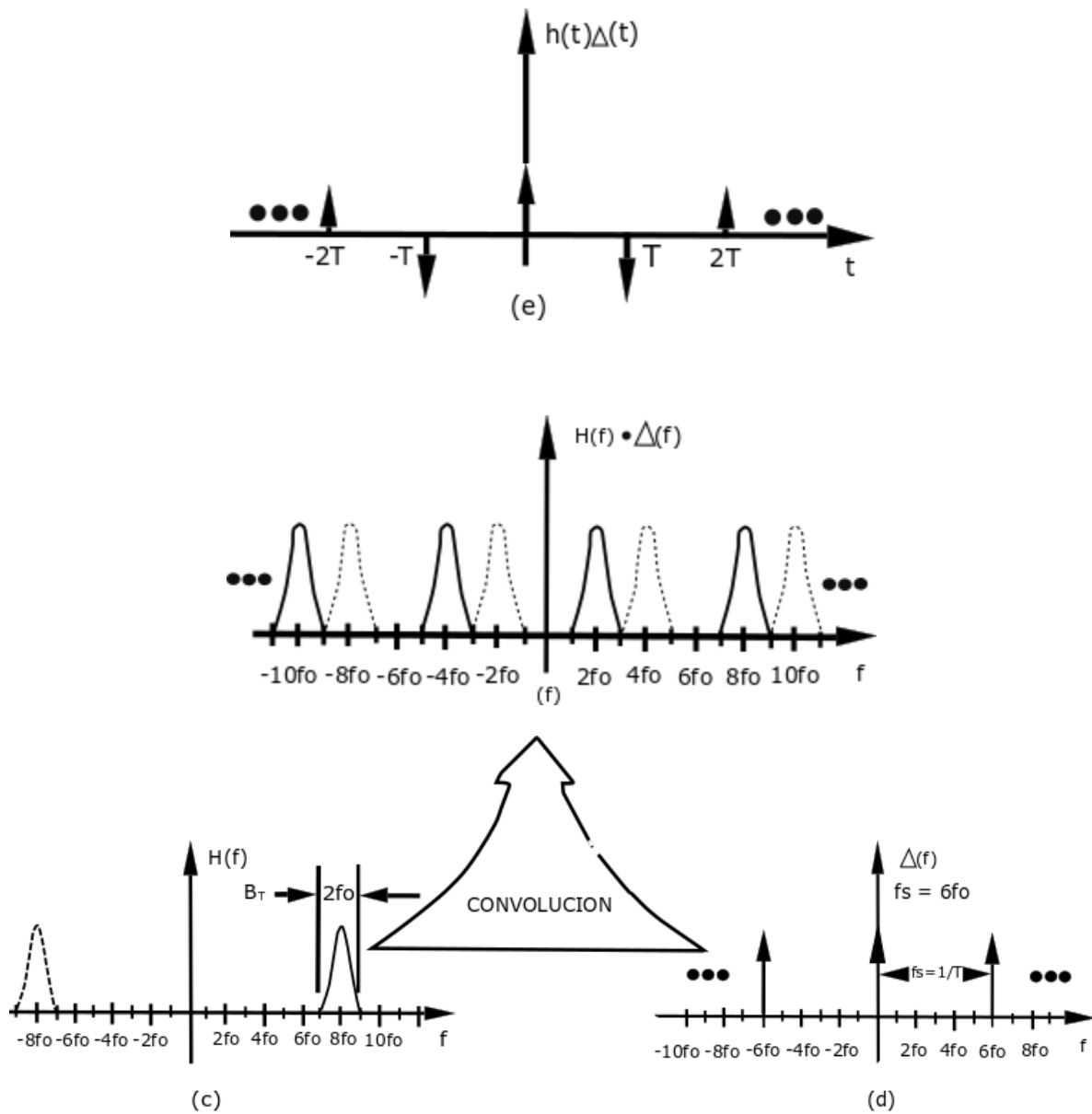


Figura 19. Transformada de Fourier con alias no superpuesto de una forma de onda paso banda que se muestrea a menos del doble del componente de frecuencia más alto [42].

Las funciones de frecuencia muestreadas centradas en $\pm 4f_0$ y $\pm 10f_0$ en la Figura 19 (f) también son los resultados del aliasing. Estos términos pueden ignorarse porque se puede mostrar que un filtro paso bajas con ancho de banda $3f_0$ reconstruye la señal original $h(t)$ con solo un cambio de la frecuencia central de $8f_0$ a $2f_0$ [42].

La componente de frecuencia más alta de la forma de onda paso banda de la Figura 19 (a) es $9f_0$. Por lo tanto, la aplicación del teorema del muestreo de Nyquist para señales en banda base requiere una frecuencia de muestreo de $18f_0$. Debido a que se muestrea a una tasa de solo $6f_0$ sin pérdida de información, se dice que la forma de onda se ha submuestreado o decimado. Se puede obtener muestras sin superposición de espectros siempre que la frecuencia de muestreo f_s satisfaga el teorema del muestreo paso banda de la ecuación (3) [42].

Un espectro que no se puede obtener directamente de la ecuación 3 se muestra en la Figura 20 utilizando un desarrollo gráfico del teorema del muestreo paso banda. En la Figura 20 (a), se ilustra la función en el dominio de la frecuencia de una señal paso banda. Esta señal tiene una frecuencia central de $14f_0$ y un ancho de banda $BT < 2f_0$ (es decir, la amplitud en frecuencia de la señal para las frecuencias f_h y f_l es igual a cero). El procedimiento gráfico de convolución en frecuencia se utiliza en las Figuras 20 (b) a (i) para ilustrar el efecto de muestrear una señal paso banda [42].

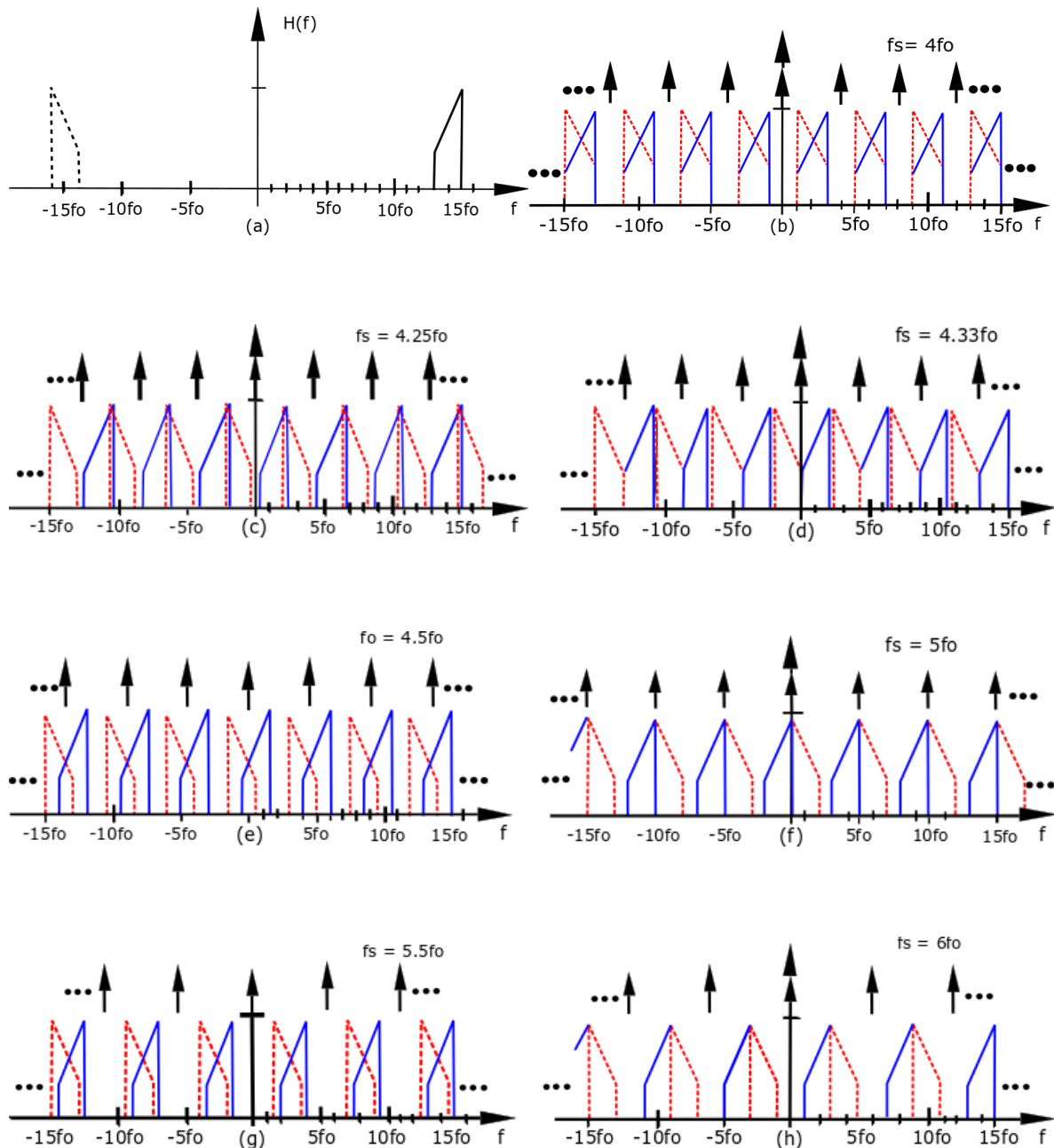
Debido a que $BT < 2f_0$, entonces la elección natural para la frecuencia de muestreo es $f_s = 2BT = 4f_0$, como se muestra en la Figura 20 (b). Sin embargo, se observa que esta elección de f_s produce superposición de espectro. Lógicamente, se aumenta f_s para eliminar la superposición del espectro. En la Figura 20 (c), se configura $f_s = 4.25f_0$. Tener en cuenta que todavía hay una superposición de espectro, pero si aumenta f_s a $4.33f_0$, como se muestra en la Figura 20 (d), se logra un espectro de frecuencia muestreada no superpuesta. Pero si se configura $f_s = 4.5f_0$, como se ilustra en la Figura 20 (e), nuevamente se encuentra la superposición del espectro. Usando el teorema de convolución, se puede determinar el rango de f_s que producirá un espectro muestreado no superpuesto. Este es el resultado dado por la ecuación 3 [42].

De la ecuación 3, para el ejemplo de espectro paso banda ilustrado en la Figura 20 (a), se nota que $2 \leq n \leq 7$, porque $f_h = 15f_0$ y $f_l = 13f_0$. Sea $n = 7$. Entonces, de la ecuación (3), se obtiene $4.29f_0 \leq f_s \leq 4.33f_0$. Observar las Figuras 20 (c) y (e) contienen cierta superposición de espectro para los resultados de muestreo $4.25f_0$ y $4.5f_0$. Mediante un análisis gráfico, se obtiene el rango de frecuencias de muestreo aceptables dadas por la ecuación (3). Al utilizar la frecuencia de muestreo $f_s = 4.33f_0$, que se encuentra en el extremo inferior de este rango como se observa la Figura 20 (d) no produce la superposición del espectro [42]

Ahora, con $n = 6$ en la ecuación (3); se obtiene un rango de frecuencias de muestreo aceptables por $5f_0 \leq f_s \leq 5.2f_0$. Una ilustración gráfica del rango de estas frecuencias de muestreo se ilustra en las Figuras 20 (l) hasta (g). Para $f_s = 4.5f_0$, que no pertenece a este rango, como se muestra en la Figura 20 (e), se obtiene un espectro superpuesto; pero para $f_s = 5f_0$, como se muestra en la Figura 20 (f), se nota que el espectro muestreado no se superpone. Se obtienen resultados desfavorables para $f_s = 5.5f_0$, como se muestra en la Figura

20 (g). Como se mencionó anteriormente, se puede ajustar cuidadosamente f_s para obtener el rango dado por la ecuación. (3) para $n = 6$.

De los resultados ilustrados en las Figuras 20 (h) e (i), se llega a la conclusión de que $6f_o \leq f_s \leq 6.5f_o$ es un rango aceptable para f_s . La ecuación (3) proporciona este resultado para la opción $n = 5$. Para $n = 4, 3$ y 2 en la ecuación. (3), se obtienen los resultados $7.5f_o \leq f_s \leq 8.67f_o$, $10f_o \leq f_s \leq 13f_o$, y $15f_o \leq f_s \leq 26f_o$, respectivamente. Las Figuras 20 (j) a (i) muestran opciones aceptables de f_s dentro de cada uno de estos rangos [42].



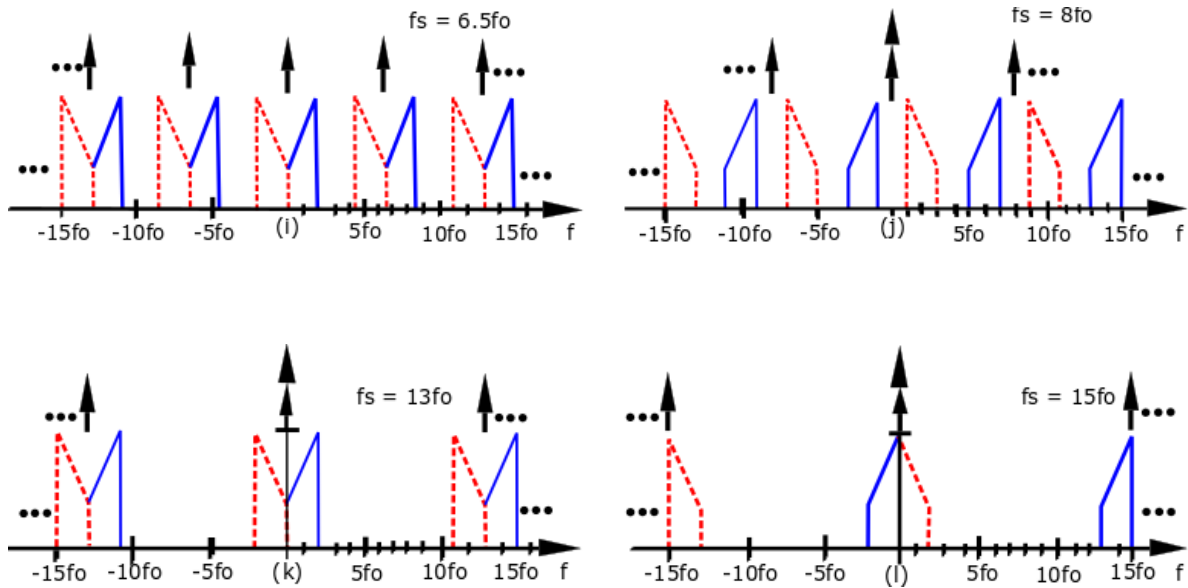


Figura 20. Transformada de Fourier con aliasing de una forma de onda de paso banda que se muestrea utilizando diferentes frecuencias [42].

Tener en cuenta que los resultados del espectro paso bajo de las Figura 20 (t), (j) e (i) son espectros invertidos con respecto a la Figura 20 (a). Estos resultados corresponden a los rangos de frecuencia determinados a partir de la ecuación. (3) para n par. Si se elige n impar, los resultados del espectro muestreado no se invierten, como se ilustra en las Figura 20 (d), (h), (i) y (k). También observe que, si f_s es una frecuencia de muestreo aceptable, entonces $p \cdot f_s$, donde p es un valor entero, también es una frecuencia de muestreo aceptable. Por ejemplo, si $f_s = 4.33f_0$, entonces $f_s = 8.66f_0$ y $f_s = 13f_0$ también son frecuencias de muestreo aceptables, como se muestra en la Figura 17 (d), (j) y (k). Esto se deduce de la periodicidad de la secuencia de frecuencia de muestreo en la función impulso. Se llega a la conclusión de que la determinación de los intervalos de frecuencia de muestreo que no producen resultados de espectro superpuestos no es trivial [42].

Una forma alternativa de examinar el teorema de muestreo paso banda es observar que para cada caso ilustrado en la Figura 20, la función propuesta paso banda con frecuencia central $14f_0$ se desliza mediante el proceso de submuestreo o de diezmo [42].

Debido a que el submuestreo resulta en el cambio de frecuencia de la señal paso banda, es posible posicionar el espectro desplazado mediante una elección apropiada de la frecuencia de muestreo f_s . Observe en la Figura 20 que la selección de f_s que satisface la ecuación 3 de modo que $n f_s = f_l$, donde n es un valor entero, cambia el intervalo de espectro de la señal paso de banda, $13f_0 < f < 15f_0$, al intervalo de frecuencia, $0 < f < 2f_0$, que generalmente se conoce como señal paso bajo equivalente a la señal paso banda. Las frecuencias de muestreo $f_s = 4.33f_0$ para $n = 3$, $f_s = 6.5f_0$ para $n = 2$ y $f_s = 13f_0$ para $n = 1$ satisfacen esta condición

y los resultados gráficos se muestran en las Figuras 20 (d), (i) y (k), respectivamente. Para muchas aplicaciones de procesamiento de señales, el método preferido es el submuestreo al equivalente de señal paso bajo. También se debe tener en cuenta que la selección de la frecuencia de muestreo tal que $n f_s = f_h$ también cambia la señal paso banda a un equivalente paso bajo, como se muestra en las Figuras 20 (f) y (i), pero el espectro está invertido.

También es posible seleccionar una frecuencia de muestreo que resulte en cambio de frecuencia a una frecuencia central cero. Note del desarrollo gráfico en la Figuras 20 que si $n f_s = (f_h - f_l) / 2$, es decir, la frecuencia central del espectro paso banda, entonces se traduce a frecuencia central cero. Esta selección de f_s siempre produce superposición de espectro y no satisface la ecuación (3). En la mayoría de los casos, la superposición del espectro es una operación irreversible. Sin embargo, las formas de onda que se muestrean a una frecuencia central cero siempre son recuperables si la señal paso banda se muestrea en cuadratura [42].

El muestreo de RF mediante submuestreo, o muestreo de paso banda, es un método potencial para muestrear la señal de RF utilizando CAD's. La frecuencia de muestreo se establece de modo que la frecuencia portadora ' f_c ' y la frecuencia de muestreo ' f_s ' satisfagan la relación $f_s = f_c / n$, en donde el factor de submuestreo n es un número entero positivo.

$$f_s = \frac{f_c}{n} \quad (n = 1, 2, 3, \dots) \quad (5)$$

La señal de RF que pasa a través de un filtro anti-aliasing, o un filtro de sistema cerca de la parte de la antena en este caso, se puede colocar en la señal de banda base (BB) con la frecuencia de muestreo que se observa en 5. Cuando la señal de RF cae a la señal de BB, aparece como se muestra en la Figura 21. Luego, la señal de BB se puede recuperar pasándola a través del filtro paso bajo [42].

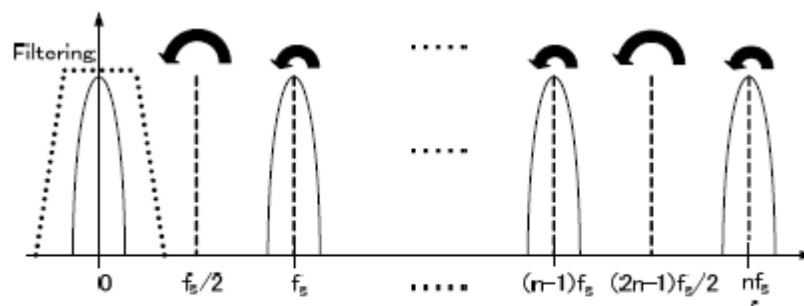


Figura 21. Proceso de corrimiento a banda base de la señal modulada [43].

