

UCUENCA

Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería en Telecomunicaciones

**Implementación de un sistema de transmisión multiportadora sobre una
plataforma FPGA**

Trabajo de titulación previo a la
obtención del título de Ingeniero
en Telecomunicaciones

Autores:

Henry Mauricio Castro Mendoza

Jhonatan Paúl Cambisaca Sánchez

Director:

Kenneth Samuel Palacio Baus

ORCID:  0000-0002-7318-8062

Co-Director:

Lizandro Damián Solano Quinde

ORCID:  0000-0001-7427-4889

Cuenca, Ecuador

2024-09-13

Resumen

Los sistemas de modulación multiportadora aumentan la tasa de bits en comunicaciones digitales y optimizan el uso del ancho de banda, en comparación con los sistemas de modulación de portadora única. En este trabajo, se implementan los sistemas OFDM y GFDM en una FPGA STEMLab 125-14 de Red Pitaya. La implementación de un sistema de transmisión multiportadora en una FPGA implica trasladar conceptos matemáticos complejos a nivel de hardware. Por tanto, es esencial dominar el entorno de descripción de hardware para abordar problemas de síntesis y retardos. Esto para formar un sistema de transmisión autónomo, sin necesidad de hardware externo. La implementación de OFDM y GFDM se basa en la representación vectorial de los sistemas multiportadora, cuyas características dependen del filtro formador de pulsos. En OFDM se emplea un pulso rectangular, mientras que en GFDM se usa un filtro de coseno elevado. Ambos sistemas utilizan símbolos 4-QAM y tramas de sincronización para facilitar la recepción. Para evaluar el sistema, se utiliza el dispositivo HackRF One junto con el software GNU Radio. La señal recibida se compara con la teórica mediante una correlación cruzada y se mide la tasa de error de bits (BER) en función de la distancia. Los resultados demuestran que la señal recibida coincide con la simulada y se observa que GFDM requiere un 7% más de SNR que OFDM para lograr una tasa de error similar. Estos resultados confirman la correcta implementación y establecen las bases para el desarrollo de nuevas tecnologías en sistemas de comunicación multiportadora.

Palabras clave del autor: programación paralela, sistema de transmisión, sistemas multiportadoras



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

Repositorio Institucional: <https://dspace.ucuenca.edu.ec/>

Abstract

Multicarrier modulation systems increase the bit rate in digital communications and optimize bandwidth usage compared to single carrier modulation systems. In this work, we implement OFDM and GFDM systems on a Red Pitaya FPGA STEMLab 125-14 platform. Implementing a multicarrier transmission system on an FPGA involves translating complex mathematical concepts into hardware-level operations. Mastery of the hardware description environment is crucial for addressing synthesis and delay issues, enabling the creation of a standalone transmission system without external hardware. The implementation of OFDM and GFDM is based on the vector representation of multicarrier systems, where the characteristics depend on the pulse-shaping filter used. OFDM employs a rectangular pulse, while GFDM uses a raised cosine filter. Both systems utilize 4-QAM symbols and synchronization frames to facilitate signal reception. To evaluate the system, we use the HackRF One device in conjunction with GNU Radio software. The received signal is compared with the theoretical signal using cross-correlation, and the bit error rate (BER) is measured as a function of distance. The results show that the received signal closely matches the simulated signal and it was observed that GFDM requires 7% more SNR than OFDM to achieve a similar error rate. These findings validate the implementation and lay the groundwork for developing new technologies in multicarrier communication systems.

Author keywords: parallel programming, transmission system, multicarrier systems



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

Institutional Repository: <https://dspace.ucuenca.edu.ec/>

Índice de contenido

1. Generalidades	14
1.1. Introducción	14
1.2. Objetivos	15
1.2.1. Objetivo general	15
1.2.2. Objetivos específicos	15
1.3. Alcance	15
1.4. Justificación	16
2. Marco Teórico	18
2.1. Sistemas de Modulación Multiportadora	18
2.1.1. Orthogonal Frequency Division Multiplexing: OFDM	18
2.1.1.1. Beneficios de la modulación OFDM en comunicaciones inalámbricas	18
2.1.1.2. Modelo en Bloques del Sistema de Transmisión OFDM	19
2.1.1.3. Problemas Asociados con OFDM	21
2.1.2. Técnicas avanzadas de acceso múltiple para 5G	22
2.1.2.1. Generalized Frequency Division Multiplexing: GFDM	22
2.2. Sistemas de Radio definidos por Software: SDR	26
2.2.1. HackRF One	27
2.2.2. GNU Radio	29
2.3. Field Programmable Gate Array	29
2.3.1. Características técnicas de los dispositivos FPGA	30
2.3.2. Spartan 3E	30
2.3.3. Basys 3	31
2.3.4. STEMLab 125-14	32
2.4. Método Analytic Hierarchy Process	33
2.4.1. Metodología del Análisis Jerárquico AHP	33
2.4.2. Estructuración de los niveles de jerarquía	34
2.4.3. Escala fundamental de Saaty	34
2.4.4. Matrices de comparación y cálculo de prioridades	36
2.4.5. Consistencia del Análisis	36
2.4.6. Síntesis de Prioridades	38
2.5. Trabajos Relacionados	38

3. Desarrollo del Sistema	41
3.1. Análisis de los dispositivos FPGA disponibles en la Universidad de Cuenca . . .	41
3.1.1. Definición el Problema	41
3.1.2. Diseño de la estructura jerárquica con AHP para la selección de la tarjeta FPGA	41
3.1.3. Criterios de evaluación para el análisis del Proceso de Jerarquía Analítica (AHP)	42
3.1.4. Comparación de los criterios	43
3.1.5. Comparación por pares para cada criterio	46
3.1.5.1. Hardware RF	46
3.1.5.2. Reloj Interno	46
3.1.5.3. Conversor digital analógico (DAC)	47
3.1.5.4. Laboratorio	47
3.1.5.5. Look-Up Tables (LUT)	47
3.1.5.6. Flip Flop	48
3.1.5.7. Entornos de desarrollo (IDE)	48
3.1.6. Matriz de prioridades de criterios y opciones	48
3.1.7. Interpretación de los Resultados	49
3.2. Esquema del sistema de comunicación	50
3.3. Implementación del Sistema de Transmisión sobre la FPGA	51
3.3.1. Codewords	54
3.3.2. Generación de muestras multiportadora (OFDM/GFDM)	56
3.3.2.1. Modelo Matemático de Sistemas Multiportadoras	56
3.3.2.2. Diseño del producto matriz-vector en VHDL	57
3.3.2.3. Implementación de OFDM en VHDL	62
3.3.2.4. Implementación de GFDM en VHDL	64
3.3.3. Interpolador	65
3.3.3.1. Descripción Matemática	65
3.3.3.2. Diseño en VHDL	66
3.3.4. Mezclador de Cuadratura	70
3.3.5. Cuantización uniforme	72
3.3.5.1. Descripción Matemática	72
3.3.5.2. Diseño en VHDL	73
3.4. Desarrollo de Receptor	76