3. Diseño del sistema.

3.1 Consideraciones de diseño.

Los sistemas de RMN utilizan generalmente un SDR que permita generar los pulsos de RF para la excitación de los núcleos, un receptor que permita detectar las señales que envían los núcleos después de aplicado los pulsos de RF, además de almacenar los datos que sirven para generación de imágenes.

Para este tipo de sistemas se requieren de FPGA's como DSPs. Estos sistemas embebidos permiten realizar tareas específicas con eficiencia, rapidez y alta capacidad de procesamiento. Para este caso de estudio se utiliza un FPGA ya que, por ser reconfigurable, brinda las herramientas que un SDR requiere para su diseño.

Para este trabajo se requiere llevar la señal de RF a banda base, generar secuencias de pulsos tanto para la señal de excitación como para los gradientes magnéticos y, utilizando un cpu, almacenar los datos.

Para la selección de los convertidores analógico y digital se toma en cuenta, que en los sistemas de RMN la fuerza del campo magnético y la frecuencia de Larmor, varía según el isótopo a excitar (ver Tabla 2.1). Entonces se tiene un CDA que es utilizado para la secuencia de pulsos con una tasa de datos de hasta 150MSPS, pero esta a su vez, se actualiza cada dos veces por flanco de subida logrando frecuencias con una tasa de 300MSPS. Por otro lado, con un CAD de 65 MSPS, donde se puede hacer submuestreo con un limitante en el ancho de banda, pero con una resolución de 14 bits suficientes para la aplicación en radios digitales utilizados en sistemas de RMN. El FPGA Arty A7 tiene un reloj de 100MHz, logrando así generar frecuencias de hasta 500MHz tanto interna como externamente, por lo tanto, podrá trabajar los dos convertidores sin inconvenientes.

En tiempo continuo una señal sinusoidal tiende a repetir su fase angular en el rango de 0 a 2π . En la implementación digital no es diferente. El acumulador de fase actúa como una rueda de fase en la implementación del DDS. El registro del acumulador de fase tiene una longitud de n bits y con cada ciclo de reloj a este se le suma un incremento de fase m. Si este valor de acumulador de fase se le asigna un dígito binario se tendría que para un ángulo de 0° un número binario de cero y para 360° un equivalente de 1111.111. La rueda de fase está determinada por la resolución del acumulador de fase (n) [44].

Para el diseño del DDS y DDC se utilizan IPs que pueden ser sintetizados por el FPGA. Sin embargo, cabe mencionar que se debe tener un amplio conocimiento en la utilización de estos componentes, ya que son elementos ya prediseñados para aplicaciones específicas, por lo

tanto, se podría ver como una desventaja; sin embargo, para reducir tiempo en diseño y programación las IPs son muy útiles, esto se puede ver como una gran ventaja. Con la ayuda de las GUI (Guide User Interface) el diseñador podrá modificar la IP de una forma útil y rápida, ya que en esta se puede modificar varios parámetros sin la necesidad de entrar en el ambiente de programación de VHDL o verilog. Además, en el ambiente de VIVADO, de XILINX, se tienen herramientas útiles que facilitan la programación de los FPGAs.

3.2 Arquitectura del Sistema.

El sistema contiene 5 módulos: transmisor, receptor, generador de pulsos, generador de pulsos de los gradientes y almacenamiento. En la Figura 22 se muestra la implementación del sistema en el FPGA y sus demás componentes.

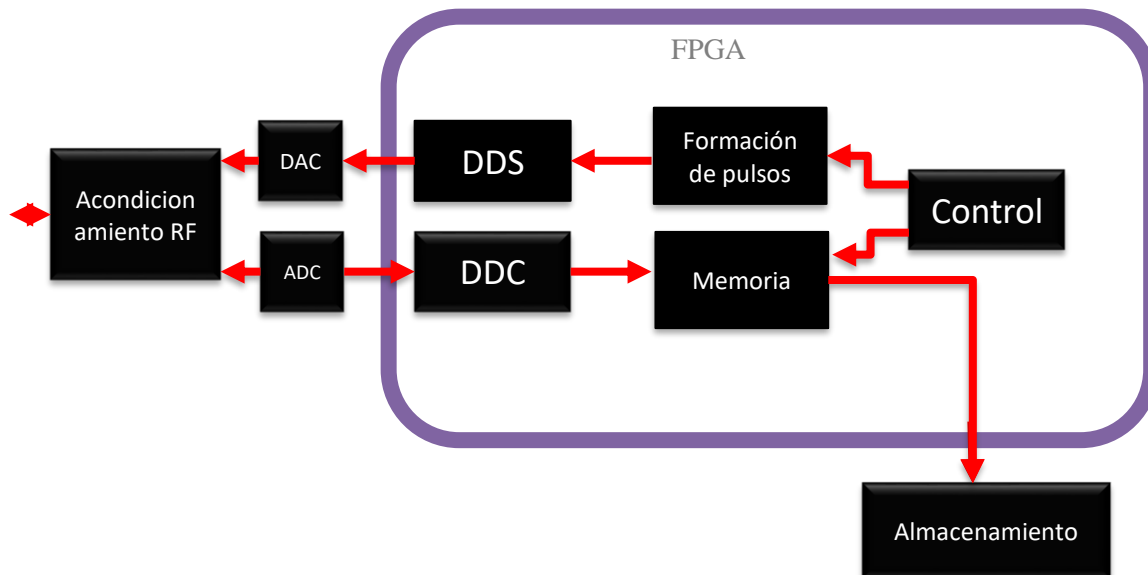


Figura 22. Arquitectura del sistema.

3.3 Transmisor.

El primer paso es diseñar el sistema de transmisión por medio de un DDS (Digital Direct Synthesizer) que genere las señales de RF. El núcleo del compilador Xilinx LogiCORE™ DDS implementa circuitos de alto rendimiento, generación de fase y compatibles con AXI4-Stream.

Utilizando una IP (intellectual property) llamada DDS de Xilinx, programada en VIVADO, se generan los pulsos de RF. Un DDS consta de un generador de fase ($\Delta\theta$), un acumulador de fase dado por $B_{\theta(n)}$, un lookup table y un reloj; como se puede observar en la Figura 23. Esta IP tiene varias aplicaciones, por ejemplo, en radios digitales y módems; SDR; convertidores descendentes/ascendente digitales, para estaciones base celulares y PCS; generador de frecuencias, etc [23].

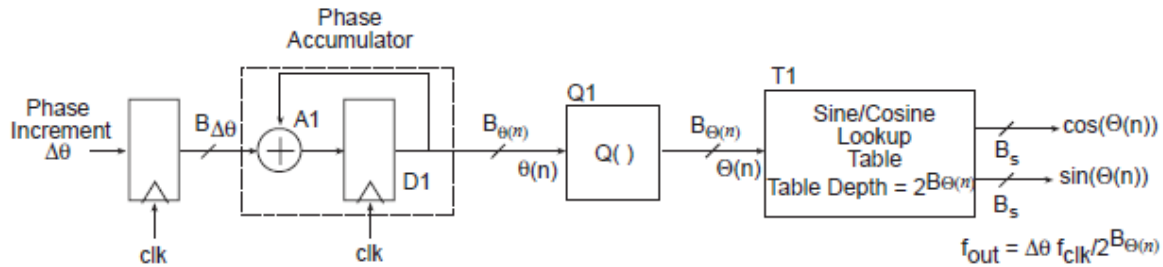


Figura 23. Esquema simplificado del DDS [23].

Los DDS utilizan un esquema de direccionamiento con una tabla de búsqueda apropiada para formar muestras de una senoide de frecuencia arbitraria. Si se requiere una salida analógica, el DDS presenta estas muestras a un convertidor digital - analógico (DAC) y un filtro paso bajo para obtener una forma de onda analógica con la estructura de una frecuencia específica. Por supuesto, las muestras también se usan comúnmente en el dominio digital. La tabla de búsqueda tradicionalmente almacena muestras espaciadas uniformemente de un coseno y un seno. [23]

La frecuencia de salida del bloque DDS se calcula de la siguiente manera:

$$f_{out} = \frac{f_{clk} \Delta\theta}{2^{B_{\theta(n)}}} \quad (6)$$

Donde f_{out} es frecuencia de salida deseada, f_{clk} es la frecuencia del reloj del sistema, ambas frecuencias dadas en Hz, $B_{\theta(n)}$ es el ancho de fase, es decir el número de bits del acumulador de fase y $\Delta\theta$ es el incremento de fase. [23]

Para el cálculo de la resolución Δf , se considera que el acumulador de fase es de 32 bits, por lo tanto:

$$\Delta f = \frac{f_{clk}}{2^{B_{\theta(n)}}} = \frac{4.3MHz}{2^{32}} = 1mHz \quad (7)$$

Además, la GUI del DDS proporciona las características siguientes: generación de fase para seno/ coseno, señales seno y coseno en cuadratura, opción de fase para el SFDR (Spurious Free Dynamic Range, por sus siglas en inglés), permite una resolución de 3 hasta 26 bits, entre otras cosas.

Por otra parte, por medio del GUI puede ser modificada la IP DDS según las necesidades del usuario, como por ejemplo, el reloj, el número de canales, el modo de operación, el ancho de fase y la resolución como se puede observar en la Figura 24.

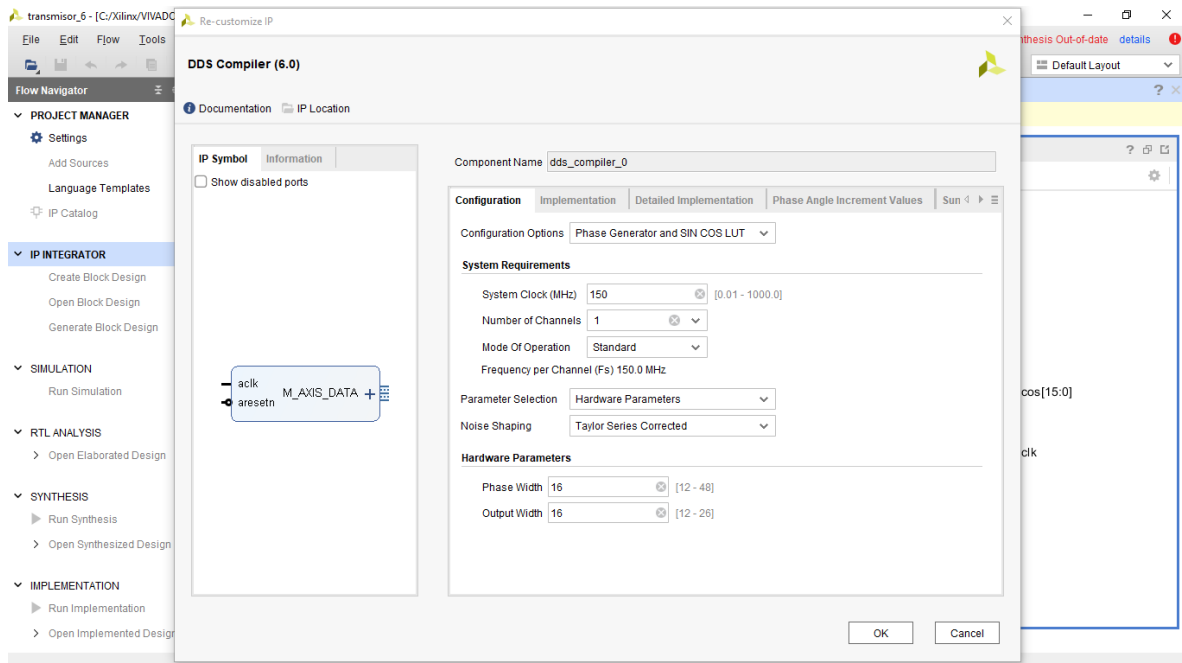


Figura 24. Interfaz de usuario del DDS.

El DAC requiere solamente de los datos a convertir y una entrada de reloj, por lo cual se necesita de un generador de reloj de frecuencia variable. Para este propósito se utiliza el Clocking Wizard, que es una IP que permite crear circuitos de reloj.

El núcleo del LogiCORE™ IP Clocking Wizard (v3.6 para ISE y v4.2 para las herramientas Vivado) facilita la creación de código fuente HDL para circuitos de reloj personalizados. El GUI permite la configuración del reloj y asimismo permite ajustar valores. Además de proporcionar un HDL para implementar el circuito de reloj deseado, el Clocking Wizard también ofrece un resumen de parámetros de tiempo generado por las herramientas de Xilinx ver Figura 25.

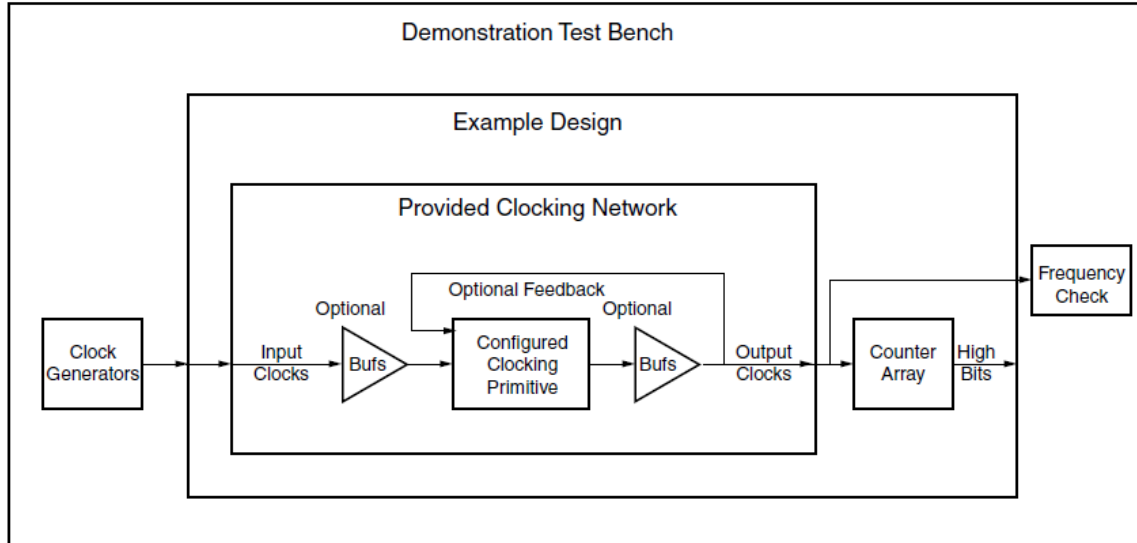


Figura 25. Esquemático del Clocking Wizard [45].

La IP DDS tiene en sus salidas formatos de complemento a dos y signada. Hay que contemplar estos formatos para el DAC AD9755, ya que este dispositivo solo permite un formato binario estándar. Entonces al DDS se le agrega un sumador que permita cambiar el complemento a dos a binario estándar según la hoja de datos del AD9755. De esta manera se agrega una IP ADD/SUB para hacer la función de cambiar el formato como se observa en la Figura 26.

Además, por medio de la GUI el usuario puede modificar el reloj, acepta hasta dos relojes de entrada, permite siete diferentes salidas de reloj, alineación de fase, minimiza el rendimiento de alimentación, entre otros; se puede analizar en la Figura 27 [45].

Por medio del Clocking wizard se generará un reloj de 150MHz, para uno tanto interno como externo que utilizará el DDS y el CDA, para generar la señal RF.

Para este caso de estudio se trabajará con señales seno a una frecuencia de 25MHz. Esta señal será modulada por pulsos cuadrados; cuya duración dependerá de las secuencias típicas aplicadas a los sistemas de RMN.

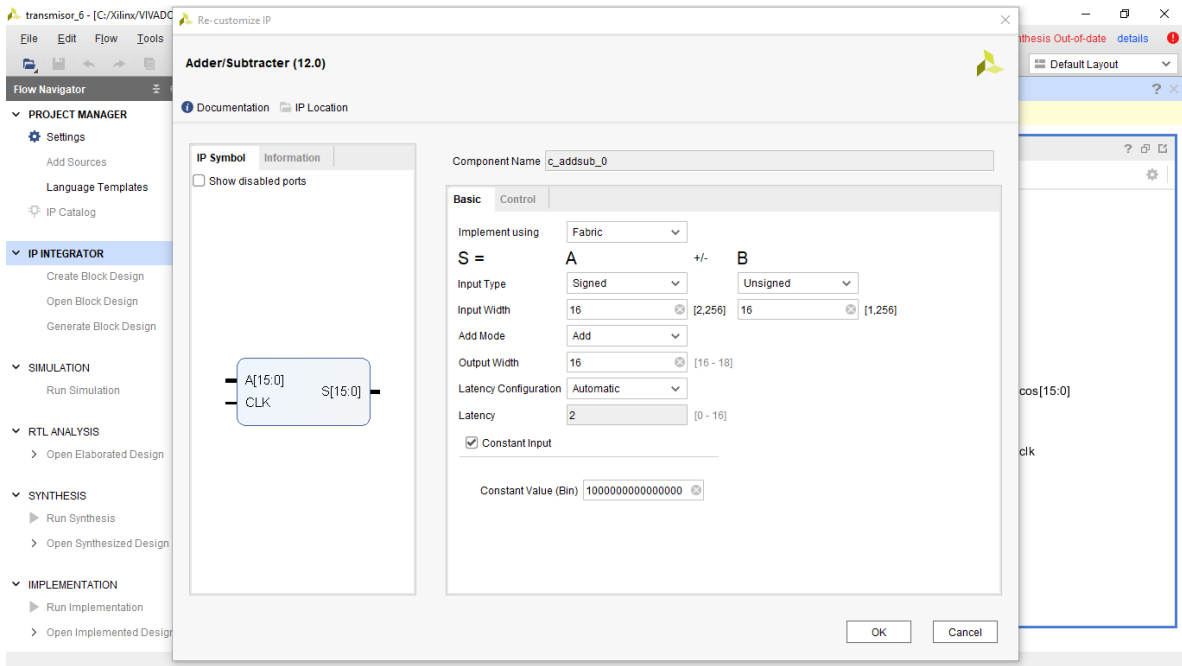


Figura 26. Asistente de programación de ADD/SUB [46].