3.4.1. Análisis matemático para la implementación del receptor en GNU Radio .	77
3.4.1.1. Desfases entre los relojes del sistema transmisor y receptor . .	77
3.4.1.2. Detección de inicio de tramas multiportadora	78
3.4.1.3. Demodulador Multiportadora	80
3.4.2. Software GNU Radio	81
3.4.2.1. Bloque Osmocom source para la recepción de la señal	81
3.4.2.2. Bloque DC Blocker	82
3.4.2.3. Bloque PLL Carrier Tracking corrector de fase	83
3.4.2.4. Bloque de filtrado Interpolating FIR Filter	84
3.4.2.5. Bloque Embedded Python Block para la adición de código en Python	85
3.4.2.6. Bloques de conversión	85
3.4.2.7. Bloques de visualización de resultados	85
3.4.2.8. Bloques de procesamiento de resultados	85
3.4.3. Implementación del sistema de recepción multiportadora en GNU Radio .	86
3.4.3.1. Receptor RF	86
3.4.3.2. Demodulador Multiportadora	87
4. Evaluación del Sistema	90
4.1. Metodología para realizar las pruebas y mediciones	90
4.1.1. Metodología para realizar el experimento 1	91
4.1.2. Metodología para realizar el experimento 2	92
4.2. Resultados de la evaluación experimental	92
4.2.1. Evaluación del experimento 1	92
4.2.2. Evaluación del experimento 2	96
5. Conclusiones, Desafíos, Recomendaciones y Trabajos Futuros	99
5.1. Conclusiones	99
5.2. Desafíos en la Implementación	100
5.3. Recomendaciones	101
5.4. Trabajos Futuros	102
Referencias	103
Anexos	111
Anexo A. Desarrollo de Pruebas experimentales	111

Anexo B. Implementación del sistema de transmisión Multiportadora	111
---	-----

Índice de figuras

2.1. Principio fundamental de la modulación OFDM. Fuente: [1].	20
2.2. Diagrama de bloques de un sistema transmisor típico para una modulación OFDM. Fuente: [2].	21
2.3. Acceso múltiple para la tecnología del futuro. Fuente: [3] © 2016 IEEE	23
2.4. Partición del tiempo y la frecuencia GFDM. Fuente: [4] © 2014 IEEE	23
2.5. Sistemas de comunicaciones GFDM. Fuente: [4] © 2014 IEEE	24
2.6. Modulador GFDM. Fuente: [4] © 2014 IEEE	24
2.7. Diagrama de bloques del receptor SDR. Fuente: [5].	27
2.8. Dispositivo HackRF One. Fuente: [6].	28
2.9. Diagrama de bloques del lado receptor del HackRF One. Fuente: [7].	28
2.10. Bloque Osmocom de GNU-RADIO. Fuente: [8]	29
2.11. Dispositivo FPGA Spartan 3E. Fuente: [9]	31
2.12. Dispositivo FPGA Basys 3. Fuente: [10]	32
2.13. Dispositivo FPGA STEMLab 125-14. Fuente: [11]	32
2.14. Estructura jerárquica del método AHP. Fuente: [12]	35
3.1. Esquema Jerárquico para la selección de la tarjeta FPGA.	42
3.2. Resultado del análisis AHP.	50
3.3. Visión global del sistema de comunicaciones implementado	51
3.4. IP Core Floating Point	52
3.5. Diagrama de bloques del sistema transmisor implementado sobre la FPGA STEM- lab 125-14	53
3.6. Diagrama de bloques del sistema generador de codewords.	55
3.7. Comportamiento de la interacción entre los valores flotantes pertenecientes a las muestras del filtro formador de pulsos y los símbolos 4-QAM	59
3.8. Diagrama de bloques para la obtención de una muestra de la onda multiportadora 60	
3.9. Diagrama de bloques simplificado de la implementación del producto matriz- vector	61
3.10. Respuesta al Impulso del filtro pasa bajo en el dominio de la frecuencia	67
3.11. Diagrama de Bloques del componente Interpolador	69
3.12. Diagrama de Bloques del componente Mezclador de Cuadratura	71
3.13. Diagrama de Bloques del componente cuantizador.	75

3.14. Diagrama esquemático del bloque Generador de Señales del ejemplo Frequency Counter. Fuente: [13]	75
3.15. Circuitería RF en STEMLab 125-14. Fuente: [14]	76
3.16. Configuración para el bloque Osmocom. Fuente: [8]	82
3.17. Configuración para el bloque DC Blocker. Fuente: [15].	82
3.18. Configuración para el bloque PLL Carrier Tracking. Fuente: [16]	84
3.19. Configuración para el bloque Interpolating FIR Filter. Fuente: [15]	84
3.20. Configuración para el bloque File Sink. Fuente: [17]	85
3.21. Configuración para el bloque File Source. Fuente: [18]	86
3.22. Receptor RF implementado en GNU Radio	87
3.23. Demodulador Multiportadora implementado en GNU Radio	89
4.1. Señal OFDM receptada versus señal OFDM simulada	94
4.2. Señal GFDM receptada versus señal GFDM simulada	95
4.3. Constelaciones 4-QAM para OFDM y GFDM	95
4.4. Comparación del BER vs Distancia para modulaciones OFDM y GFDM.	97
4.5. Comparación del BER vs SNR para modulaciones OFDM y GFDM.	98
A1. Ejecución de los experimentos 1 y 2	111

Índice de tablas

2.1. Características técnicas de los dispositivos FPGA. Fuentes: [9] [19] [20] [10] [11] [21]	30
2.2. Escala fundamental de Saaty. Fuente: [22]	35
2.3. Ejemplo de Matriz de Comparación Pareada.	36
2.4. Tabla de referencia para el índice de consistencia aleatorio. Fuente: [23]	38
3.1. Matriz de comparación de criterios.	44
3.2. Matriz de comparación de criterios normalizada.	44
3.3. Matriz de comparación para el criterio: Hardware RF	46
3.4. Matriz de comparación para el criterio: Reloj Interno	46
3.5. Matriz de comparación para el criterio: DAC	47
3.6. Matriz de comparación para el criterio: Laboratorio	47
3.7. Matriz de comparación para el criterio: LUTs	47
3.8. Matriz de comparación para el criterio: Flip Flop	48
3.9. Matriz de comparación para el criterio: IDE	48
3.10. Matriz de resultados del análisis AHP.	49
3.11. Parámetros de señales multiportadoras	70
3.12. Descripción de los bloques y señales en el proceso de cuantización uniforme	74
3.13. Parámetros de configuración del bloque Osmocom Source de GNU Radio.	83
4.1. Resultados del experimento 2.	96

Dedicatoria

A mis padres Zoila y Carlos por todo su apoyo incalculable de aportes a mi vida y por ser el pilar fundamental en mi formación personal y profesional. A mis hermanos con quienes he compartido toda mi vida les agradezco inmensurablemente todo su sacrificio y cariño entregado, a todos ustedes dedico este trabajo de integración curricular. Gracias por todo.

Jhonatan

Dedicatoria

Dedico este trabajo de integración curricular a mis padres Henry y Diana, quienes me brindaron su apoyo incondicional durante toda mi formación académica y representaron mi primera escuela en la instrucción de valores, ayudándome a convertirme en una persona de bien.