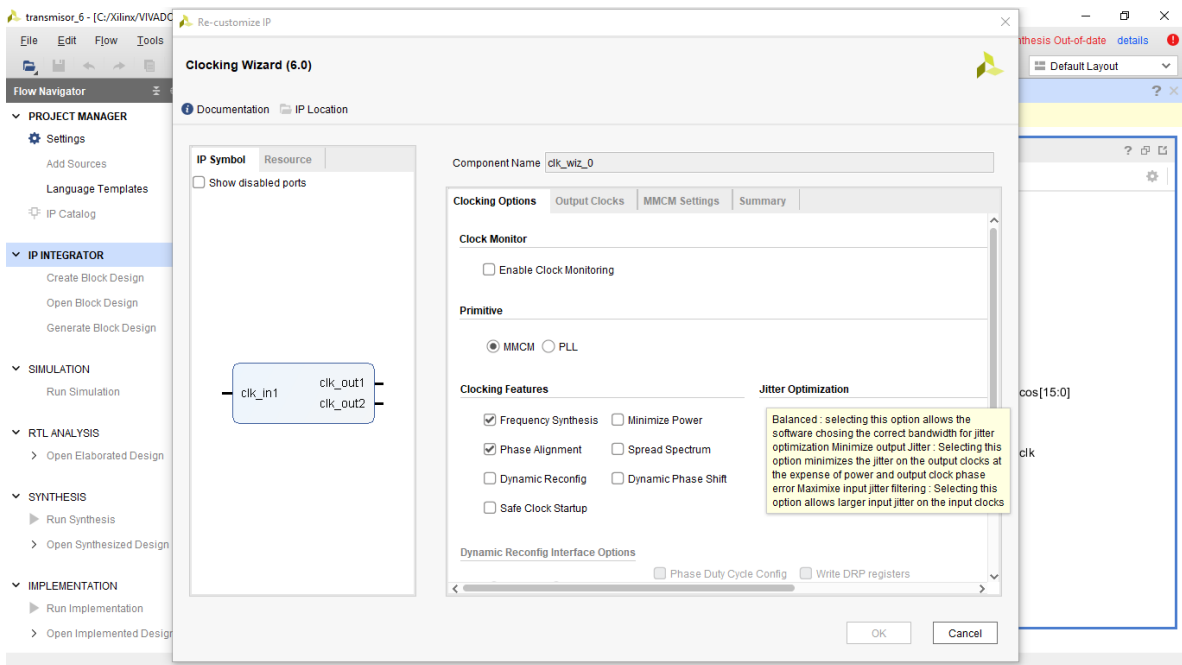


Figura 27. Asistente de programación del Clocking Wizard [45].

3.4 Receptor.

Para la etapa del receptor se requiere de un sistema que realice la conversión analógica a digital; un DDC (duc_ddc_compiler_0) que realice la demodulación de la señal y el submuestreo paso banda; un sistema FIFO (fifo_generator_1) para almacenar los datos del CAD; y un sistema UART (UART_0) para la transmisión de los datos a la PC como se mira en la Figura 28.

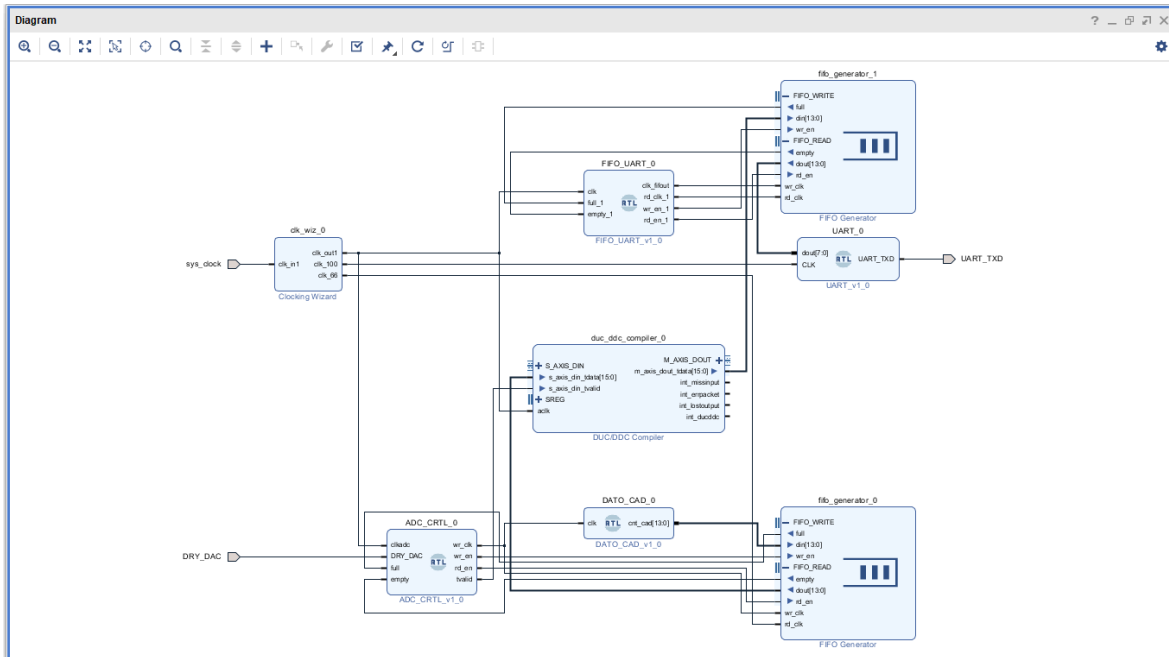


Figura 28. Diagrama completo del sistema de recepción.

El DDC está compuesto por un mezclador, un oscilador y un filtro pasa bajas. Por medio del IDE VIVADO se utilizan IP's que permiten realizar el diseño del DDC ver Figura 29.

En la primera etapa el CAD AD6644 realizará la digitalización de la señal analógica que proviene, para este caso, de la señal de RF. Posteriormente en el FPGA y por medio de la IP DDC se realizará la adquisición y manipulación de los datos obtenidos para la demodulación de la señal de RF para trasladarla a banda base y reducir muestreas.

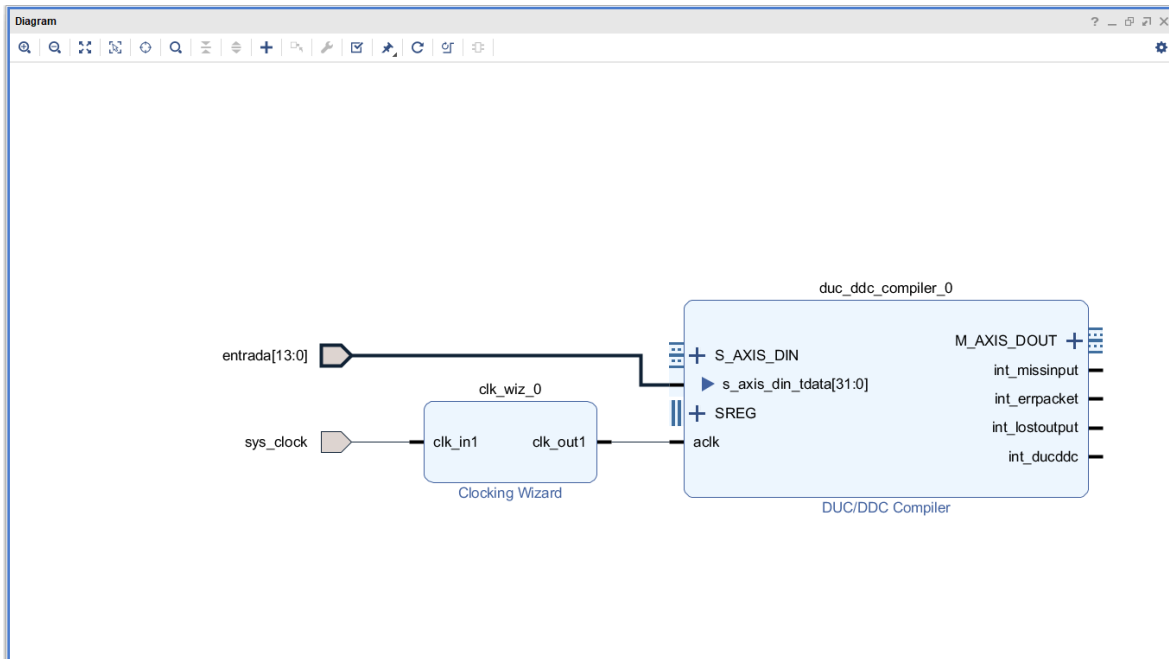


Figura 29. IP Digital Dow Converter.

El DDC es una IP del IDE de VIVADO que permite implementar módulos de convertidores ascendentes y descendentes digitales optimizados de alto rendimiento para su uso en estaciones base inalámbricas y otras aplicaciones. Además de una amplia gama de opciones de parámetros, las opciones de compensación de recursos están disponibles para adaptar el núcleo a una aplicación en particular [47].

El CAD realizará un submuestreo de la señal analógica adquirida, haciendo que disminuya su velocidad de muestreo sin perder información. El CAD tiene una frecuencia de muestreo de 65MSPS con una resolución de 14 bits. Por cada conversión analógica a digital el CAD manda una señal en DATA READY para indicar que el primer paquete de datos está listo.

Cada vez que el DATA READY (DRY_DAC) este en alto la FIFO (fifo_generator_1) almacena el dato binario posteriormente del CAD, para entonces enviarlo a la PC por medio del UART (UART_0).

3.5 Control de pulsos y gradientes magnéticos.

Como ya se ha indicado antes para la obtención de las imágenes se utilizan diversas secuencias de pulsos. Entre ellas se tienen el espín – espín, tal como se muestra en la Figura 30. En esta secuencia según [46] la duración de los pulsos es por lo general menor a 10 μ s, y la frecuencia de radiación está en el orden de 10² a 10³ MHz. Además, el intervalo entre los

pulsos T es de uno a varios segundos (Figura 30). Durante el tiempo T , los núcleos excitados emiten al relajarse una señal de radiofrecuencia en el dominio del tiempo, que se denomina señal de decaimiento libre de inducción (DIL, por sus siglas en inglés.) que se aproxima a cero después de una milésima de segundo.

Por ejemplo, para la secuencia espín - espín se requiere generar pulsos de 90° ; el tiempo de estos pulsos es de 1 a $10\mu\text{s}$ con un intervalo entre pulsos de $T = 1\text{s}$ [48].

Para la secuencia espín - eco se tiene dos pulsos de RF, estos son de 90° y 180° Figura 31. Por lo tanto, para el pulso de 90° se tiene una duración de $5\mu\text{s}$, el DIL un tiempo de 110ms , el pulso de 180° con una duración de $10\mu\text{s}$. Para el tiempo de adquisición de datos o tiempo de eco (TE) se tiene un valor de 221ms . La secuencia espín- eco tiene que aplicarse antes de que TR cumpla con 1s y se repita la secuencia.

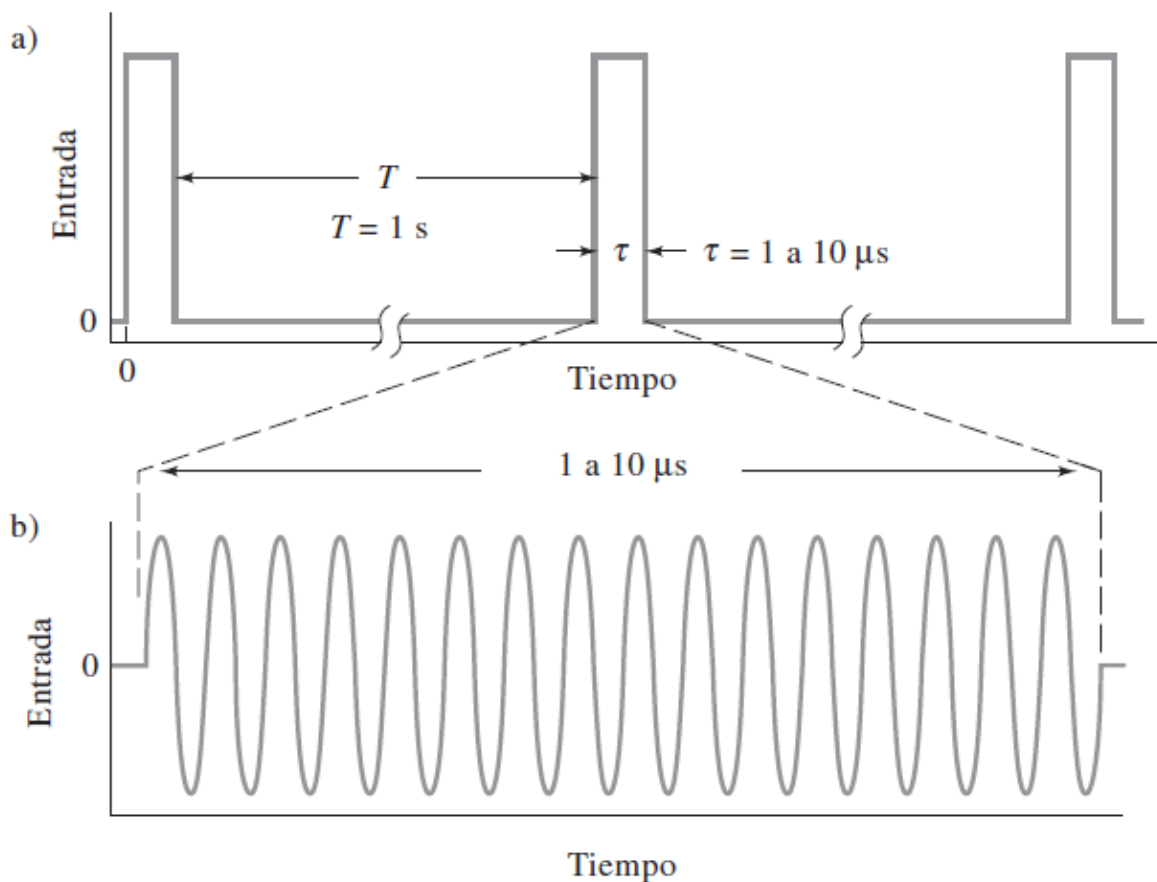


Figura 30. Diagramas de tiempo para la secuencia espín - espín [48].

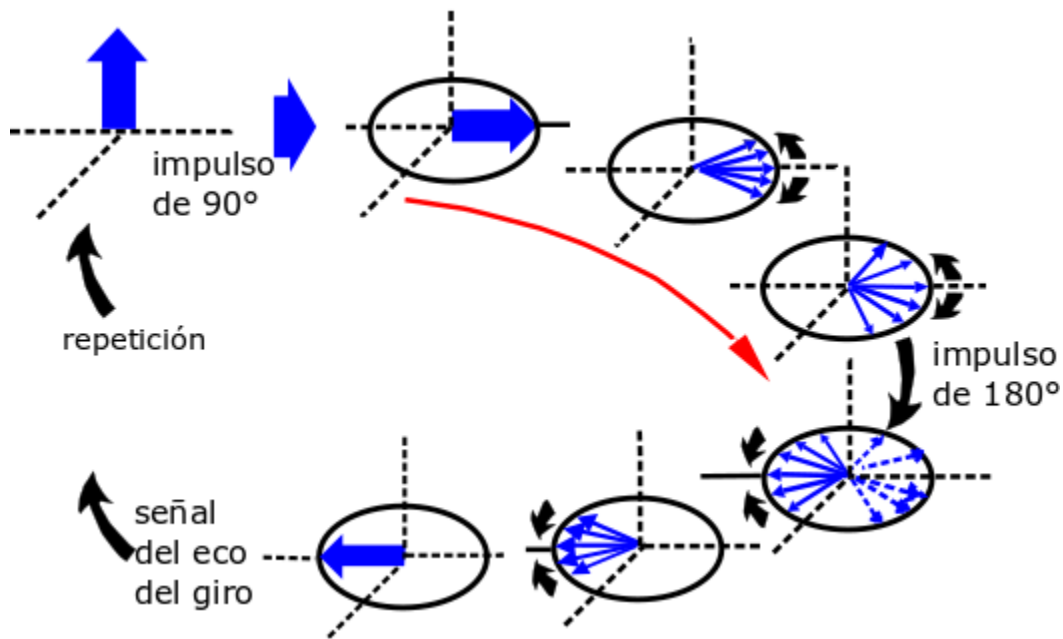
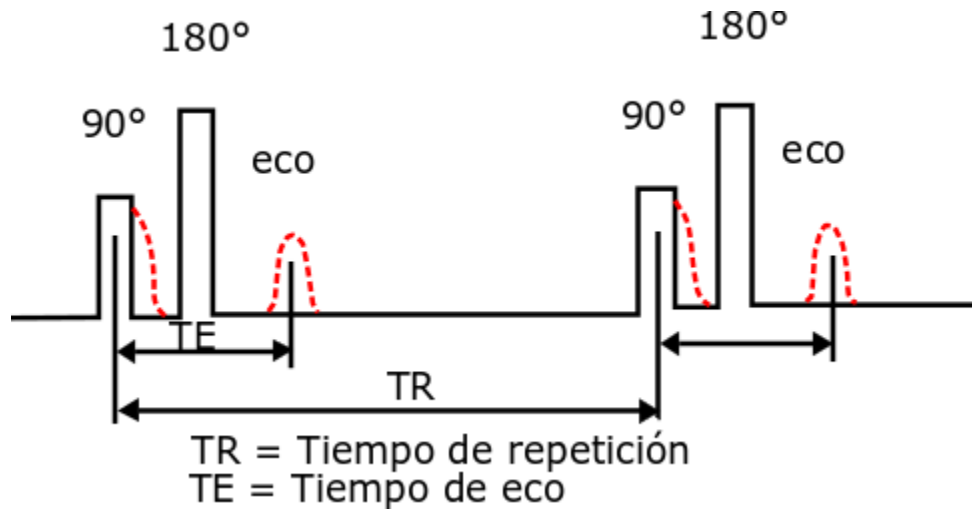


Figura 31. Secuencia espín – eco [11].

Otra secuencia importante en los sistemas de RMN es el eco – gradiente ver Figura 32. En este tipo de secuencia se genera un impulso de 90° acompañado con una aplicación del gradiente z (G_z) que selecciona una capa. La señal de DIL aparece, pero declina rápidamente debido a la combinación de la de la relajación transversal y el desfase por falta de homogeneidad del campo magnético B_0 . El gradiente y (G_y) es aplicado en el tiempo intermedio entre los pulsos de 90° y de 180° , este gradiente produce la codificación de fase para una dimensión espacial. Posteriormente se aplica un pulso de 180° para producir la señal

de eco. Mientras el refase se está llevando a cabo, el gradiente x (G_x) se enciende para la codificación de frecuencia de la señal para la segunda dimensión. Este juego de pulsos y gradientes se realiza periódicamente en un tiempo de repetición TR. Entre cada periodo el valor de G_y tiene un ligero aumento en su campo magnético, esto con el fin de adquirir suficientes datos [11].

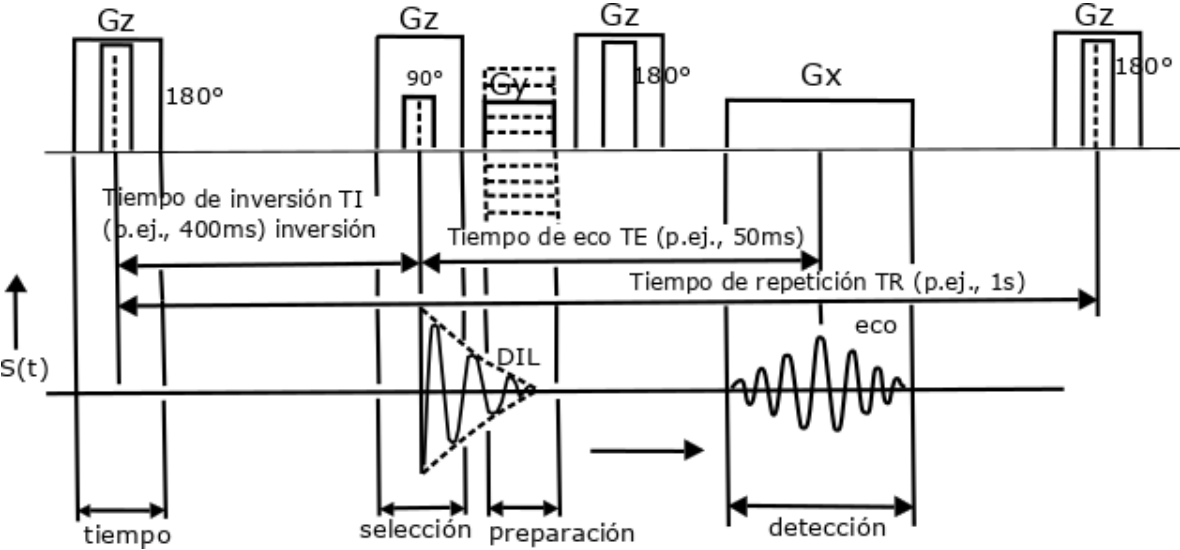


Figura 32. Secuencia eco gradiente [11].

Para la generación de los pulsos se utilizan contadores que determinan los diferentes tiempos de las secuencias antes mencionadas, tal como se establece en el diagrama de flujo de la Figura 33.

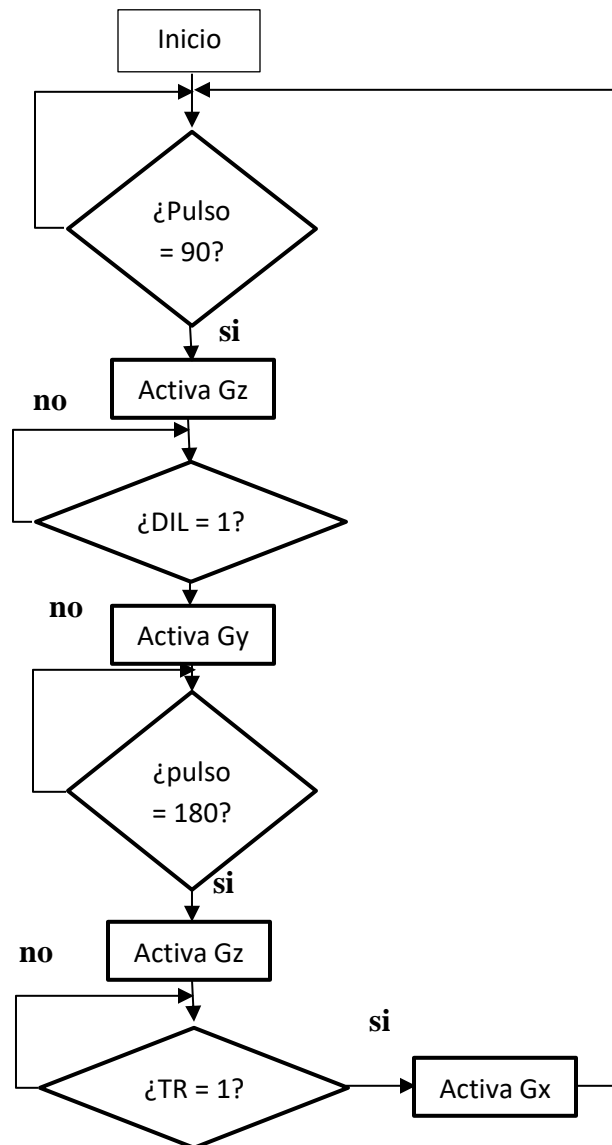


Figura 33. Diagrama de flujo de control de gradientes; eco-gradiente.

3.6 UART y Almacenamiento.

En la etapa de procesamiento de la señal se tiene el sistema UART (*Universal Asynchronous Receiver-Transmitter*, ver Figura 34) que su función es el de enviar los datos procesados del FPGA hacia el CPU en forma binaria. Este es un dispositivo que permite la transmisión (tx) y recepción (rx) de información, de forma serial y asíncrona. El receptor y el transmisor asíncronos universales se utilizan para la comunicación asíncrona de datos. El UART se puede usar para controlar el proceso de reconfigurar los datos paralelos de la PC en datos en

serie que se puedan transmitir. Se compone de un módulo receptor y un módulo transmisor. UART ha sido una herramienta de entrada / salida importante durante décadas y todavía se usa ampliamente [49].

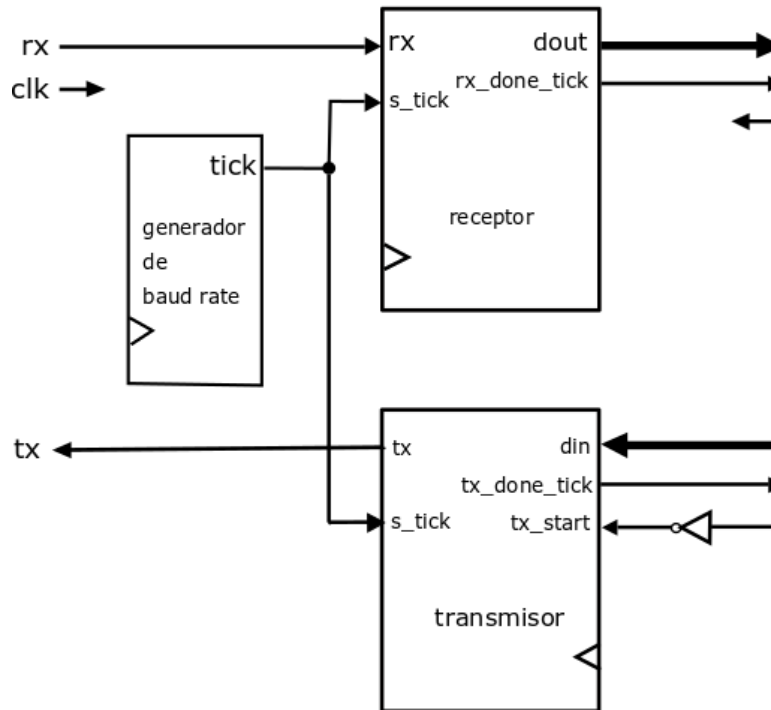


Figura 34 Diagrama de bloques del UART.

El diagrama de bloques del UART consta de tres componentes principales: control del transmisor (tx_done_tick, tx_start), control del receptor (dout, rx_done_tick) y generador de velocidad de transmisión (Baud Rate, s_tick). El Baud Rate son los bits por segundo que se están transmitiendo y se calcula con la siguiente formula:

$$t_b = \frac{1}{baudrate} [s] \quad (8)$$

Las velocidades del UART abarcan desde los 9600, siendo el más usado por los usuarios, hasta los 921600 bauds.

Al transmitir, el UART toma ocho bits de datos paralelos, convierte los datos a un flujo de bits en serie que tiene un bit de inicio (lógica '0'), 8 bits de datos y un bit de stop (lógica '1'). Al recibir, el UART detecta inicialmente un bit de inicio, luego recibe un flujo de 8 bits de datos y presenta los datos en paralelo cuando detecta el bit de stop como se ve en la Figura

35. Como no se transmite ningún reloj, el UART debe sincronizar el flujo entrante de bits con el reloj del FPGA [50].

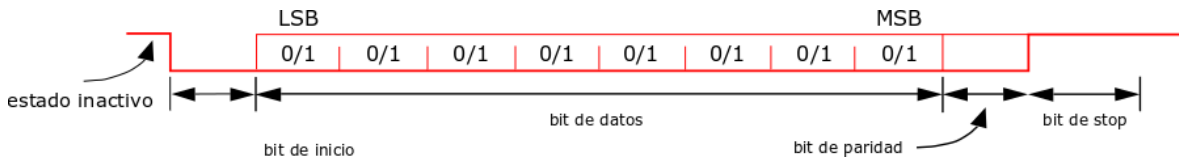


Figura 35 Diagrama de tiempo del UART.

Tomar en cuenta que la velocidad del receptor es 16 veces mayor al del transmisor. Por lo tanto, si se tiene una velocidad de transmisión de 9600 baudios, el receptor debe tener $16 \times 9600 = 153,600$ baudios. [39]

El UART (UAR_5_0) solo puede mandar un paquete de datos de 8 bits. El CAD genera 14 bits de resolución. Por lo tanto, se estarán concatenando dos paquetes de 8 bits para completar la palabra de 14 bits. Este procedimiento se realiza en la PC. Este paquete de datos de 8 bits se estará almacenando en un sistema FIFO (fifo_generator_0), esto con el fin de desfasar el tiempo de conversión del CAD y el tiempo de transmisión (UART_TXD) de datos hacia la PC. Esto quiere decir que la velocidad de conversión del CAD es mayor que la transmisión del UART; por lo tanto, se utiliza una IP FIFO GENERATOR (fifo_generator_0) para el desfase de tiempo como se observa en la Figura 36.

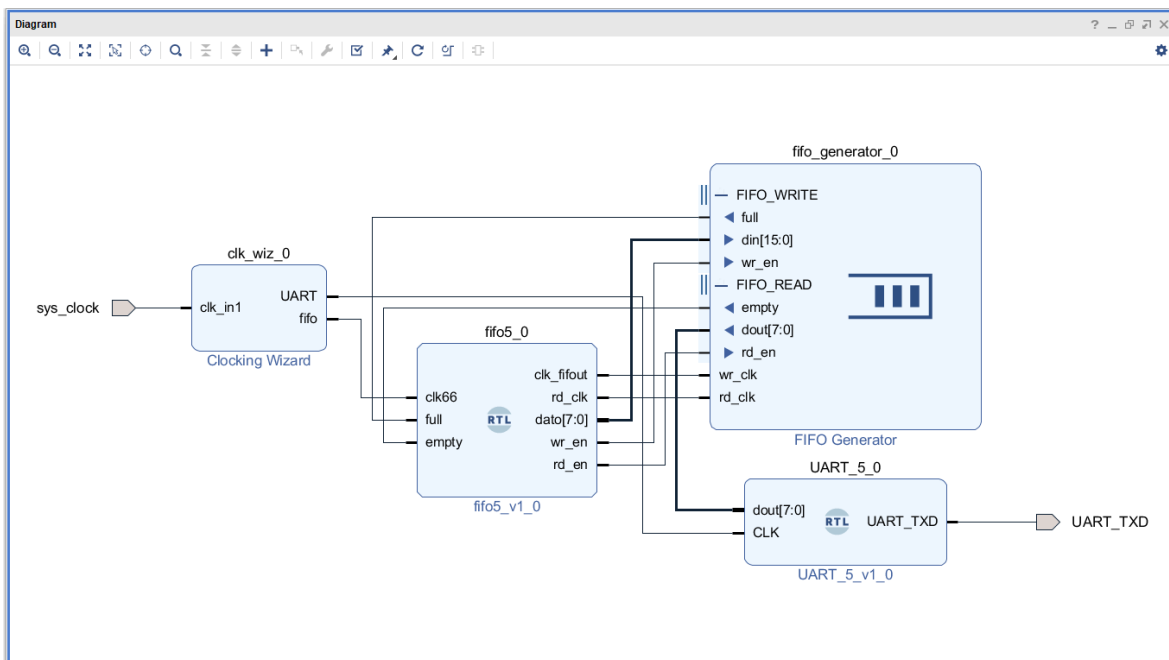


Figura 36. Diagrama de IP FIFO.

Dentro del ambiente de las IP se tiene FIFO GENERATOR que genera filas de memoria ideales para almacenamiento de datos. Este núcleo ofrece un máximo rendimiento de hasta 500MHz con recursos mínimos. Por medio del IDE VIVADO el usuario tiene la facilidad de modificar, personalizar la estructura, incluyendo el ancho y la profundidad de la memoria, los indicadores de estado, el tipo de memoria y las relaciones de los puertos de escritura y lectura, así como también los relojes de escritura y lectura. Esta FIFO en particular proporciona un ancho de palabra para escritura de datos y lectura de datos 1024 con una profundidad de 131072 respectivamente [51].

El sistema búfer FIFO proporciona más espacio de almacenamiento y reduce aún más la posibilidad de saturación de datos. Podemos ajustar el número deseado de palabras en FIFO para satisfacer la necesidad de procesamiento del sistema. La señal rx-ready-tick está conectada a la señal wr del FIFO. Cuando se recibe una nueva palabra de datos, la señal wr se establece en alto un ciclo de reloj y los datos correspondientes se escriben en FIFO. El sistema principal obtiene los datos del puerto de lectura de FIFO. Después de recuperar una palabra, se establece en alto rd en un ciclo de reloj de la FIFO para eliminar el elemento correspondiente. La señal vacía de la FIFO puede usarse para indicar si hay alguna palabra de datos recibida disponible. Se produce un error de saturación de datos cuando llega una nueva palabra de datos y la FIFO está llena [52].

La configuración de la IP FIFO permite escribir en memoria una palabra de 14 bits y leer 8, ya que el UART solo permite la transmisión de 8 bits. Hay que tomar en cuenta que la IP FIFO lee un paquete de 8 bits por periodo de reloj, por lo tanto, requiere de dos periodos de reloj para transmitir el dato completo de 14 bits. Por medio de MATLAB se crea un algoritmo de control que permite adquirir los datos que provienen del UART. Este dato viene en paquetes de 8 bits. Por otro lado, se requerirá concatenar las dos palabras para generar el dato completo de 14 bits y graficar la señal de RMN.

Además, a cada palabra de 8 bits se le concatenan un bit de inicio y un bit de paro, que permiten a la PC identificar el inicio y el final de un paquete de datos. Se desarrolla un algoritmo en MATLAB que permite identificar estos bits y puede generar un paquete de 14 bits de datos para ser graficados y verificar el desempeño del sistema.

4.Implementación y Pruebas.

SISTEMA DE TRANSMISION.

En la primera etapa se diseñó el transmisor digital en el FPGA. Como se estado mencionando esta etapa contiene un reloj una IP DDS y un sumador. Se utilizaron herramientas como el IDE VIVADO y las IP, estas facilitan la programación en VHDL con sus ventajas y desventajas. Como desventaja hay que leer los manuales que son muy extensos; la ventaja es su fácil manejo y rapidez en la implementación de diseño. Hay que tener un amplio conocimiento en el manejo de estas IPs, además de saber qué tipo de FPGA se tiene que trabajar ya que hay variedad en tarjetas de desarrollo. Para el diseño de este proyecto se revisaron puntos como puertos PMOD de entrada/ salida, periféricos (UART), alimentación, capacidad de memoria (ROM, RAM), compatibilidad con el software IDE VIVADO y velocidad de reloj. Existen otras tarjetas con diferentes características, algunas contienen periféricos como CAD, ethernet, video, audio, OLED, GPIOs (switch y led -rgb), HDMI, display 7 segmentos, entre otros.

También hay que saber que para interactuar con las IP y con un programa en VHDL, el código que se haya diseñado tendrá que convertirse a un sistema RTL (Register Transfer Level) para conectar los pines. Por ejemplo, en la Figura 37 el bloque reset_2_0 en un diseño propio en VHDL, este programa permite realizar un reset (pul_90) en el DDS (aclken).

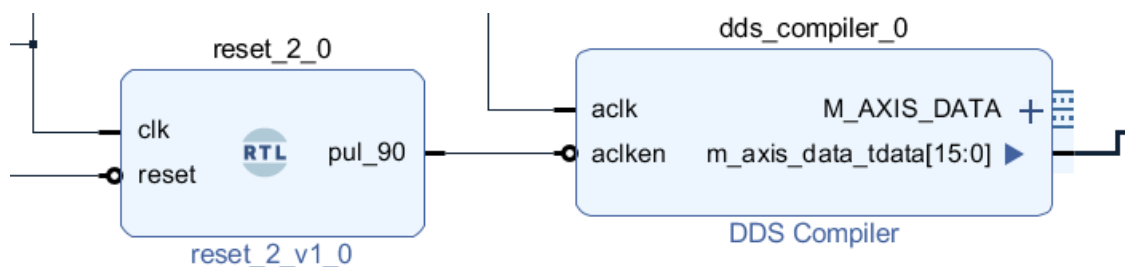


Figura 37. Modificación de un programa en VHDL a bloque.

A continuación, se utiliza la IP DDS para generar los pulsos de RF. Por medio del GUI se configuran los parámetros requeridos para generar las ondas senoidales. A continuación, en

la Figura 38, se muestran los bloques del DDS (dds_compiler), ADD (c_addsub_0). y el Clocking Wizard (sys_clock - clk_wiz_0),

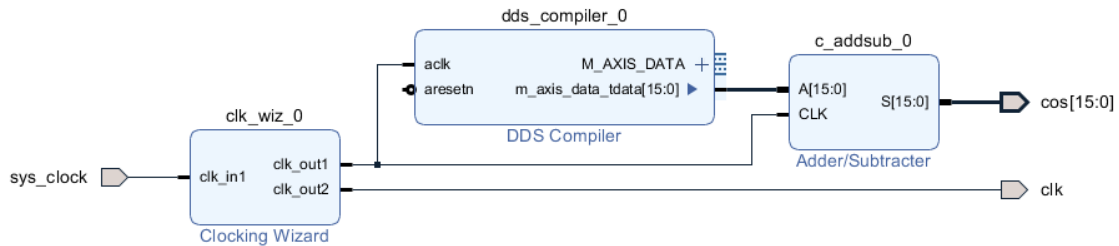


Figura 38. Diagrama de IP del Transmisor.

El Clocking Wizard permite utilizar la señal del reloj interna del FPGA. Esta señal de reloj se conecta al sistema DDS para generar la frecuencia de la señal senoidal. La señal se puede ver en un sistema binario que sale por el puerto M_AXIS_DATA, por ejemplo, de tamaño de 32 bits. Siendo el valor menor de la señal senoidal una palabra de 32 bits con 0000...0 y el máximo valor de la señal de 1111...1. Por lo tanto, al tener estos valores se dispone a realizar la conversión digital analógico por medio de los puertos de salida del FPGA y conectarlos a una tarjeta de desarrollo AD9755. Esta tarjeta será la encargada de realizar la conversión con las muestras binarias que el FPGA envíe.

Las señales senoidales están moduladas con una señal cuadrada. Las ondas senoidales se generan con una frecuencia típica de los sistemas RMN. A continuación, se muestra un diagrama de simulación del funcionamiento del sistema de trasmisión.

En la Figura 39 se muestra el valor del reloj interno del FPGA que es de 100MHz (sys_clock), así como también el reloj que genera el Clocking Wizard (clk) de 150MHz. Para este caso se muestra una señal senoidal de 14 bits (cos[15:0]) a 20MHz.