Mauricio

Agradecimientos

Agradecemos a nuestros tutores, Ing. Kenneth Palacio e Ing. Lizandro Solano, por su apoyo académico y emocional durante el desarrollo de este trabajo. Su experiencia y compromiso han sido esenciales para su culminación.

También agradecemos a los ingenieros Ing. Fabian Astudillo e Ing. Alcides Araujo por facilitar el acceso a los equipos FPGA necesarios a través de la Universidad de Cuenca.

Finalmente, extendemos nuestro agradecimiento a todos los ingenieros que han contribuido a nuestra formación académica, cuyo conocimiento se refleja en este trabajo de integración curricular.

Henry Mauricio Castro Mendoza

Jhonatan Paúl Cambisaca Sánchez

Generalidades

1.1. Introducción

Los sistemas de comunicación móvil se han convertido en una herramienta universal indispensable, lo que ha impulsado su evolución continua desde su creación. Durante el apogeo de la tecnología 3G, se identificó la necesidad de incrementar tanto la velocidad como la capacidad de los sistemas de comunicación existentes. En marzo de 2003, la *3rd Generation Partnership Project* (3GPP) realizó un Estudio de Viabilidad de OFDM (*Orthogonal Frequency Division Multiplexing*) para la mejora de UTRAN (*Universal Terrestrial Radio Access Network*) [24]. Según la 3GPP, OFDM es la técnica adecuada para la multiplexación y el acceso al medio, especialmente para los servicios multimedia que requieren alta velocidad y robustez en la transmisión.

En diciembre de 2005, OFDM fue reconocida como la técnica idónea para el acceso al medio, y su primera estandarización fue aprobada en 2007 [1]. Actualmente, las redes móviles 4G están ampliamente desplegadas a nivel mundial. Sin embargo, la tecnología de redes móviles vigente enfrenta limitaciones en cuanto a la velocidad de transmisión y las técnicas de multiacceso, lo que ha generado la necesidad de utilizar modulaciones de orden superior para transmitir mayor cantidad de información dentro del mismo ancho de banda.

Una de las técnicas propuestas como candidata para su uso en las redes móviles 5G es la tecnología *Generalized Frequency Division Multiplexing* (GFDM), caracterizada por utilizar filtros formadores de pulso con formas de onda más flexibles y no necesariamente ortogonales, a diferencia de OFDM. Aunque GFDM es considerada una alternativa viable para 5G, la literatura revela un vacío en el desarrollo de sistemas transmisores multiportadora en plataformas FPGA (*Field-Programmable Gate Array*), particularmente en lo que respecta a la implementación de un modulador GFDM en este tipo de hardware, el mismo que por su versatilidad facilita el desarrollo de prototipos y sistemas comerciales.

El presente trabajo tiene como objetivo diseñar e implementar un sistema de transmisión multiportadora sobre una plataforma FPGA, enfocándose en resolver la complejidad de configurar un sistema de multiplexación por división de frecuencia en una plataforma de descripción de hardware como VHDL (*VHSIC Hardware Description Language*). La evaluación del sistema se realizará a través de un SDR (*Software-Defined Radio*) que utiliza GNU Radio para el proceso de demodulación. Este proyecto busca comparar los sistemas de modulación OFDM y GFDM a nivel de prueba de concepto teórico-práctico con fines educativos y de investigación.

1.2. Objetivos

1.2.1. Objetivo general

Implementar un sistema de transmisión basado en esquemas de modulación multiportadora sobre una plataforma embebida FPGA.

1.2.2. Objetivos específicos

1. Analizar distintas plataformas FPGA disponibles en la Universidad de Cuenca y su viabilidad para el desarrollo de sistemas de comunicaciones multiportadora.
2. Implementar un sistema transmisor con modulación multiportadora sobre una plataforma embebida FPGA.
3. Evaluar el rendimiento del transmisor mediante un receptor implementado en *Software Defined Radio*.

1.3. Alcance

El trabajo de integración curricular se enfoca en la implementación de dos sistemas de transmisión multiportadora. El primer sistema corresponde a la modulación OFDM, descrita mediante un algoritmo de Transformada Discreta de Fourier Inversa (I-DFT) en lugar del uso convencional de la Transformada Rápida de Fourier Inversa (I-FFT) para el proceso de modulación. El segundo sistema emplea la modulación *Generalized Frequency Division Multiplexing* (GFDM), considerada una opción prometedora para la tecnología 5G. Entre otras, una de las metas de este trabajo es realizar una evaluación comparativa entre ambas modulaciones.

Ambos sistemas de modulación están precedidos por una premodulación QAM de 4 símbolos, utilizando la codificación Gray en el mapeo de símbolos. Fuera de esta codificación, el sistema de transmisión no incorpora técnicas de entrelazado ni mecanismos de detección y corrección de errores.

En la modulación OFDM, se emplea la I-DFT de seis símbolos, acompañada de un prefijo cíclico de 2 muestras. Por otra parte, en la modulación GFDM se utilizan tanto subsímbolos como subportadoras, con la capacidad de transmitir 6 símbolos pertenecientes a 4-QAM, durante una trama GFDM.

La metodología del trabajo de integración curricular consta de los siguientes pasos:

- Investigar y desarrollar las expresiones matemáticas que modelan el comportamiento de los sistemas de modulación multiportadora.
- Comprender el estándar IEEE 754 para gestionar la representación de números en punto flotante en Vivado (la suite de desarrollo para FPGA de Xilinx¹).
- Seleccionar una tarjeta FPGA que cumpla con los requisitos necesarios para el desarrollo del proyecto de titulación.
- Realizar la descripción del hardware utilizando VHDL para implementar una modulación multiportadora en una FPGA.
- Verificar el proceso de descripción del hardware del transmisor mediante depuración (Debug) para confirmar la síntesis del proyecto en el entorno de Vivado.
- Realizar pruebas y validar el sistema de transmisión multiportadora utilizando GNU Radio, demodulando la señal.

Como resultado del trabajo de integración curricular, se espera obtener un sistema de transmisión multiportadora funcional que pueda ser utilizado por la Universidad de Cuenca para fomentar el aprendizaje en la carrera de telecomunicaciones, específicamente en el área de laboratorio. Además, este proyecto establece un precedente en la implementación de la modulación GFDM sobre una plataforma de hardware FPGA.

1.4. Justificación

Los sistemas de comunicación multiportadora se exploran y evalúan comúnmente en la carrera de Telecomunicaciones mediante simulaciones en software, como MatLab [25] entre otros. Sin embargo, los lenguajes de alto nivel usados en estos simuladores, no permiten de manera general, una implementación eficiente y que pueda funcionar en tiempo real, además de consumir recursos de cómputo considerables. En este sentido, la implementación de una modulación multiportadora en una plataforma FPGA abre las oportunidades de implementar un procesamiento más eficiente, y en el que se pueda explorar diferentes arquitecturas. El uso de hardware reconfigurable además facilita el prototipado, la evaluación de consumo energético y limitaciones de procesamiento.