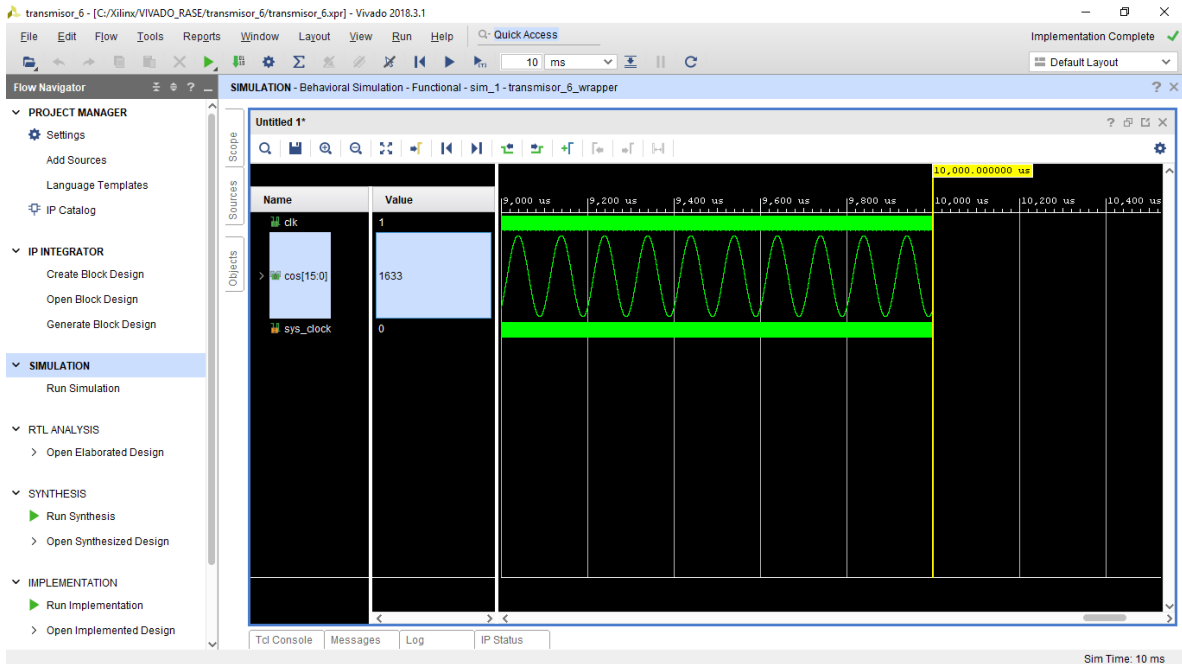


Figura 39. Simulación del sistema de transmisión.

El DDS codifica la senoide en palabras de 14 bit en complemento a dos, habrá que hacer la conversión a binario estándar (sin complemento a dos), ya que el convertidor digital analógico no reconoce este formato. En la Figura 26 se muestra una IP ADD que permite realizar la conversión de complemento a dos a formato estándar binario sumándole un 1 al bit más significativo (10000000000000). Por otro lado, hay que tener en cuenta que el DDS y el ADD deben compartir la misma frecuencia de reloj. De esta manera al obtener la palabra de 14 bits del DDS en formato estándar, se conecta el FPGA a través de los PMOD JA y JD con el CDA para realizar la decodificación, teniendo en cuenta el teorema de Nyquist. A continuación, se muestran diferentes pruebas a frecuencias de muestreo variables para observar el funcionamiento del DDS y el CDA.

En la Figura 40 se muestra una señal de 20MHz con una frecuencia de muestreo de 40MHz.

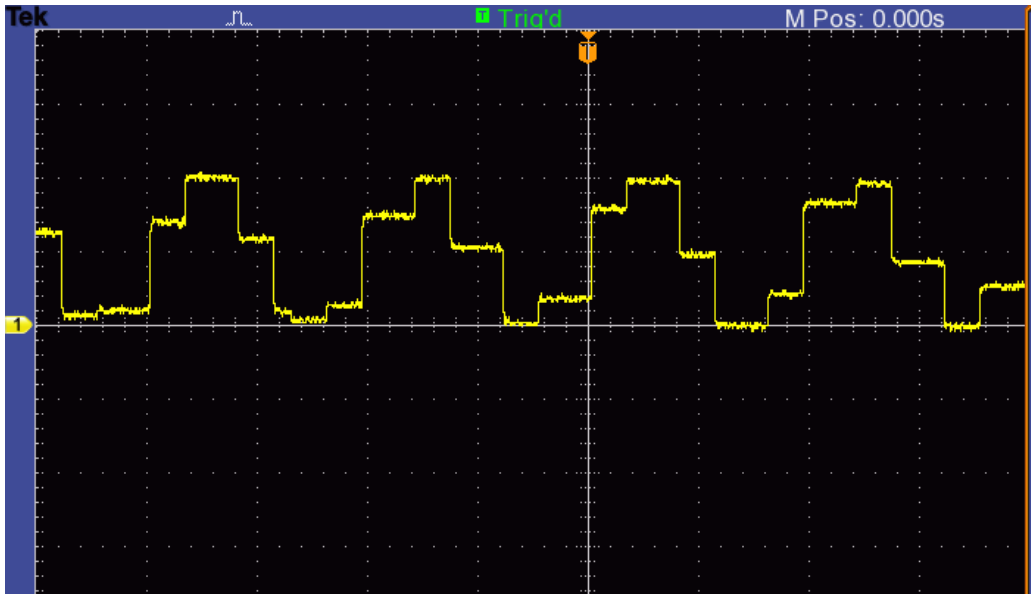


Figura 40. Muestreo de una señal senoidal de 20 MHz con un Fm de 40MHz.

Aplicando sobremuestreo, la señal tiene mayor número de muestras, por lo tanto, se tiene una mejor resolución de cuantificación de la señal ver Figura 41.

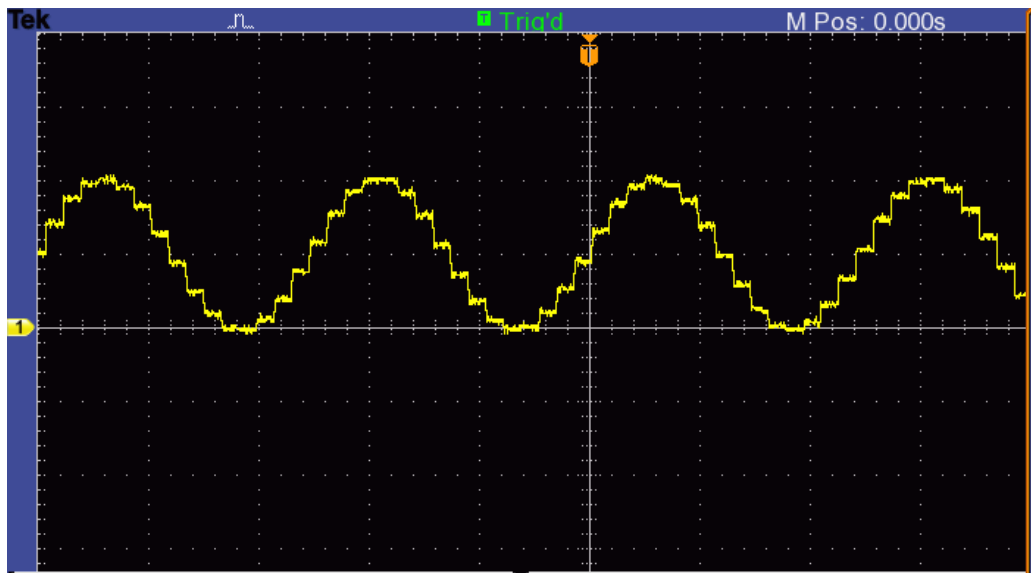


Figura 41. Señal senoidal de 20 MHz con un Fm de 60MHz.

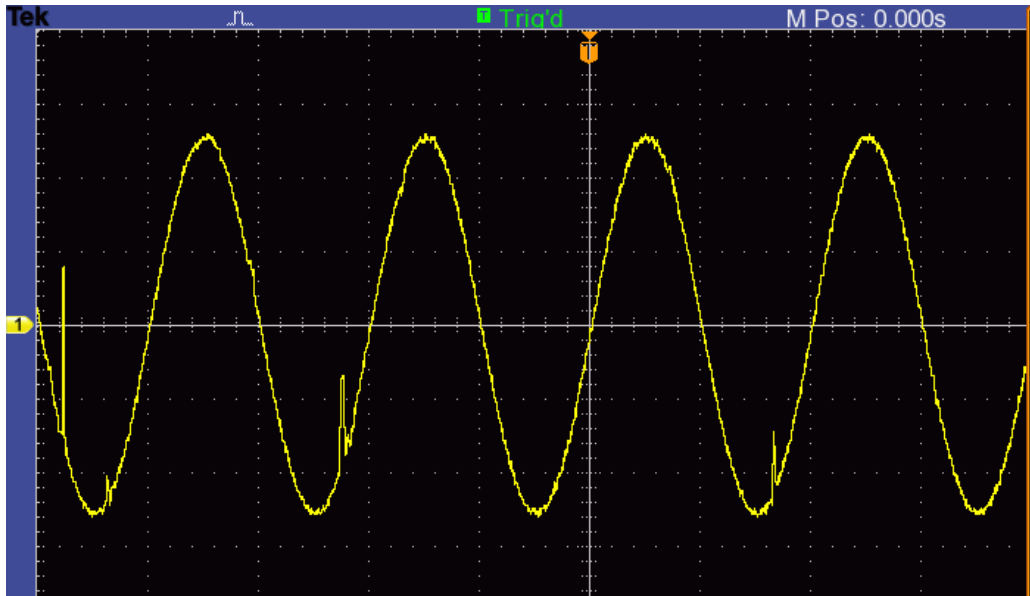


Figura 42. Señal senoidal de 20 MHz con un Fm de 150MHz.

Para el transmisor, según la teoría, se requiere de una modulación por pulsos on – off. En la Figura 43 se muestran los pulsos que se generan con el DDS para la señal de RF. El periodo de los pulsos dependerá del tipo de secuencia que se esté aplicando, así como la frecuencia para el tipo de molécula y campo principal utilizado, B_0 .

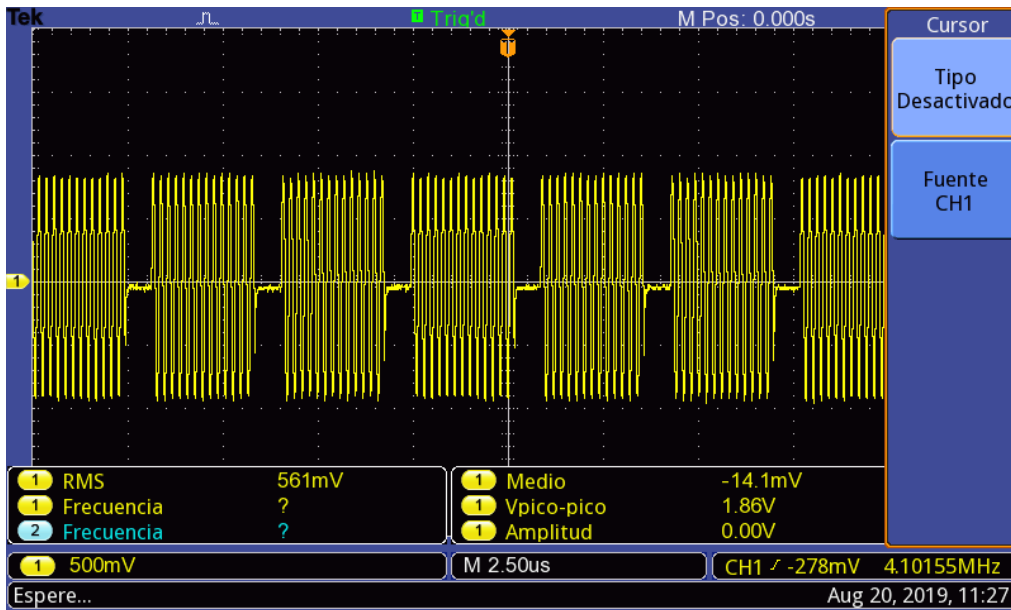


Figura 43. Pulsos de 4MHz modulando una señal de 20MHz.

SECUENCIA ESPIN - ESPIN.

En las siguientes Figuras se muestran las diferentes secuencias utilizadas en los sistemas de RM. En la Figura 44 se muestra la secuencia espín eco con una señal de RF a 90° con un tiempo de $6\mu\text{s}$ (Δx) y un periodo entre pulsos de $106.8\mu\text{s}$. En este tiempo se puede adquirir la señal de FI en forma modulada que es con la que los sistemas de RM trabajan para generar las imágenes del tejido. Esta secuencia se puede observar en la Figura 43.

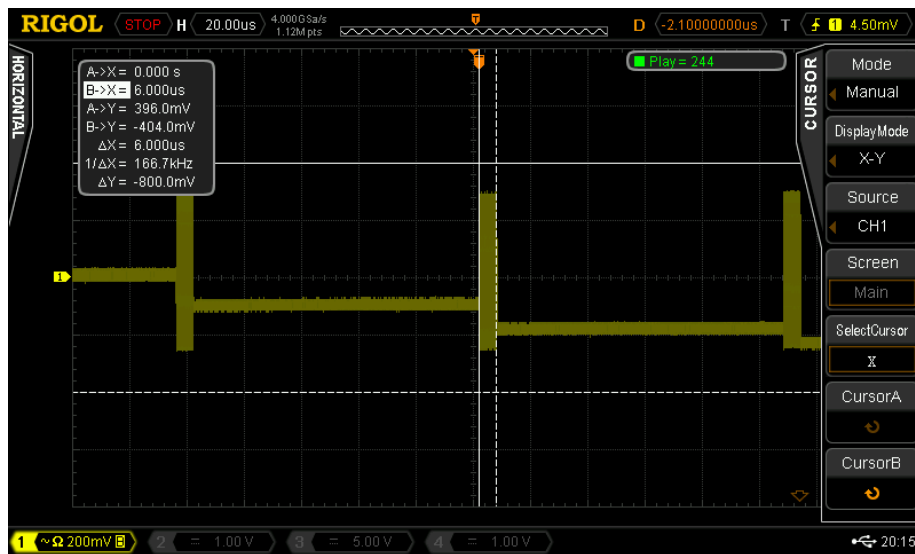


Figura 44. Secuencia espín – espín.

Cada pulso de 90° modula una señal de 20MHz . La señal moduladora son pulsos diseñados con un algoritmo en VHDL que conectado al DDS permite cambiar los tiempos de la señal RF. Esto confirma la teoría que el sistema es reconfigurable y el FPGA permite este tipo de diseños utilizando pocos recursos de la tarjeta. En la Figura 45 se observa el periodo entre pulsos ($\Delta x = 106.8\mu\text{s}$) de la secuencia espín eco mostrada por los cursores del osciloscopio.

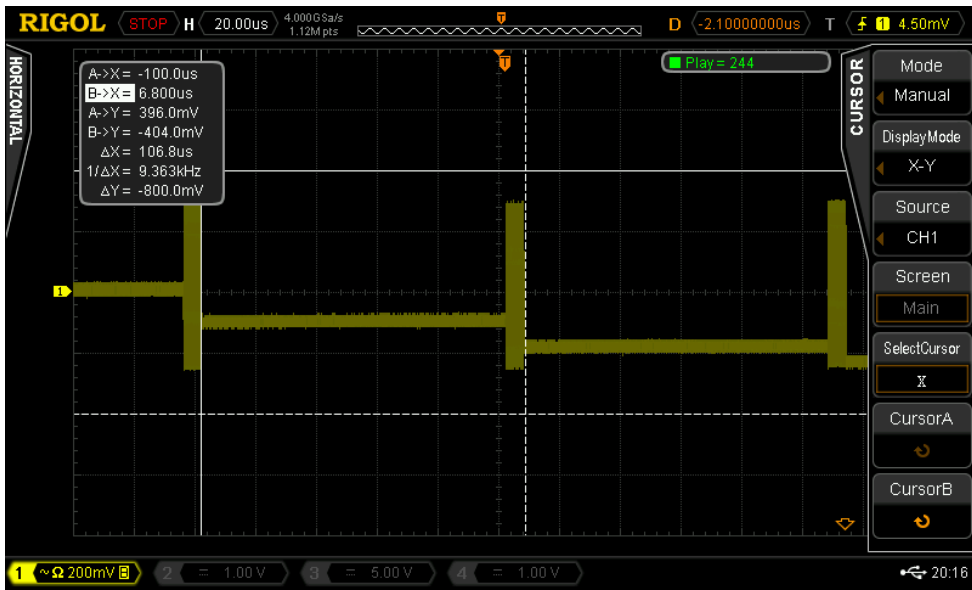


Figura 45. Periodo entre pulsos de 90°.

En la Figura 46 se demuestra que en cada pulso de 90° contiene la señal senoidal a una frecuencia de 20MHz.

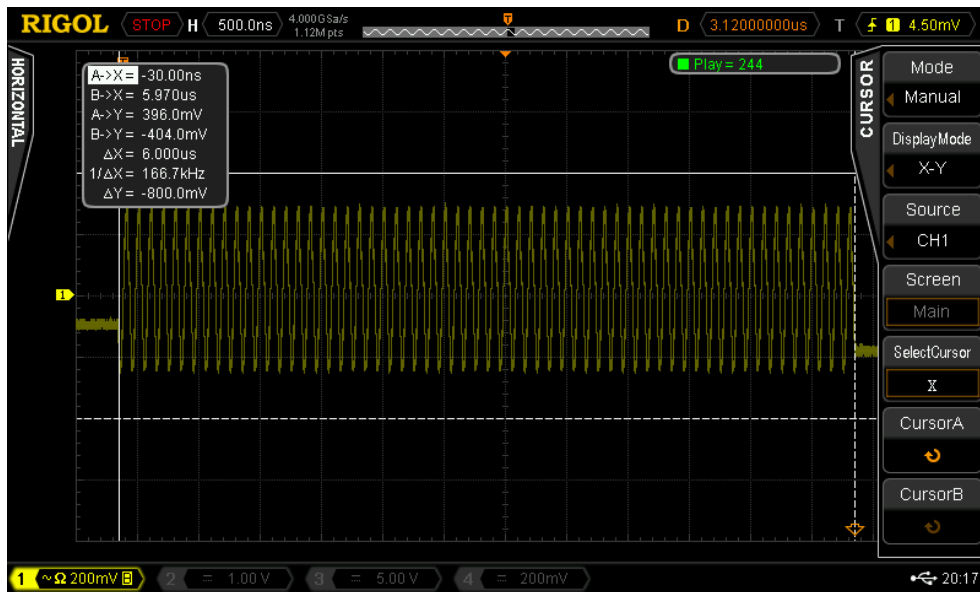


Figura 46. Señal senoidal modulada de 90°.

SECUENCIA ESPIN - ECO.

Para la secuencia espín - eco, se tiene un pulso de 90° , seguido de un pulso de 180° y el tiempo de adquisición de datos.