¹<https://www.xilinx.com/products/design-tools/vivado.html>

En contraste, la implementación de un algoritmo enteramente a través de software en un lenguaje de alto nivel es una herramienta valiosa para la visualización de un esquema de modulación. Se puede mencionar la facilidad de uso de herramientas como MatLab, en donde el lenguaje facilita muchos procedimientos mediante funciones que deslindan al usuario cualquier necesidad de conocer la arquitectura sobre la cual se ejecuta el código (que para Matlab corre sobre una máquina virtual de Java). No obstante, la implementación de una modulación multiportadora utilizando un lenguaje de bajo nivel como VHDL, posibilita reducir el uso de recursos en comparación a aquel que demanda una aplicación por software sobre un computador. Una implementación en hardware adicionalmente ofrece múltiples oportunidades de aprendizaje, y crea precedentes para futuros proyectos en la misma línea de trabajo. Realizar la implementación de un sistema de transmisión multiportadora sobre una plataforma embebida FPGA permitirá comprender mejor los desafíos que presentan estos esquemas de modulación desde el punto de vista de la práctica de ingeniería y su convergencia con la teoría. La experiencia adquirida en la implementación física proporcionará una base sólida para el diseño de sistemas de comunicaciones multiportadora más complejos.

Con propósitos de validación y verificación de la implementación del transmisor realizado en la plataforma FPGA, se utilizará un receptor basado en *Software Defined Radio* (SDR) [15]. El uso de SDR se ha masificado en años recientes, gracias a la flexibilidad que se maneja en este tipo de sistemas. SDR se utilizan hoy en día ampliamente en la investigación, la industria y la academia. En este contexto, se propone el uso de GNU Radio, en conjunción con herramientas de hardware RF externo comercial para la recepción de señales. La programación de código abierto (*Open Source*) que emplea GNU radio permite configurar los parámetros del demodulador para conseguir la información transmitida. Al mismo tiempo, el uso de GNU radio facilitará la creación e implementación de modelos de evaluación del sistema mediante la medición del BER, entre otros, utilizando herramientas de terceros para la generación de datos, por ejemplo, MatLab.

Marco Teórico

2.1. Sistemas de Modulación Multiportadora

Un sistema de modulación multiportadora divide su ancho de banda en varios segmentos pequeños, cada uno de los cuales es transmitido usando una subportadora distinta. Esto significa que la información se envía a través de varias subportadoras dentro del ancho de banda en lugar de usar una sola frecuencia portadora. Esta característica hace que los sistemas multiportadora sean resistentes al desvanecimiento rápido de los canales inalámbricos. Por esta razón, las modulaciones multiportadora se utilizan en las comunicaciones móviles, ya que su robustez ayuda a mitigar los efectos de la movilidad, la sección 3.3.2.1 proporciona una descripción más detallada.

2.1.1. Orthogonal Frequency Division Multiplexing: OFDM

La Multiplexación por División de Frecuencias Ortogonales (OFDM) es una técnica de modulación multiportadora que divide el espectro disponible en subportadoras ortogonales de banda estrecha para transmitir datos simultáneamente, lo que la hace resistente a interferencias y efectos multitrayecto [26] [27] [28]. Un aspecto interesante de OFDM, es que utiliza la Transformada Rápida de Fourier (FFT) y la Transformada Rápida de Fourier Inversa (IFFT) para generar subportadoras ortogonales para la transmisión paralela de señales digitales, asegurando una alta eficiencia espectral y propiedades anti dispersión [29]. Este esquema de modulación ha encontrado aplicaciones en diversos campos como enlaces de datos *Synthetic Aperture Radar* (SAR), redes inalámbricas 5G, canales acústicos submarinos y sistemas de comunicación por fibra óptica, donde ha mejorado significativamente la eficiencia y el rendimiento [30] [31]. A pesar de su robustez frente al multitrayecto en canales móviles, los sistemas OFDM son susceptibles a errores de frecuencia, lo que ha impulsado la investigación en arquitecturas de receptor con estimadores de frecuencia y fase para mitigar estos problemas y mejorar el rendimiento del sistema.

2.1.1.1. Beneficios de la modulación OFDM en comunicaciones inalámbricas

OFDM es una técnica de modulación y multiplexación que se ha vuelto especialmente vital en sistemas de comunicación 4G como Wi-Fi, WiMAX y LTE, ya que proporciona altas velocidades de transmisión con interferencias mínimas [32]. OFDM reduce la distorsión multitrayecto

y las interferencias de frecuencia, lo que la hace ideal para redes CDMA y mejora el acceso inalámbrico en redes 4G [33].

Hassan, en su libro [34], presenta ciertas ventajas de un sistema de transmisión inalámbrico que utiliza OFDM:

- Los efectos del multitrayecto pueden ser contrarrestados utilizando OFDM, ya que esta modulación divide el espectro en varias subportadoras. De este modo, los desvanecimientos rápidos no afectan significativamente la información debido a los efectos del canal. Esto se debe a que dentro de esas subportadoras pueden introducirse portadoras piloto, que se utilizan para ecualizar el efecto del canal.
- OFDM se caracteriza por utilizar subportadoras ortogonales en frecuencia, lo cual se consigue mediante la transformada de Fourier. Esta característica permite un uso eficiente del ancho de banda, al posibilitar la superposición de portadoras ortogonales.
- Utilizar la Transformada Discreta de Fourier, lo que permite aumentar la tasa de bits, ya que posibilita incrementar el número de símbolos de premodulación antes de aplicar la IDFT en la etapa de modulación.

2.1.1.2. Modelo en Bloques del Sistema de Transmisión OFDM

El principio de la multiplexación por división de frecuencias ortogonales (OFDM) consiste en ubicar símbolos modulados $a_k^{(m)}$ en subportadoras ortogonales. Formalmente, la expresión para OFDM básico, sin prefijo cíclico y dentro del intervalo $(mT_u \leq t \leq (m+1)T_u)$ [1], se describe como:

$$x(t) = \sum_{k=0}^{N_c-1} x_k(t) = \sum_{k=0}^{N_c-1} a_k^{(m)} e^{j2\pi k \Delta f t} \quad (2.1)$$

Donde,

- x_k : Corresponde a la k-ésima subportadora modulada con frecuencia $f_k = k \cdot \Delta f$, [1].
- $a_k^{(m)}$: Es el símbolo modulado que será ubicado en la k-ésima subportadora durante el m-ésimo intervalo de símbolos OFDM, [1].
- T_u : Es el tiempo de símbolo de modulación por subportadora, [1].
- N_c : Cantidad de moduladores complejos, también determina la cantidad de símbolos modulados.
- Δf : Es el espaciado entre subportadoras y está definido como: $\Delta f = 1/T_u$ [1].

La Figura 2.1 ilustra visualmente el funcionamiento básico de un modulador OFDM, mismo que corresponde a la representación gráfica de la Ecuación 2.1, [1].

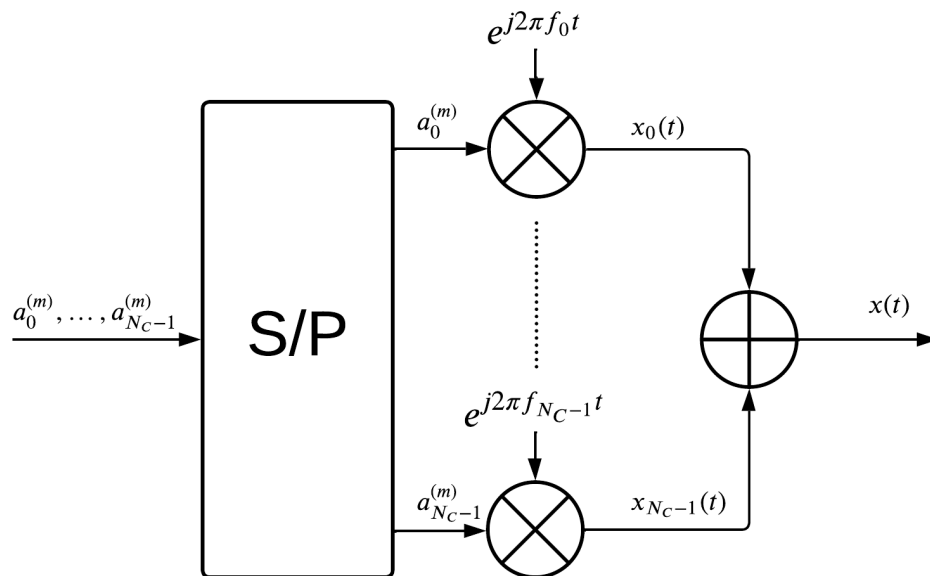


Figura 2.1: Principio fundamental de la modulación OFDM. Fuente: [1].

El proceso de modulación OFDM comienza con la conversión de bits, que corresponden a la fuente de información (*Information source encoding*), en símbolos modulados. Para realizar este procedimiento de premodulación se utilizan modulaciones digitales como QPSK o M-QAM. Esta conversión la realiza el bloque *Mapper*. Dado que la modulación OFDM se utiliza en comunicaciones inalámbricas móviles por su robustez frente a los efectos del canal variable, es necesario incluir una etapa de entrelazamiento para contrarrestar los efectos del desvanecimiento selectivo en frecuencia y evitar largas cadenas de bits con el mismo valor. Esta acción la realiza el bloque *Interleaver* [2].

Una vez terminado el proceso de entrelazamiento y premodulación, se procede a realizar una conversión de serial a paralelo (*bloque S/P*) de cada uno de los símbolos modulados. Luego de la etapa de paralelización, generalmente se insertan símbolos piloto que serán utilizados para estimar el comportamiento del canal o en la etapa de sincronización en el receptor. Posteriormente, los símbolos modulados se ubican en subportadoras ortogonales utilizando el algoritmo de la IFFT (*bloque IFFT*). Para controlar las emisiones fuera de banda, es necesario aplicar un prefijo cíclico (*CP insertion*). Adicionalmente, se puede incluir una etapa de filtrado para cumplir con la máscara de emisión espectral (SEM) [2]. Finalmente, la señal digital modulada se convierte en una señal analógica en banda base mediante el bloque (*D/A*). La señal analógica pasa por un proceso de amplificación y acoplamiento en radio frecuencia (RF) para

ser radiada al aire.

El funcionamiento completo del transmisor OFDM se puede observar en la Figura 2.2 a continuación.

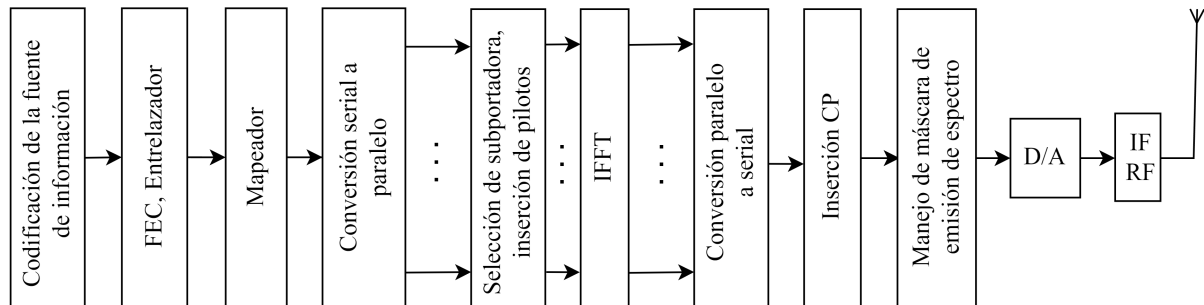


Figura 2.2: Diagrama de bloques de un sistema transmisor típico para una modulación OFDM. Fuente: [2].

2.1.1.3. Problemas Asociados con OFDM

Aunque OFDM se considera una opción adecuada para redes de banda ancha en frecuencias milimétricas, surge un desafío significativo debido a la demanda de los usuarios de latencia ultrabaja, que está vinculada a la sincronización y la ortogonalidad, elementos esenciales de OFDM [35]. Se sugiere adoptar un enfoque asincrónico que podría ser más prometedor que depender de algoritmos de sincronización avanzados, ya que estos últimos consumen más recursos de espectro y potencia [36] [3].

La necesidad de una trama más corta es crucial debido a las restricciones de tiempo críticas. En este contexto, la duración de símbolo de 67 microsegundos en OFDM de acuerdo al estándar LTE-A no parece ser una solución prometedora según [36].

Una modulación basada en *Orthogonal Frequency Division Multiplexing* (OFDM) presenta dos complicaciones notables que están directamente asociadas a su ventaja de ortogonalidad. La primera complicación ocurre si la señal presenta desfases o desplazamientos de frecuencia entre las subportadoras, comprometiendo así la ortogonalidad y la superposición de subportadoras [34]. La segunda complicación también afecta la ortogonalidad de las subportadoras, ya que para mantener esta ortogonalidad es necesario conservar una elevada relación de Potencia Pico a Promedio (PAPR). Sin embargo, mantener un valor de PAPR elevado requiere que el sistema cuente con dispositivos con un amplio rango lineal. Dado que el amplificador de potencia es un componente no lineal en el transmisor, podría causar la degradación del sistema [34].

2.1.2. Técnicas avanzadas de acceso múltiple para 5G

En el libro 5G Mobile Communications [37], se ha propuesto que la nueva capa física en las comunicaciones 5G se adapte a conceptos innovadores como la Multiplexación por División de Frecuencia Generalizada (GFDM) o la Multiportadora de Banco de Filtros (FBMC) [38]. La modulación GFDM es considerada una posible candidata debido a su estructura de bloques que agrupan intervalos de tiempo y subportadoras para el envío de información [36]. Esta modulación se caracteriza por su flexibilidad, ya que puede ajustarse a los requisitos de baja latencia. Inclusive se ha propuesto algoritmos FFT/IFFT para la implementación de GFDM en [39]. Por otro lado, la Multiportadora de Banco de Filtros (FBMC) es considerada esencial para la capa *Media Access Control (MAC)* emergente en 5G [40].

Los sistemas multiportadora son caracterizados por su filtro formador de pulso, por lo que para el caso de FBMC se utiliza un filtro diseñado para localizar mejor a las subportadoras [41]. Métodos como *Staggered Multi-Tone (SMT)* y *Cosine modulated Multi-Tone (CMT)* son fundamentales en FBMC para maximizar la eficiencia del ancho de banda [41] [42]. SMT emplea *QAM offset* y CMT utiliza modulación de amplitud de pulsos para la transmisión de símbolos. Ambos permiten la superposición de bandas adyacentes para optimizar la eficiencia del ancho de banda.