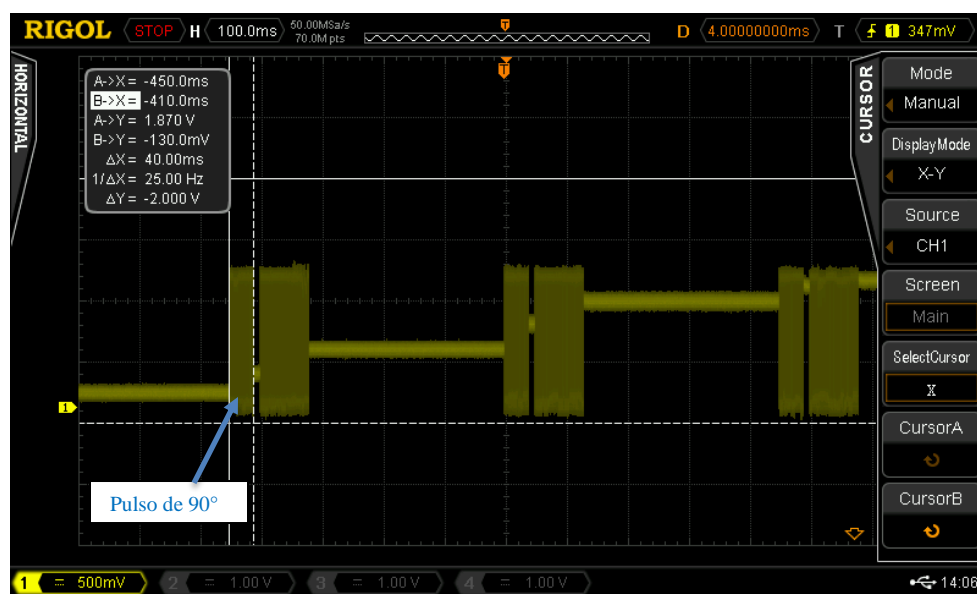
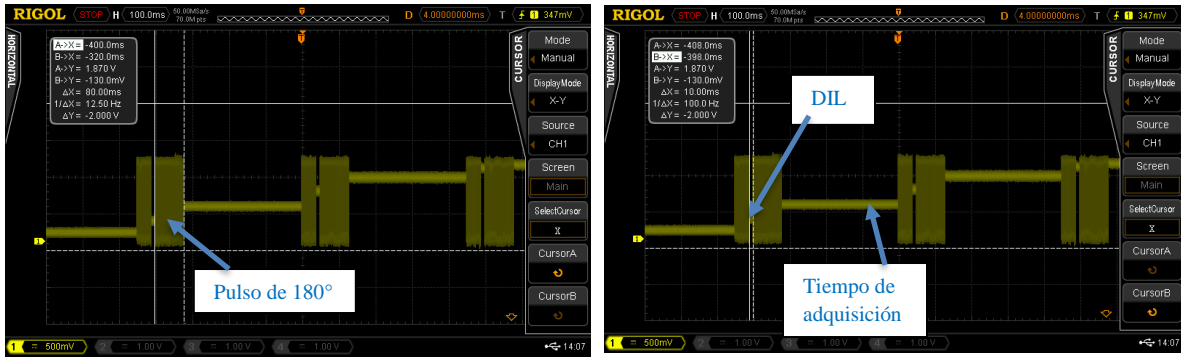


Figura 47. Secuencia spin – eco.

En la Figura 47 se muestra la secuencia espín eco con un pulso de 90° con un tiempo de 40ms. Después de un tiempo de 10ms, que es el tiempo de DIL (Figura 48 (b)), se aplica el pulso de 180° , que según la teoría tiene que ser el doble de tiempo que el pulso de 90° , por lo que será de 80ms como se observa en la Figura 48 (a). Posteriormente se tiene el tiempo de adquisición que es también el tiempo de eco. Hay que tomar en cuenta que este tiempo tiene que ser adecuado para poder enviar los datos por el UART hacia la PC.

Por otro lado, hay que recordar que al aplicar un pulso de 90° , las moléculas empiezan a realizar un desfase entre ellas ocasionado poco contraste en la imagen. Para tener un mejor realce en la imagen, se aplica un pulso de 180° de tal forma que se compense el desfase previo y la imagen sea más nítida.



(a)

(b)

Figura 48. Pulso de 180° con un tiempo de 80ms (a), tiempo de DIL de 10ms (b).

SECUENCIA ECO DE GRADIENTES.

Por último, la secuencia eco de gradientes, como se menciona en la teoría, permite tener un gradiente G_x que es la selección de plano de corte, G_y representa el gradiente de codificación de fase y G_z representa el gradiente de codificación de frecuencia.

Tomando como referencia la Figura 10, se generan los diferentes pulsos para los diferentes gradientes; tomando los tiempos de 90°, 180° y el tiempo de DIL.

En la Figura 49, la señal de tono amarillo muestra las señales de RF 90° y 180° respectivamente; las señales de tono azul muestran los pulsos de los gradientes.

G_x en la Figura 49 abarca un tiempo de 100ms, que es el tiempo de adquisición de datos. En la Figura 50 se muestran los tiempos de los pulsos de control de los gradientes.

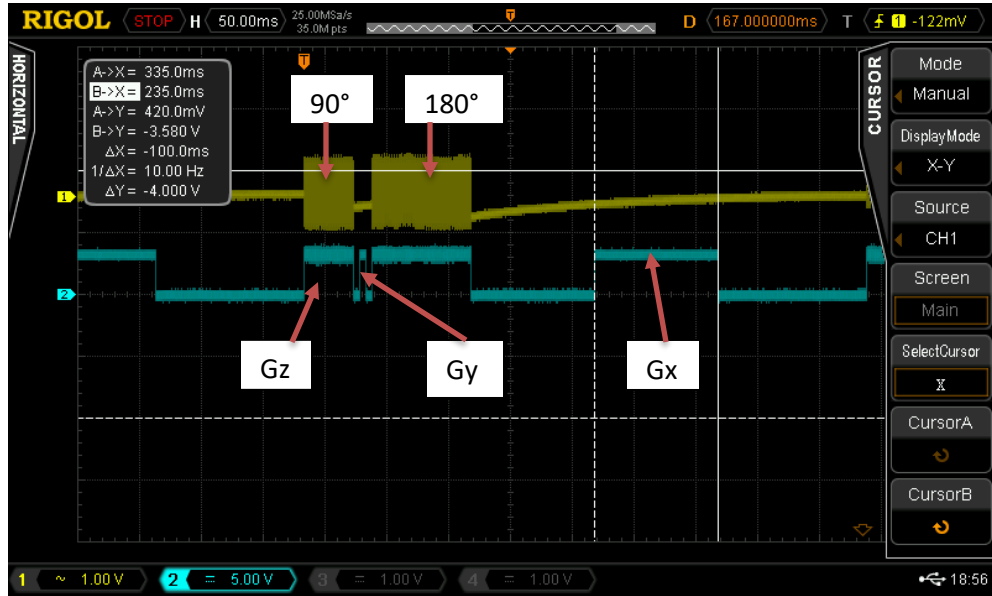
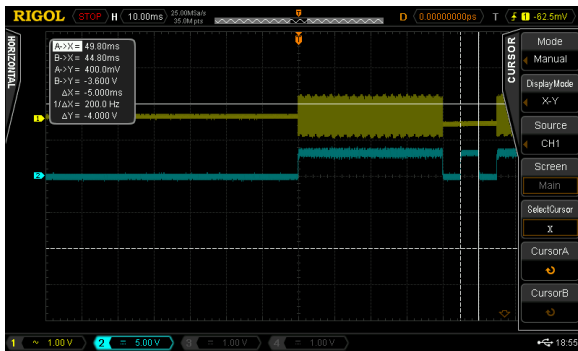
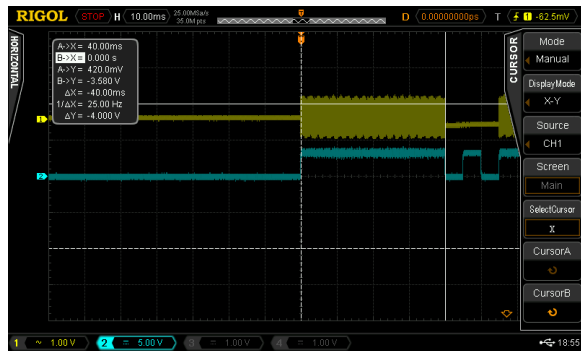


Figura 49. Señal de gradientes de izquierda a derecha gradiente Gz, Gy, Gz y Gx (señal azul).

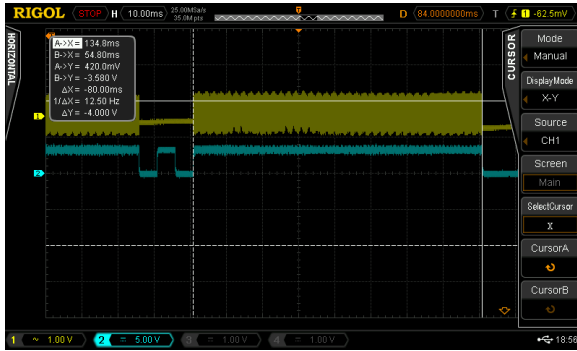
A continuación, se muestra una ampliación de la Figura 49 para clarificar cada gradiente.



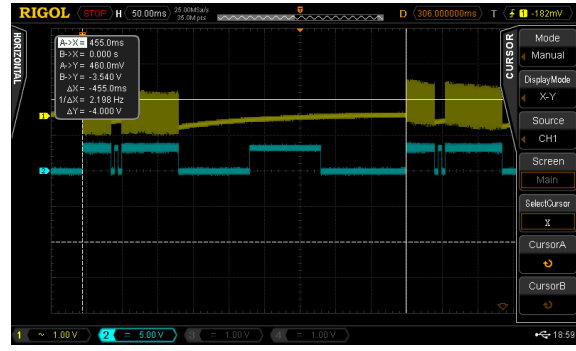
(a)



(b)



(c)



(d)

Figura 50. (a) Gradiente Gy, (b) y (c) Gradiente Gz, (d) Gradiente Gx.

Figura 50 (a) muestra el pulso correspondiente al gradiente Gy con un tiempo de 5ms; la Figura 50 (b) y (c) el pulso de gradiente Gz que se activa tiempo antes de los pulsos de RF de 90° y 180° para después desactivarse; con valores de 40 y 80 ms respectivamente. Y (d) de la Figura 50 muestra el gradiente Gx, del cual también es el tiempo de adquisición de datos. Cabe mencionar que estos valores de gradientes pueden ser modificados por el usuario ya que el sistema es reconfigurable.

Hasta aquí se han mostrado los resultados obtenidos para la parte correspondiente al sistema de transmisión, lo cuales han mostrado un buen desempeño de esta parte del sistema.

SISTEMA DE RECEPCION.

A continuación, se muestra el sistema de recepción también implementado en un FPGA.

En esta segunda etapa se necesita adquirir la señal de RF proveniente del tejido. Esta señal modulada necesita ser demodulada en el FPGA. Para ello se utilizan las herramientas de IDE VIVADO, las IP correspondientes y programas diseñados en VHDL.

Como se ha indicado, se utiliza la IP DDC que permite trasladar la señal de entrada en FI a banda base y con una frecuencia intermedia programada por el usuario. En esta IP la señal de entrada será submuestreada a una FI aproximadamente en 0Hz para después aplicar un filtro pasa bajas según la teoría del muestreo paso banda.

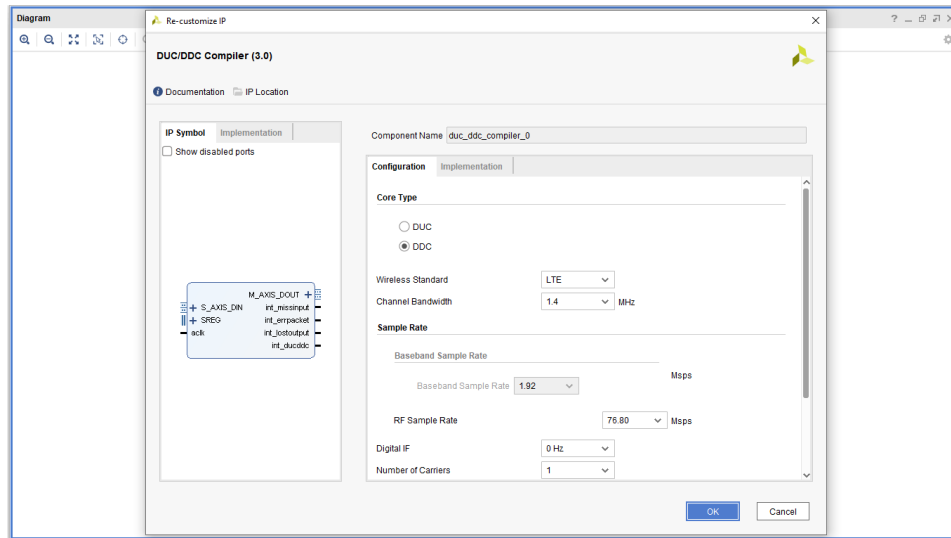


Figura 51. Interfaz de usuario del DDC.

En la Figura 51 se muestra la interfaz de usuario que permite cargar los parámetros con los que se va a procesar la señal modulada. Permite seleccionar los dos elementos fundamentales en su sistema SDR que son el DUC y el DDC. También tiene la función de seleccionar los tres diferentes estándares inalámbricos, tales como: LTE, TD-SCDMA y W-CDMA. Luego se tiene el ancho de banda del canal, la frecuencia IF entre otras funciones. Para mayor información véase el manual de usuario DDC de XILINX.

Hay que tomar en cuenta que el demodulador emplea la misma frecuencia de oscilación utilizada en la modulación, con el objetivo de recuperar la señal de RF de forma correcta.

El DDC permite multiplicar la señal modulada con un mezclador interno con el fin de obtener las señales en fase, I, y en cuadratura Q, ambas señales se multiplican por una exponencial compleja $e^{-j2\pi f' t}$ donde f' es la frecuencia central de la senoidal reconstruida. Asumir que $f' = f_0$. Después se multiplica por un factor de 2 para posteriormente ser filtradas y recuperar la señal de información. Esta técnica permite que el convertidor analógico – digital opere a la mitad de la velocidad [42].

El convertidor analógico digital AD6644 es una tarjeta de desarrollo que permite muestrear a 65MSPS en complemento a dos. Este puede operar con un reloj interno de 66MHz, así como también con reloj externo por medio de un BNC (J3) según la hoja de especificaciones. El convertidor tiene una resolución de 14 bits que están conectados a un header (J2) B13 – B00; donde B13 es el bit más significativo y B00 el bit menos significativo. La tarjeta cuenta con una alimentación independiente de 5V en CD para la señal analógica y 3.3V de CD para la señal digital.

El FPGA tiene que adquirir 14 bits que provienen del CAD desde el header (J2). Estos 14 pines van conectados a los puertos PMOD de entrada/salida. Los 14 bits deben ser adquiridos por el FPGA a una velocidad por lo menos el doble de 65MHZ, con el fin de permitir que se ejecute el código diseñado. También el aumento de velocidad del reloj sirve para sincronizar el tiempo del DATA READY con el FPGA, que es cada 15.4 ns. Por lo tanto, el FPGA es capaz de estar leyendo el puerto DATA READY y en paralelo ejecutando el resto del código. Para que el FPGA tenga el tiempo necesario para leer este puerto se maneja un reloj de 250MHz.

En la IP DDC no pueden adquirir los datos directamente del CAD desde los puertos PMOD, por lo que es necesario utilizar la IP FIFO GENERATOR que permite ir almacenando los datos para posteriormente el DDC los adquiera y procese como se ve en la Figura 52.

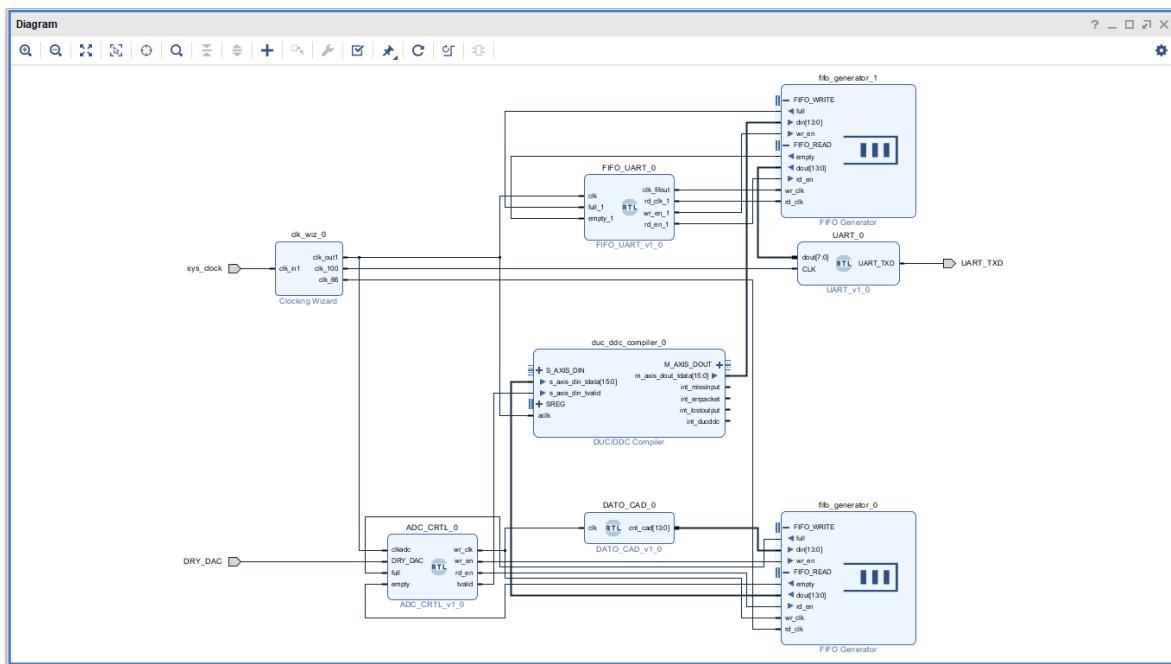


Figura 52. Sistema de recepción completo.

Una IP FIFO (`fifo_generator_0`) requiere escribir (`din`) en memoria el dato proveniente del CAD por medio de los PMOD JA y JD (`DATO_CAD_0`) a una velocidad de 65MHz e ir leyendo (`dout`) a la misma velocidad para sincronización y no perder datos. Hay que recordar que las IP FIFO utilizadas tienen la posibilidad de manejar relojes de escritura (`wr_en`) y lectura (`rd_en`) independientes. Este sistema es controlado por un programa en VHDL llamado `ADC_CRTL`, este es transformado en bloque RTL para realizar la conexión entre el

puerto DATA READY (DRY_DAC), el DDC (duc_ddc_compiler_0) y el sistema FIFO (fifo_generator_0) como se muestra en la Figura 52.

Ya almacenados los datos provenientes del CAD, la IP FIFO manda los datos hacia el DDC (M_AXIS_DIN_TDATA) para realizar el procesamiento. A la salida de datos del DDC (M_AXIS_DOUT_TDATA) a pesar que van en banda base hay que configurar la velocidad de transmisión hacia la PC por medio del UART (UART_0). Es por ello que se tiene otra IP FIFO (fifo_generator_1) que permita guardar los datos de salida del DDC y poder transmitir hacia la PC con una velocidad de 115200 bauds que es equivalente a 8.681us por cada bit enviado.

Otra gran ventaja de la FIFO es que permite seleccionar el tamaño del registro, para este caso en escritura serán de 16 bits y de lectura de 8 bits. Esto es porque el DDC envía en sus pines de salida (M_AXIS_DOUT_TDATA) palabras de 14 bits. Según la teoría, el UART solo permite la transmisión de palabras de 8 bits más un bit de inicio y de un bit stop. Por lo tanto, el bit de stop y el bit de inicio más los 8 bits suman palabras de 10 bits. Utilizando aritmética básica es necesario que la FIFO deba leer a una velocidad de 86.81us por cada palabra de datos. Por lo tanto, se estarán enviando palabras de 8 bits que al ser recibidos en la PC se concatenen para generar la palabra original de 14 bits y recuperar la señal original.

Esta última parte es manejada por medio de un programa en VHDL y transformado a bloque para la conexión con los pines de la FIFO, en la Figura 52 se muestra con el nombre de FIFO GENERATOR_1, FIFO_UART_0 y UART_0.

Como todo sistema se tienen desventajas, para este caso se están utilizando dos IP FIFO. Una IP FIFO para escritura y lectura para almacenamiento de la palabra proveniente del CAD y otra IP FIFO para la transmisión de la palabra de 14 bits de la salida del DDC hacia la PC a través del UART.