En la Figura 2.3 se resumen las principales características de las nuevas propuestas de acceso múltiple para la quinta generación de redes móviles [3].

La 5ta Generación de redes móviles requiere que los patrones de haces espaciales deseables puedan garantizar fácilmente el acceso múltiple por división espacial (SDMA). Es necesario seguir investigando, no solo en SDMA, sino también en SCMA, IDMA, FBMC y GFDM para lograr baja latencia y alto rendimiento en las futuras redes 5G [3].

2.1.2.1. Generalized Frequency Division Multiplexing: GFDM

GFDM, a diferencia de OFDM, contiene un mayor espaciamiento en frecuencia y una mejor eficiencia energética, lo que la hace idónea en comunicaciones móviles 5G. Estas características de GFDM se deben a la optimización del modelo del filtro del pulso, de tal forma que son eficientes en canales con desvanecimiento selectivo de frecuencia [43]. Otra característica principal de GFDM es establecer una estructura bidimensional para la transmisión de símbolos, en la cual estos representan información a transmitir del usuario. La estructura bidimensional de GFDM se visualiza en la Figura 2.4, la cual consiste en la distribución de símbolos en

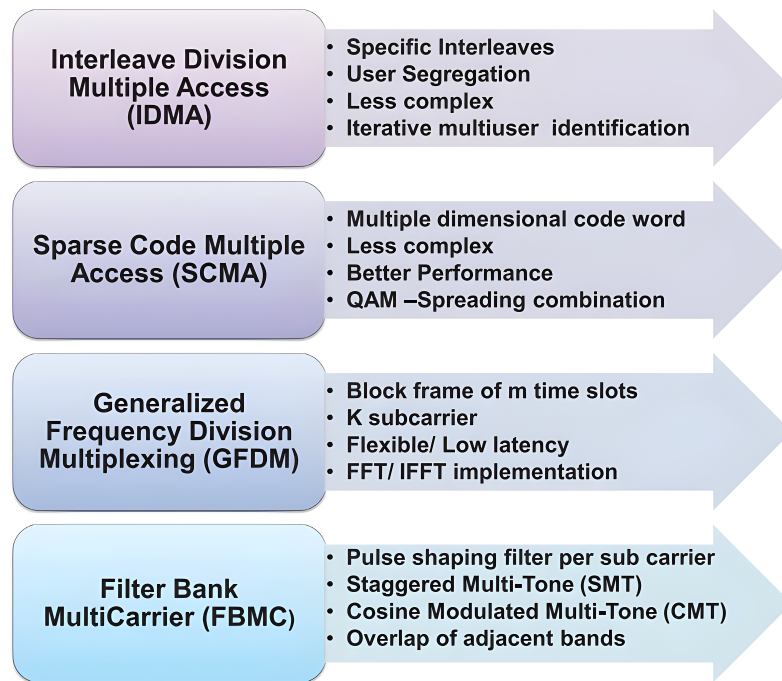


Figura 2.3: Acceso múltiple para la tecnología del futuro. Fuente: [3] © 2016 IEEE

instantes de tiempo y subportadoras. Esto se logra mediante, una convolución circular de un filtro formador de pulsos. Mediante ésta estructura se obtiene una mayor eficiencia espectral y flexibilidad sobre una implementación.

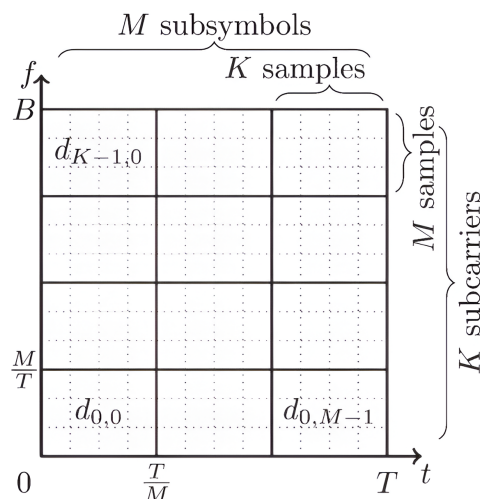


Figura 2.4: Partición del tiempo y la frecuencia GFDM. Fuente: [4] © 2014 IEEE

La Figura 2.5 muestra un esquema simplificado de un sistema de comunicaciones *Generalized Frequency Division Multiplexing (GFDM)*. En él se observa cómo los símbolos pertenecientes a una constelación QAM son obtenidos a partir de una fuente de información. Posteriormente, cada uno de los símbolos son mapeados a una subportadora y a un sub-símbolo (Figura 2.4). Tras definir la posición de cada uno de los símbolos, estos se transmiten a través de un filtro

formador de pulsos, como se muestra en la Figura 2.6. Finalmente, se agrega un prefijo cíclico a la señal resultante para evitar la interferencia intersímbolo. En el receptor, se debe ejecutar el proceso inverso al del sistema de transmisión. Además, el efecto del canal depende de las condiciones a las que se someten el sistema transmisor y receptor. Si se considera el caso de un canal móvil, tanto el transmisor como el receptor deben implementar medidas para mitigar los efectos de desvanecimiento.