El problema con los relojes de escritura (wr) y lectura (rd) es que tienen que estar sincronizados con el DATA READY del CAD. El DATA READY no es un reloj como tal sino un estado de habilitación del CAD, que se establece en alto cuando hay una palabra por enviar. Por lo tanto, su frecuencia del DATA READY es variable y el reloj de la IP FIFO es estable, por esta razón, había pérdida de datos y no hay sincronización.

Para entender el funcionamiento de la FIFO en la Figura 53 se muestra una simulación de la IP donde sys_clock es el reloj interno de la tarjeta de desarrollo ARTY A7. wr_clk es el reloj de escritura que como se ha mencionado es estable y programable. rd_clk es el reloj de lectura que también puede ser programado y es estable. La IP FIFO también cuenta con habilitación de escritura y lectura; wr_en y rd_en respectivamente. Estos pines se mantienen en alto para habilitar, ya sea la escritura o la lectura; se mantienen en bajo para deshabilitar. Además, estas IP tienen banderas que permiten indicar cuando la FIFO está llena y vacía; full y empty respectivamente. Por cada periodo de reloj de wr_clk la FIFO almacena un dato (din[15:0]).

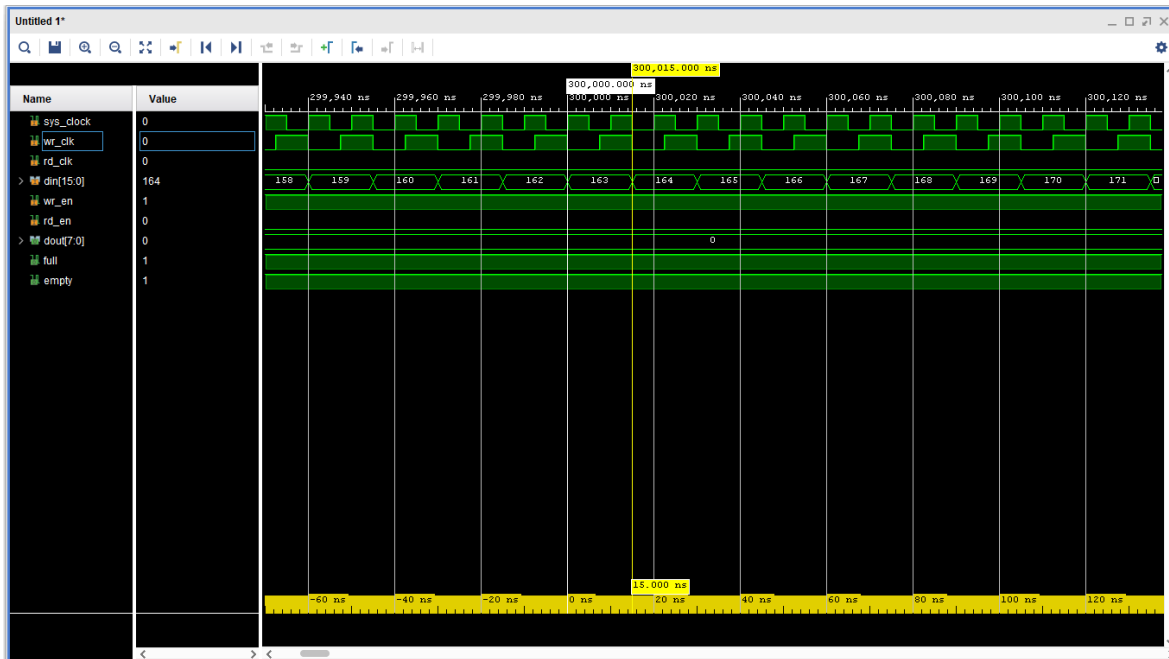


Figura 53. Simulación de la IP FIFO.

IP BLOCK MEMORY GENERATOR.

Para dejar a un lado el problema de sincronización se opta por cambiar el sistema de almacenamiento de datos por una IP BLOCK MEMORY GENERATOR, BMG como se muestra en la Figura 54. Este núcleo tiene dos puertos totalmente independientes que acceden a un espacio de memoria compartida. Los puertos A (BRAM_PORTA) y B (BRAM_PORTB) tienen una interfaz de lectura y escritura con un puntero de direccionamiento de registro. En las arquitecturas FPGA UltraScale™, Zynq®-7000 y serie 7, cada una de las cuatro interfaces; single port RAM, single dual port, true single port RAM RAM, etc. La IP se puede configurar con tamaño de palabras de 1 hasta 4608 bits y una profundidad de memoria de $2^{19} = 524288$ para ambos puertos [52].

La IP BMG no requiere de un reloj sincronizado para escritura y lectura. Puede operar cada puerto con un puerto de habilitación de escritura (wea) e ir direccionado cada uno de los registros de memoria (add).

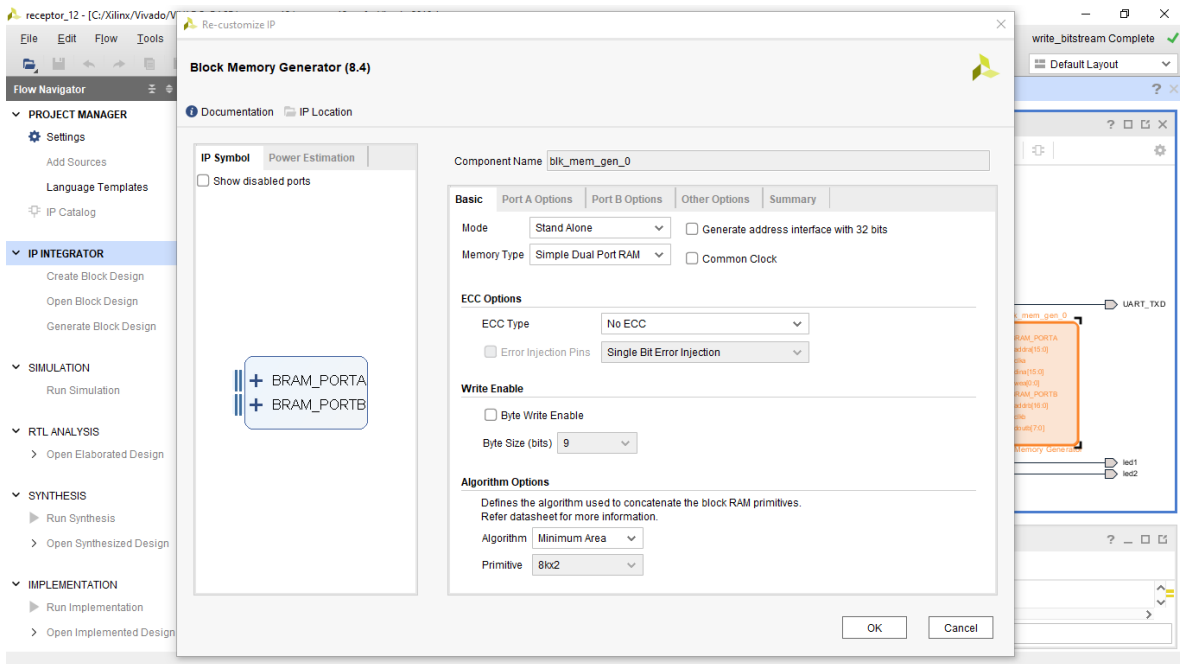


Figura 54. Interfaz de usuario de la IP BMG.

Para este proyecto se utilizó una anchura de puerto de 16 bits para la lectura de los datos (BRAM_PORTA) y una profundidad de direccionamiento de 262 144.

Hay que tomar en cuenta que se están almacenando 16 bits para el puerto de escritura, por lo tanto, la GUI reconfigura el puerto B de lectura a 8 bits. Además, la profundidad se está direccionando 262144 registros para el puerto A, en automático la profundidad del puerto B se duplica a 522288.

Por otro lado, en la pestaña del puerto A (Port A options) en la opción “operating mode” hay que cambiar a “write first” esto se debe a que se están manejando dos puertos diferentes. Entonces hay que dar la instrucción a la GUI quien de los dos puertos va ser de lectura y escritura. Realizando la operación para el puerto A; para el puerto B no es necesario realizar cambios ya que en automático la GUI lo actualiza. Otro aspecto importante es la opción “enable Port Type” este debe cambiarse por “Always Enable” ya que, sin este, no se podrá escribir y leer en el BMG. Para más información revisar [52].

De esta forma el programa tiene una instrucción que se encarga de estar revisando el pin DRY_DAC, que es la bandera que se establece en alto cada vez que hay una palabra en la salida del DAC. Por lo tanto, si DRY_DAC está en alto, addra se direcciona en el primer registro como se observa en la Figura 55. Así que la palabra de 16 bit entra por el pin dina y se almacena en el registro 1. Addra se incrementa en 1 para almacenar la siguiente palabra

en el siguiente registro. Y así consecutivamente con los demás paquetes de datos hasta alcanzar la profundidad de la memoria del puerto A que en este caso es de 262144.

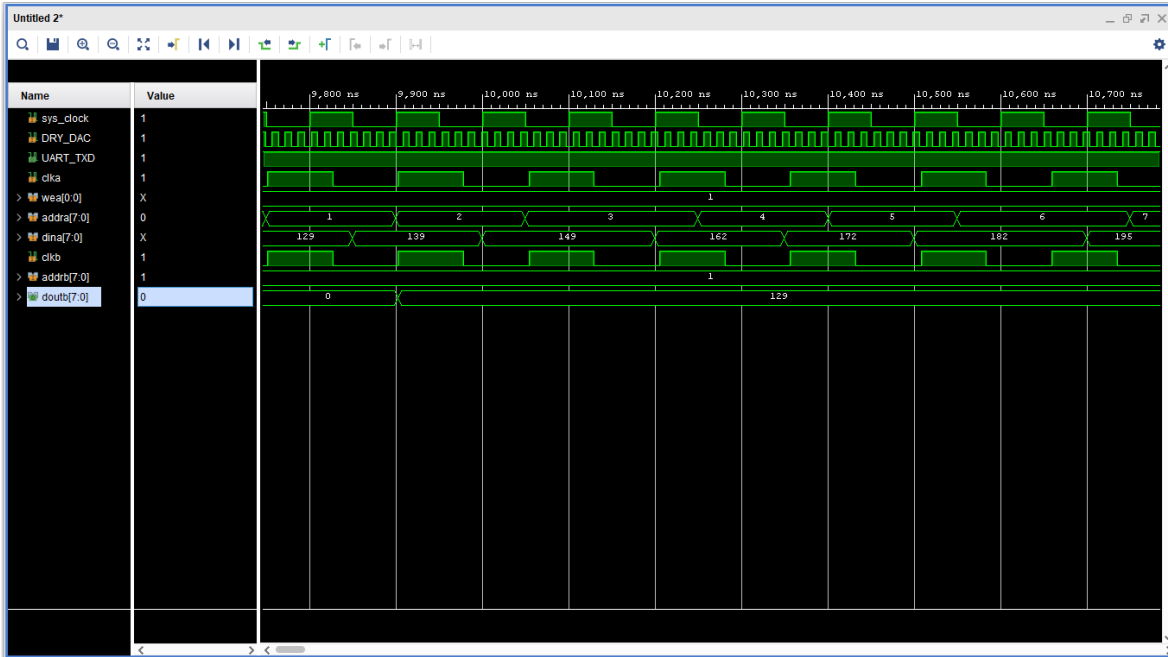


Figura 55. Diagrama de tiempo de IP BMG.

La memoria ha alcanzado la profundidad de almacenamiento programada; ahora hay que disgregarla a través del puerto B. Siguiendo el mismo procedimiento que el puerto A. Con solo direccionar el primer registro, la palabra sale por el pin dout [7:0] a una velocidad de 86.81us (UART_TXD) como se muestra en la Figura 56. Se puede ver que por cada incremento de dirección (addrb - color rojo) en paralelo se envía una palabra de 8 bits por el puerto Tx (doutb - color morado).

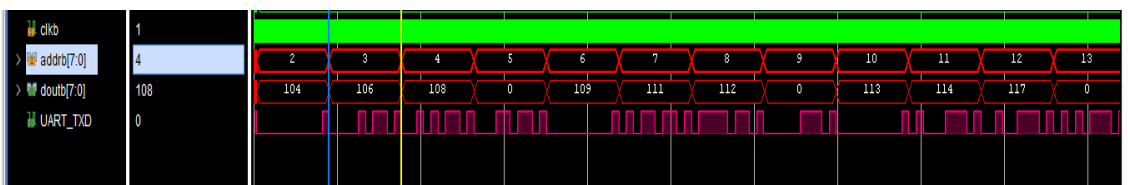
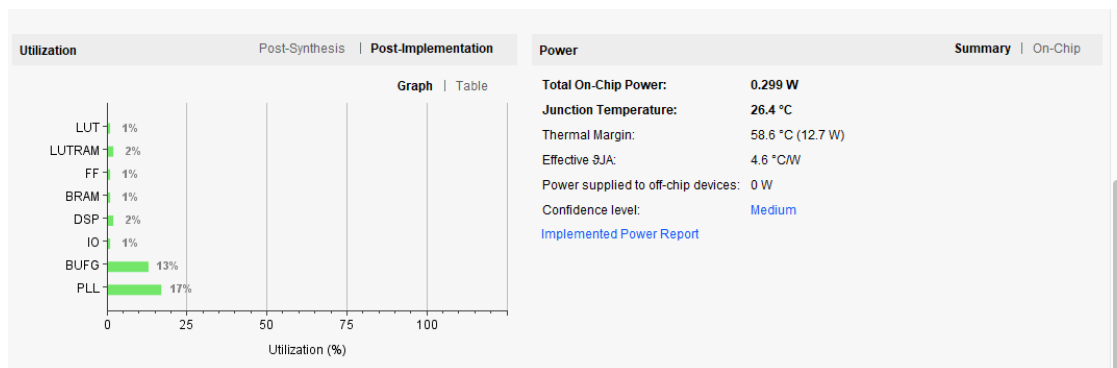


Figura 56. Diagrama de tiempo de UART.

Después del envío de datos por medio del UART, en MATLAB se implementa un algoritmo que permita adquirir las palabras enviadas desde la RAM, las concatene para formar palabras de 14 bits y se grafiquen para visualizar la señal muestreada por el CAD.

Recursos del hardware

A continuación, se muestran una gráfica de los recursos utilizados en el FPGA para la implementación completa del sistema.



Se puede observar que los recursos utilizados fueron mínimos, por lo que la aplicación de un FPGA en la realización de un SDR tiene en su mayoría buenos resultados.

5. Conclusiones.

Metas Logradas.

Al inicio del trabajo se utilizó una tarjeta de desarrollo ATLYS de XILINX de la familia spartan 6, en la cual se tuvo el problema de que su software de programación ISE DESIGN SUIT corría sobre una máquina virtual que presentaba dificultades de conexión usb de la tarjeta hacia la PC, así como también la programación de la tarjeta por medio del JTAG. Por este motivo se decidió por la búsqueda de una tarjeta que fuera compatible con Windows 10 y que su ambiente de programación fuera amigable para el usuario.

Por lo tanto, se utilizó una tarjeta NEXYS VIDEO de XILINX de la familia spartan 7. El ambiente de programación IDE VIVADO es amigable con el usuario, facilitó el diseño y permitió simulaciones. Por otro lado, dicha tarjeta se eligió por la variedad de puertos en su terminal “FMC Connector” con más de 100 pines, alguno de los cuales son pines dedicados tanto para reloj, entrada/salida, y pines de alta impedancia. Pero surgió el problema que, para poder habilitar estos pines, habrá que realizar un programa para intercambiar el voltaje de alimentación de los pines (PMOD, swithes, etc) hacia el FMC Connector. Por lo cual, por falta de experiencia en este campo, al tratar de realizar este cambio de voltaje, se quemó la tarjeta. La tarjeta no tiene un circuito de apoyo contra fallos de voltaje o de corto circuito, a pesar de su alto costo, además que el tiempo de envío del origen hacia el destino, es muy largo.

En consecuencia, se decide realizar la adquisición de otra tarjeta que no presente este tipo de problemas, que tenga puertos de entrada/salida, conexión UART y además que sea compatible con el software IDE VIVADO. Se adquiere la tarjeta ARTY A7 de XILINX de la familia spartan 7. Con esta tarjeta de desarrollo se logra implementar parte de la radio digital.

En primer lugar, se diseñó e implementó un transmisor digital como parte de un radio digital en este FPGA utilizando bloques IP. Se diseñó la programación particular en VHDL para poder conectar los pines de las diferentes IP's aplicadas en este proyecto. El transmisor funciona correctamente logrando formar pulsos de RF's configurables para generar las secuencias típicas de los sistemas de RMN. Además, se logró la generación de los pulsos correspondientes a los diferentes gradientes Gx, Gy y Gz, típicos de los sistemas de RMN.

Por parte de la conversión digital analógica cabe mencionar que hay una gran variedad de tarjetas de desarrollo con mayores características que el AD9755. En relación con esto, una ventaja de este proyecto es que se puede intercambiar el convertidor y con tan solo configurar algunos parámetros en la programación del FPGA se podrá tener otros resultados favorables.

Para mayor información visitar la página de ANALOG DIVICES. Por medio de esta tarjeta se logró realizar sobremuestreo a una tasa de 300MSPS aplicando un reloj desde un puerto del FPGA hacia el CDA, logrando uno de los objetivos planteados en este proyecto.

Por parte del receptor se logró implementar y diseñar los diferentes bloques que componen este sistema. Estos bloques son: Almacenamiento, UART y lectura de datos del CAD. Para el CAD se utilizó una tarjeta de desarrollo AD6644ST de la marca ANALOG DEVICES. Esta tarjeta se utilizó para aplicar la técnica de submuestreo, pero no se logró el objetivo a causa de la comunicación UART hacia la PC. Con la ayuda de un osciloscopio y un generador de funciones (aplicando una señal senoidal de baja frecuencia) se revisó el funcionamiento del CAD observando las señales de los pines, obteniendo buen resultado en la conversión.

Para la etapa del bloque BMG se logra almacenar la palabra de 14 bits del CAD en la memoria RAM para posteriormente ser enviados hacia la PC. Se logra el envío de palabras de 8 bits por la UART – Tx pero se muestran números saltados y sin lógica alguna a consecuencia de un error de comunicación entre la PC y el UART; por lo que se diseña un receptor Rx en el FPGA. Esto con el propósito de permitir el envío de palabras de 8 bits hacia la PC. Por ejemplo, se envía al FPGA, por medio del UART – Rx desde la PC, un carácter ‘A’ que indique que puede empezar a enviar las palabras almacenadas en la RAM hacia la PC, logrando de esta manera una mejor comunicación entre FPGA y PC. Por tiempo y constantes fallas en esta parte, no se logra el objetivo de Rx por lo que se queda para trabajo a futuro.

Trabajos por realizar.

Realizar la retroalimentación entre el transmisor y el receptor. De tal forma que se esté generado los pulsos de RF y en paralelo se esté demodulando la señal por medio del receptor. Aplicar la teoría del submuestreo en el bloque receptor digital utilizando un DDC y el CAD.

Reconstruir la señal original en la PC sin pérdida de información con apoyo de algún software, como MATLAB para mostrar las gráficas de frecuencia y espectro.

Apéndice A.