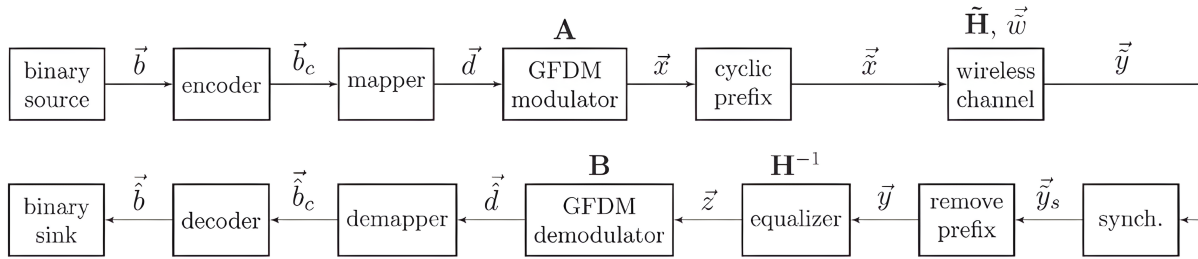


Figura 2.5: Sistemas de comunicaciones GFDM. Fuente: [4] © 2014 IEEE

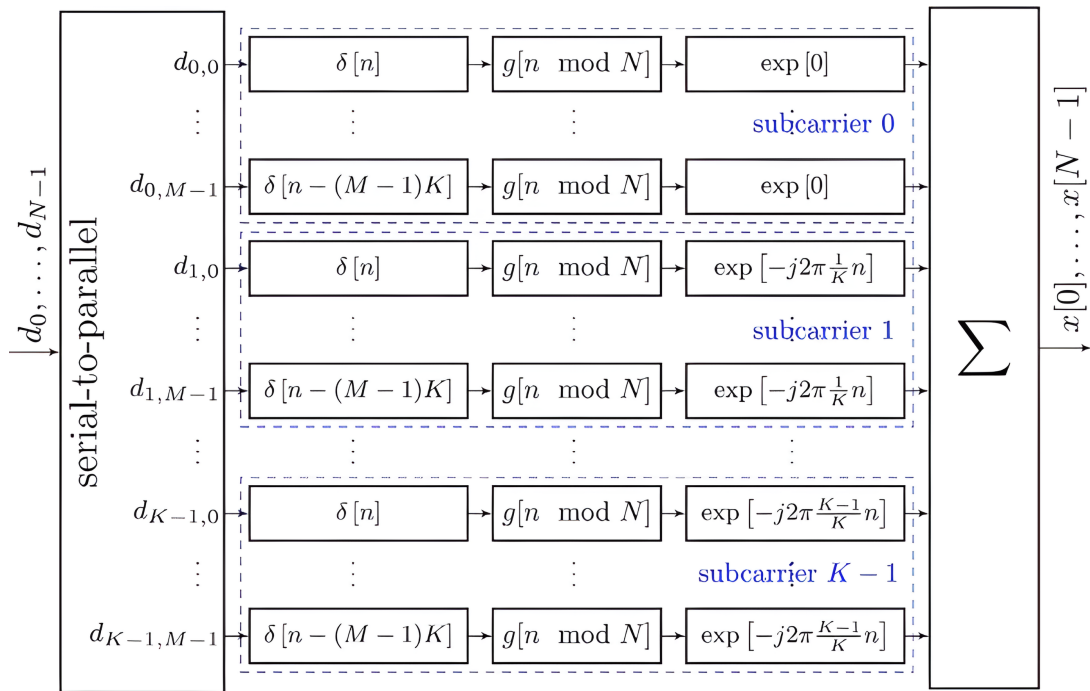


Figura 2.6: Modulador GFDM. Fuente: [4] © 2014 IEEE

Dentro de las investigaciones para GFDM se encuentra el uso recurrente de MIMO (*Multiple-Input Multiple-Output*). En [44], se ha explorado la integración de la técnica de modulación espacial (SM) con la Multiplexación por División de Frecuencia Generalizada (GFDM). La contribución principal de dicho trabajo radica en la creación de la estructura del sistema SM-GFDM y la evaluación de su *Bit Error Rate* (BER) en canales con desvanecimiento multitrayecto de Rayleigh.

La tecnología de transmisión de múltiples entradas y múltiples salidas (MIMO) es esencial para lograr velocidades de transmisión de datos más altas y una mayor eficiencia espectral [45]. La implementación de la transmisión MIMO involucra métodos, tales como, la codificación espacio-tiempo (STC), el espacio-tiempo por capas de Bell Labs (BLAST) y la modulación espacial (SM). STC busca maximizar la diversidad espacial para mejorar la eficiencia energética [46], [47]. Por otro lado, BLAST aumenta la capacidad transmitiendo múltiples flujos de datos independientes desde distintas antenas de transmisión. Variantes ampliamente utilizadas de BLAST incluyen BLAST diagonal (D-BLAST) [48] y BLAST vertical (V-BLAST) [49]. En el caso de SM, se utilizan los índices de antena de transmisión para llevar información adicional [50]. En este método, los bits de entrada determinan el punto de constelación en el conjunto de señales y el índice de la antena de transmisión activada. Durante cada intervalo de transmisión, la antena seleccionada transmite el símbolo elegido, mientras que las demás antenas permanecen en silencio, evitando así interferencias entre antenas (IAI) y la necesidad de sincronización entre antenas (IAS) [44].

Es esencial que cualquier propuesta de capa física destinada a redes inalámbricas 5G sea capaz de integrar la transmisión MIMO. Por lo tanto, existen estudios que exploran la modulación espacial (SM) en GFDM. Al combinar SM con GFDM, es posible desarrollar un sistema MIMO-GFDM que ofrece una flexibilidad significativa en la partición de tiempo y frecuencia, una baja emisión fuera de banda (OOB), una mejora en la eficiencia espectral y una sincronización flexible, todo ello sin incrementar significativamente la complejidad del sistema [44]. En [4], se muestra el detalle matemático de un sistema que implementa codificación de espacio-tiempo (STC) para dos antenas transmisoras y receptoras con GFDM. En el análisis se consideran canales selectivos de frecuencia, en los cuales el tiempo de coherencia del canal es mayor que el tiempo de símbolo de GFDM. Los resultados obtenidos establecen que GFDM, al contener una mejor resolución espectral por subportadora, permite una mejor ecualización en el dominio de la frecuencia, provocando que el SER (*Symbol Error Rate*) de STC-GFDM sea menor a STC-OFDM. En [51], se propone un esquema FDMA capaz de compartir subportadoras de una señal TR-STC (*Time Reversal Space Time Coding*) en GFDM entre múltiples usuarios. Con ello, a partir del conocimiento del estado del canal, se pueden asignar los subcanales disponibles a los usuarios, dando como resultado una mejora significativa en el rendimiento del SER (*Symbol Error Rate*). Por otro lado, en [52] y [53], se examina un sistema MIMO que utiliza multiplexación espacial y modulación GFDM, proponiendo esquemas de detección casi óptimos. Asimismo, en [54], se realiza el primer intento de aplicar la técnica de modulación espacial (SM) a un sistema MIMO-GFDM. Sin embargo, para evitar la auto interferencia en-

tre portadoras (ICI) en GFDM, se limita la transmisión a un solo símbolo por bloque GFDM, lo que resulta en una eficiencia espectral considerablemente inferior en comparación con las implementaciones convencionales.

2.2. Sistemas de Radio definidos por Software: SDR

La radio definida por software (SDR) representa un avance significativo en las comunicaciones inalámbricas, permitiendo la personalización y optimización de sistemas de manera sencilla y eficiente. A diferencia de los radios tradicionales de hardware fijo, los SDR utilizan una arquitectura flexible que combina hardware reconfigurable con software, proporcionando una versatilidad notable [55]. Esta tecnología innovadora permite que un único dispositivo opere en múltiples bandas de frecuencia, maneje diversos esquemas de modulación y se adapte a distintos estándares de comunicación, lo cual a marcando un antes y después en el ámbito de las telecomunicaciones [56].

La flexibilidad y facilidad de actualización que caracterizan a los SDR se traducen en un control preciso sobre los componentes de procesamiento de señales digitales, incluyendo mezcladores, filtros, moduladores y demoduladores [56]. Esto abre un abanico de posibilidades para el desarrollo de aplicaciones personalizadas y soluciones a medida.

Los últimos avances en el campo de la SDR han dado lugar a arquitecturas reconfigurables que minimizan el consumo de energía, la latencia y los costes, abordando así los retos que anteriormente limitaban su adopción generalizada [57]. Estas nuevas arquitecturas combinan componentes genéricos de radio con microcontroladores de bajo consumo y coprocesadores especializados, permitiendo la transmisión de audio de alta definición en tiempo real con un consumo energético mínimo y una latencia imperceptible. Este avance tecnológico facilita la adopción de sistemas SDR de próxima generación, promoviendo la innovación en sectores como la telefonía móvil, las comunicaciones de datos, la radiodifusión y la investigación científica.

La Figura 2.7 muestra el diagrama de bloques de un receptor SDR. En este diagrama, una señal de radiofrecuencia es recibida por la antena y procesada por el bloque (RF Tuner). Este bloque consiste en un amplificador RF seguido de un mezclador analógico que recibe las señales del amplificador y un oscilador local de frecuencia variable, garantizando así una señal modulada a una frecuencia intermedia (IF). El bloque (Tuner) opera de manera similar a un receptor superheterodino [5].