Códigos Matlab.

```
clear all
close all
clc
%borrar conexiones previas
delete(instrfind({'port'}, {'COM7'}));
%SE CREA UNA CONEXION EN SERIE
s = serial('COM7');
set(s, 'BaudRate', 115200)
%SE ABRE PUERTO
fopen(s)
%SE MANDA VALOR
fprintf(s, '*IDN?')
%idn = fscanf(s)
mem = 131072;
%n = (mem*2)+1;
n = (mem*2)+1;
val = zeros(1,n);
for i = 1:n
    %val(1,i) = fread(s,1,'uchar');
    val(1,i) = fread(s,1,'uchar');
end
%cerrar puerto
fclose(s)
delete(s)
clear s
val = val';
%dlmwrite('archivo.dat',val,'delimiter','\n','precision','%.2f')
a = val;
t = length(a);
a = a(1:t);           %se accede a un rango detrminado; en este caso
                    %solo se accede del segundo valor hasta t
t = length(a);       %asigna el tamaño del vector (lenght())
%
a1 = dec2hex(a);     %convierte de decimal a hexadecimal
g = 2:2:t;           %construye un vector desde 1, en incrementos 2 en 2,
                    hasta el valor de t+2 = 22
h = 1:2:t-1;         %%construye un vector desde 2, en incrementos 2 en 2,
                    hasta el valor de t+1 = 21

for k = 1:t/2
    v = h(1,k);
    w = g(1,k);
    a2(k,:) = strcat(a1(v,:),a1(w,:));
end
b = hex2dec(a2);
b2 = b;
t2 = length(b);
```

Apéndice B.

Códigos VHDL.

A continuación, se muestran los códigos diseñados en VHDL para el manejo de los diferentes bloques IP.

PROGRAMA PARA ALMACENAR DATO Y ENVIO POR UART

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.all;
entity UART is
    Port ( CLK      :in STD_LOGIC;
          UART_TXD :out STD_LOGIC;
          doutb     :in STD_LOGIC_VECTOR (7 downto 0);
          addrb     :out std_logic_vector(16 downto 0);
          addra     :out std_logic_vector(15 downto 0);
          dina      :out std_logic_vector(15 downto 0);
          wea       :out std_logic:='0';
          DRY_DAC   :in std_logic:='0';
          datain    :in std_logic_vector(15 downto 0);
          led1      :out std_logic:='0';
          led2      :out std_logic:='0';
          reset     :in std_logic
    );
end UART;
```

architecture Behavioral of UART is

component UART_TX_CTRL

Port(

 SEND : in std_logic;

 DATA : in std_logic_vector(7 downto 0);

 CLK : in std_logic;

 READY : out std_logic;

 UART_TX : out std_logic

);

end component;

--UART_TX_CTRL control signals

signal uartRdy : std_logic;

signal uartSend : std_logic := '0';

signal uartData : std_logic_vector (7 downto 0):= "00000000";

signal uartTX : std_logic;

--*****SEÑALES PARA RAM*****

signal addrb_s:std_logic_vector(16 downto 0):="0000000000000000";

signal addra_s:std_logic_vector(15 downto 0):="0000000000000000";

signal wea_s :std_logic;

signal bandera:std_logic;

--*****MAQUINA DE ESTADOS*****

--*****PARA LEER RAM*****

signal ed :std_logic_vector(2 downto 0):="000";

constant sd0: std_logic_vector(2 downto 0):="000";

constant sd1: std_logic_vector(2 downto 0):="001";

constant sd2: std_logic_vector(2 downto 0):="010";

```

constant sd3: std_logic_vector(2 downto 0):="011";
constant sd4: std_logic_vector(2 downto 0):="100";
constant sd5: std_logic_vector(2 downto 0):="101";
constant sd6: std_logic_vector(2 downto 0):="110";
constant sd7: std_logic_vector(2 downto 0):="111";

```

```

--*****PARA ESCRIBIR EN RAM*****

```

```

signal edo : std_logic_vector (2 downto 0):="000";
constant s0: std_logic_vector (2 downto 0):="000";
constant s1: std_logic_vector (2 downto 0):="001";
constant s2: std_logic_vector (2 downto 0):="010";
constant s3: std_logic_vector (2 downto 0):="011";
constant s4: std_logic_vector (2 downto 0):="100";
constant s5: std_logic_vector (2 downto 0):="101";
constant s6: std_logic_vector (2 downto 0):="110";
constant s7: std_logic_vector (2 downto 0):="111";

```

```

begin

```

```

--*****LEER RAM*****

```

```

char_load_process: process (clk,doutb, bandera,ed,reset,addrb_s)

```

```

begin

```

```

if (reset = '1') then

```

```

    ed    <= s0;

```

```

    addrb_s <= "0000000000000000";

```

```

    led1  <='0';

```

```

    led2  <='0';

```

```

elseif (rising_edge(clk)) then

```

```

    case ed is

```

```
when sd0 =>
    if addrb_s = "11011000110011000" then
        ed <= sd2;
    else
        addrb_s <= addrb_s + 1;
        ed <= sd1;
    end if;
```

```
when sd1 =>
    if (bandera = '1')then
        led1 <= '1';
        if (uartRdy = '1') then
            uartSend <= '1';
            uartData <= doutb;
            ed <= sd0;
        else
            uartSend <= '0';
            ed <= sd1;
        end if;
    end if;
```

```
when sd2 =>
    addrb_s <= "0000000000000000";
    uartSend <= '0';
    led2 <= '1';
```

```

    ed <= sd0;

    when others =>
        ed <= sd2;
    end case;
end if;
end process;

*****ESCRIBIR EN RAM*****

guarda_dato:process (clk,DRY_DAC,addra_s,bandera,edo,datain)
begin
if (reset = '1') then
    edo <= s0;
    addra_s <= "0000000000000000";
    dina <= "0000000000000000";
    wea_s <= '0';
    bandera <= '0';
elsif (rising_edge(clk)) then
    case edo is
        when s0 =>
            if (DRY_DAC = '1') then
                wea_s <= '1';
                dina <= datain;
                edo <= s1;
            elsif(DRY_DAC = '0') then
                --wea <= '0';
                edo <= s0;
            end if;

```

```

when s1 =>

    --wea_s <= '0';

    if addra_s = "1101100011001100" then    --55500 datos

        addra_s <= "0000000000000000";

        bandera <= '1';

        edo    <= s2;

    else

        addra_s <= addra_s + 1;

        edo <= s0;

    end if;

when s2 =>

    edo <= s2;

    --bandera <= '0';

when others =>

    edo <= s0;

end case;

end if;

end process;

Inst_UART_TX_CTRL: UART_TX_CTRL port map(

    SEND    => uartSend,

    DATA   => uartData,    --aqui se envia el dato de 8 bits para el por pto serial

    CLK     => CLK,

    READY   => uartRdy,

    UART_TX => uartTX

);

```



```

UART_TXD <= uartTX;

addrb <= addrb_s;

addra <= addra_s;

wea <= wea_s;

end Behavioral;

```

Control de baudaje.

```

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.std_logic_unsigned.all;

```

```

entity UART_TX_CTRL is

```

```

    Port ( SEND    : in  STD_LOGIC;

          DATA   : in  STD_LOGIC_VECTOR (7 downto 0);

          CLK     : in  STD_LOGIC;

          READY   : out STD_LOGIC;

          UART_TX : out STD_LOGIC);

```

```

end UART_TX_CTRL;

```

```

architecture Behavioral of UART_TX_CTRL is

```

```

type TX_STATE_TYPE is (RDY, LOAD_BIT, SEND_BIT);

```

```

constant BIT_TMR_MAX : std_logic_vector(11 downto 0) := "100001111010"; --2170 = (round(250MHz /
115200)) - 1

```

```

constant BIT_INDEX_MAX : natural := 10;

```

```

signal bitTmr : std_logic_vector(11 downto 0) := (others => '0');

```

```

signal bitDone : std_logic;

```

```

signal bitIndex : natural;

signal txBit : std_logic := '1';

signal txData : std_logic_vector(9 downto 0); --tamaño de envio de datos 1 + dato + 0

signal txState : TX_STATE_TYPE := RDY;

begin

next_txState_process : process (CLK)

begin

    if (rising_edge(CLK)) then

        case txState is

            when RDY =>

                if (SEND = '1') then

                    txState <= LOAD_BIT;

                end if;

            when LOAD_BIT =>

                txState <= SEND_BIT;

            when SEND_BIT =>

                if (bitDone = '1') then

                    if (bitIndex = BIT_INDEX_MAX) then

                        txState <= RDY;

                    else

                        txState <= LOAD_BIT;

                    end if;

                end if;

            when others=> --should never be reached

                txState <= RDY;

            end case;

        end if;

    end process;

```

```

bit_timing_process : process (CLK)
begin
    if (rising_edge(CLK)) then
        if (txState = RDY) then
            bitTmr <= (others => '0');
        else
            if (bitDone = '1') then
                bitTmr <= (others => '0');
            else
                bitTmr <= bitTmr + 1;
            end if;
        end if;
    end if;
end process;

```

```

bitDone <= '1' when (bitTmr = BIT_TMR_MAX) else
    '0';

```

```

bit_counting_process : process (CLK)
begin
    if (rising_edge(CLK)) then
        if (txState = RDY) then
            bitIndex <= 0;
        elsif (txState = LOAD_BIT) then
            bitIndex <= bitIndex + 1;
        end if;
    end if;
end process;

```

```
end process;
```

```
tx_data_latch_process : process (CLK)
```

```
begin
```

```
    if (rising_edge(CLK)) then
```

```
        if (SEND = '1') then
```

```
            txData <= '1' & DATA & '0';
```

```
        end if;
```

```
    end if;
```

```
end process;
```

```
tx_bit_process : process (CLK)
```

```
begin
```

```
    if (rising_edge(CLK)) then
```

```
        if (txState = RDY) then
```

```
            txBit <= '1';
```

```
        elsif (txState = LOAD_BIT) then
```

```
            txBit <= txData(bitIndex);
```

```
        end if;
```

```
    end if;
```

```
end process;
```

```
UART_TX <= txBit;
```

```
READY <= '1' when (txState = RDY) else
```

```
    '0';
```

```
end Behavioral;
```

Generador de pulsos

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.STD_LOGIC_ARITH.ALL;

use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity reset_2 is
    port( clk :in std_logic;
          pul_90:out std_logic;
          reset :in std_logic
        );
end entity reset_2;

architecture Behavioral of reset_2 is
    signal contadort : std_logic_vector(22 downto 0):="0000000000000000000000";
    signal pulso: std_logic:= '0';
    signal e :std_logic_vector(2 downto 0):="000";
    constant s0: std_logic_vector(2 downto 0):= "000";
    constant s1: std_logic_vector(2 downto 0):= "001";
begin
    pb:process(clk, reset, contadort)
    begin
        if(reset = '1') then
            contadort <= "0000000000000000000000";
        elsif(clk' event and clk = '1')then
            case e is
                when s0 =>
```

```

pulso <= '0';

if (contadort = "11110100001001000000000") then --80ms
    contadort <= "0000000000000000000000";

    e <= s1;

else

    contadort <= contadort +1;

    e <= s0;

end if;

when s1 =>

    pulso <= '1';

    if (contadort = "00011110100001001000000") then

        contadort <= "0000000000000000000000";

        e <= s0;

    else

        contadort <= contadort +1;

        e <= s1;

    end if;

when others =>

    e <= s0;

end case;

end if;

end process;

pul_90 <= pulso;

end Behavioral;

```

Bibliografía.

- [1] Bloch, F., "Nuclear Induction", *Phys. Rev.*, Vol. 70, No. 7 y 8, pp. 460-474, 1946.
- [2] Purcell, E.M., "Resonance Absorption by Nuclear Magnetic moments in a solid", *Phys. Rev.*, Vol. 69, pp. 37-38, 1946.
- [3] Proctor, G. and Yu, F. C., "The dependence of nuclear magnetic resonance frequency upon chemical compound", *Phys. Rev.*, Vol. 77, pp. 717, 1950.
- [4] Gabillard, R., "Mesure de temps de relaxation T2 en presence d'une inhomogeneite de champs magnetique superieur it la largeur de raie", *CR Acad. Sci. Paris*, 232, 1951.
- [5] Damadian, R., "Tumor detection by nuclear magnetic resonance", *Science*, Vol. 171, No. 3976, pp. 1151-1153, 1971.
- [6] Damadian, R., "Apparatus and method for detecting cancer in tissue", US Patent No 3789823, filed 17 March 1972.
- [7] Lauterburg, P. C., "Image formation by induced local interactions: examples of employing nuclear magnetic resonance", *Nature*, Vol. 242, pp.190-191, 1973.
- [8] Garroway, A. N., Grannell, P. K., and Mansfield.P., "Image formation in NMR by a selective irradiative process", *J. Phys. C*, Vol. 7, pp.457-462, 1974.
- [9] Kumar, A., Welti, D., and Ernst, R., "NMR Fourier Zeugmatography," *J. Magn. Res.*, Vol. 18, No. 1, pp.69-83, 1975.
- [10] Nikolay V. Anisimov. "Simultaneous Recording of NMR Signals from Nuclei with Different Gyromagnetic Ratios Using Undersampling Technique", springer-Verlag GmbH., Austria, 2018
- [11] Philips medical systems, "Principios de la formación de imágenes", Países bajos, 1985.
- [12] Tang, W., Sun, H., "A digital receiver module with direct data acquisition for magnetic resonance imaging systems", *Rev. Sci. Instrum.*, Vol. 83, 104701-7, 2012.
- [13] Tang, W., and Wang, W., "A single-board NMR spectrometer based on a software defined radio architecture", *Meas. Sci. Technol.* Vol. 22, 015902 (8p), 2011.
- [14] Pérez, P., "Adquisición de imágenes de resonancia magnética nuclear mediante técnicas de submuestreo", Tesis doctoral, Universidad Politécnica de Madrid, 1999.
- [15] Analog Devices, "AD6644: 14-Bit, 40 MSPS/65 MSPS Analog-to-Digital Converter", 2007.

- [16] Liang, X, “A digital magnetic resonance imaging spectrometer using digital signal processor and field programmable gate array”, *Rev. Sci. Instrum.*, Vol. 84, 054702-1 054702-8, 2013.
- [17] Hüseyin A., “Cognitive Radio, Software Defined Radio, and Adaptive Wireless System”. Springer. Agosto 2017
- [18] Rodger H. H., “Software Defined Radio Handbook”, PENTEK, April 2013.
- [19] Giro F.V. U., “Implementación de radio receptor definido por software (SDR) para la banda de (30kHz – 40 MHz)”, convención internacional de ciencias técnicas, Universidad de oriente, cuba, 2014.
- [20] Wayne, T., “Sistemas de Comunicaciones Electrónicas”, Prentice Hall, 2003.
- [21] Gallardo. V. S, “Elementos de sistemas de telecomunicaciones”, Paraninfo. 2da edición. 2019
- [22] Borth D. E., 2018, Telecommunication, <https://www.britannica.com/technology/telecommunication>.
- [23] LogiCORE IP DDS Compiler v6.0. Xilinx, December 2017. PG141.
- [24] San Roman, J. A., “Conocimientos básicos necesarios para realizar resonancia magnética en cardiología”, *Rev Esp Cardiol Supl*, 6, PP. 7E – 14E, 2006.
- [25] Tam H., Neilson K., “Desing Efficient Wireless Digital Up and Down Converters Leveraging CORE Generator and System Generator”, Xilinx, October 2007, XAPP1018.
- [26] Benitez, E., CONACYT, www.conacytprensa.mx/index.php/ciencia/salud/. 2017.
- [27] Rick P. “Magnetic Resonance in Medicine.” The Basic Texbook of the European Magnetic Resonance Forum., July 2013. www.magnetic-resonance.org.
- [28] S. S. Rathod, “Implementation of MRI Gradient Generation System and Controller on Field Programmable Gate Array (FPGA)”, International Conference on Communication 2018 IEEE.

- [29] Maxinez, G. D., “VHDL: El arte de programar sistemas digitales”, CECSA, 2002.
- [30] National Instruments, <http://www.ni.com/fpga/esa/> [Fecha de consulta: Enero 2019].
- [31] Fundamentos de FPGA. National Instrument. 5 de Marzo de 2019.
<http://www.ni.com/es-mx/innovations/white-papers/08/fpga-fundamentals.html>. [Fecha de consulta: octubre 2019].
- [32] Introducción a la Tecnología FPGA: Los cinco Beneficios Principales. National Instruments, Diciembre 2011.
- [33] Analog Devices, “Mixed Signal and DSP Desing Techniques”, Newnes/Elseiver, 2002.
- [34] Huircán, J. I., “Conversores Analógico-Digital y Digital-Analógico: conceptos basicos”.2015. <https://www.semanticscholar.org/paper/Conversores-An%C3%A1logo-Digital-y-Digital-An%C3%A1logo%3A-Huircan/4444cdaf237a0324faeaf3ef5a971281d0530e91>. Revisado Marzo 2019
- [35] Analog Devices, “AD6644, 14-Bit, 65 MSPS Analog-to-Digital Converter”, 2007.
- [36] Analog Devices, “AD9755, 14-Bit, 300 MSPS High Speed Digital-to-Analog Converter”, 2003
- [37] Proakis J. G., Manolakis D. G., “Tratamiento Digital de Señales”. PEARSON EDUCATION. Madrid. 2007.
- [38] Ambardar, A., “Procesamiento de señales analógicas y digitales”. Thomson. Segunda edición. 2002.
- [39] Lara Mendoza M. P. “Desarrollo de un sistema de recepción transmisión digital de propósito general en un FPGA”. Agosto 2015
- [40] Flayiu, LL, Nicolae, C, “Improving analog-digital converter’s resolution using the oversampling technique”. Proceeding of SPAME 2011.
- [41] <http://www.sound-pixel.com/blog/%C2%BFel-sobremuestreo-mejora-la-calidad-de-audio>. Revisado Noviembre 2019.
- [42] Brigham, O. “The Fast Fourier Transform and Its Applications”. Prentice Hall. 1988.

- [43] Matsumara N, “A new RF quadrature undersampling technique for an ideal software defined radio system”. Proceedings of Asis-Pacific Microwave Conference 2010.
- [44] Murphy E, and Slattery C. “All About Direct Digital Synthesis”. Analog Dialogue 38-08, August 2004.
- [45] LogiCORE IP Clocking Wizard 3.6 (ISE) / 4.2 (Vivado), July 2012. PG065.
- [46] LogiCORE IP Adder/Subtractor V12.0 (VIVADO), November 2015. PG120.
- [47] LogiCORE DUC/DDC V3.0 (VIVADO), November 2015. PG147
- [48] Skoog D, Holler F, J., “Principios de análisis instrumental”. CENCAGE learning. Sexta edición.
- [49] Bandhawarkar Dr. G, Aggarwal I. ”Synthesis and Implementation of UART using VHDL Codes”. Internatinoal Symposium on Computer, Consumer and Control, IEEE, 2012.
- [50] Nanda U, and Kumar S. “Universal Asynchronous Receiver and Transmitter (UART)” 3rd International Conference Advanced Computing and Communication Systems (ICACCS -2016). IEEE. India
- [51] LogiCORE FIFO generator v13.2 (VIVADO), October 4 2017. PG057
- [52] Pong P. Chu. “FPGA prototyping by VHDL examples”. John Wiley & sons, 2008, 3ra version.
- [53] LogiCORE Block Memory GeneratorV8.3 (VIVADO), Abril 5 2017. PG058