Después de demodular la señal a una frecuencia intermedia, se procede a convertir la señal analógica en una señal digital utilizando un convertidor ADC. Como se muestra en la figura, el siguiente paso es procesar la señal utilizando un *Digital Down Converter (DDC)*, compuesto por un mezclador, un oscilador y un filtro paso bajo de tipo FIR, todos ellos componentes digitales. Este proceso de conversión descendente digital es lo que caracteriza a un sistema SDR. El objetivo final del DDC es convertir la señal de pasa banda a banda base [5]. El bloque final constituye un *Digital Signal Processor (DSP)* que se encarga de procesar la señal de acuerdo con los requisitos del usuario.

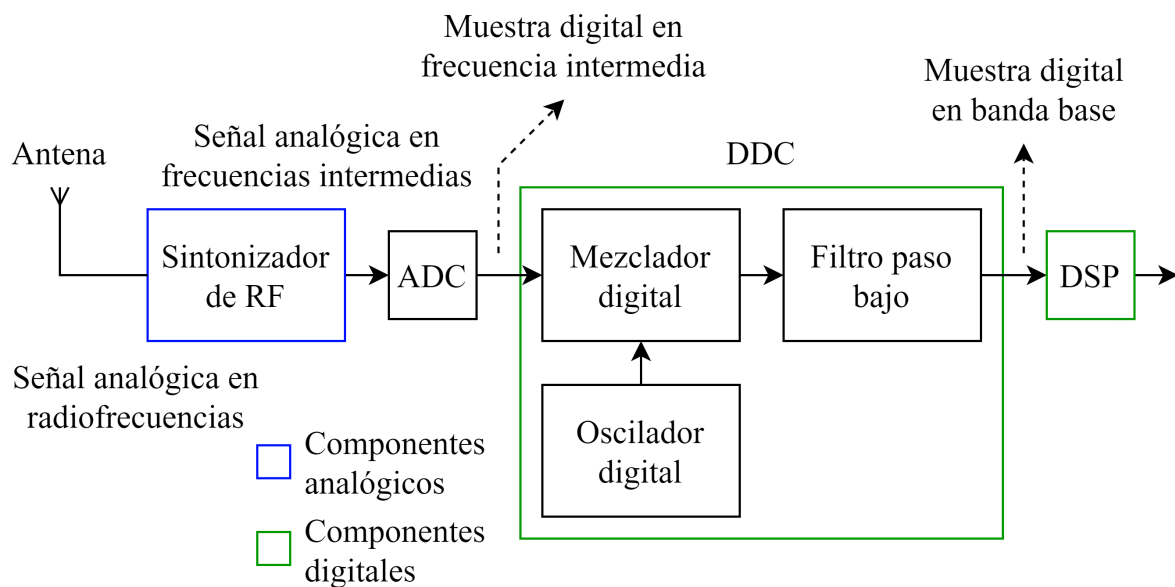


Figura 2.7: Diagrama de bloques del receptor SDR. Fuente: [5].

2.2.1. HackRF One

El dispositivo HackRF One mostrado en la Figura 2.8, desarrollado por Great Scott Gadgets, implementa tecnología *Software Defined Radio (SDR)* para la recepción de señales de radiofrecuencia. HackRF One es una herramienta de hardware de código abierto que es utilizado como un periférico mediante el bus de comunicaciones USB, es capaz de transmitir o recibir señales de radio de 1 MHz a 6 GHz [58].

El funcionamiento básico del dispositivo HackRF One se ilustra en la Figura 2.9. Primero, la señal RF es captada por una antena receptora con un puerto de entrada SMA. Dado que esta señal suele ser débil, se requiere amplificación para mejorar su potencia, por lo que pasa por una etapa de amplificación RF con una ganancia de 14 dB [7]. La señal amplificada no está exenta de componentes de alta frecuencia no deseadas. Para eliminarlas, se emplea un bloque *Low Pass Filter (LPF)* con una frecuencia de corte variable. Este filtro actúa como

una barrera selectiva, permitiendo el paso de las frecuencias de interés mientras bloquea las restantes. A continuación, la señal filtrada es procesada por medio de un mezclador en cuadratura junto a un oscilador local. Este proceso genera dos señales de banda base: una en fase (I) y otra en cuadratura (Q) [7].

Las señales I y Q se suman y luego se convierten en una señal digital de 8 bits por cada canal mediante un convertidor analógico-digital (ADC). Finalmente, la señal digital en banda base se entrega al usuario para su posterior procesamiento. Esta señal digitalizada contiene la información de la señal RF original, ahora en un formato que puede ser analizada.



Figura 2.8: Dispositivo HackRF One. Fuente: [6].

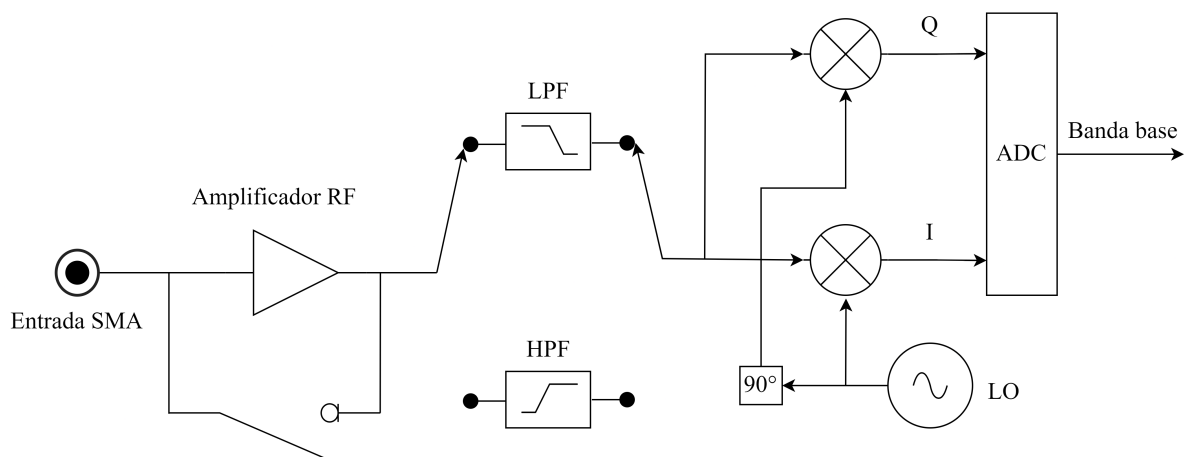


Figura 2.9: Diagrama de bloques del lado receptor del HackRF One. Fuente: [7].

2.2.2. GNU Radio

GNU Radio es un conjunto de herramientas de desarrollo de software libre y de código abierto, utilizado para crear aplicaciones de procesamiento de señales en tiempo real, especialmente en sistemas de comunicación inalámbricos, y diseñado para funcionar en ordenadores mono placa de bajo coste [15] [59].

GNU Radio proporciona una interfaz gráfica basada en bloques con un marco backend en Python, lo que permite a los usuarios redefinir los sistemas de comunicación predefinidos al modificar su programación en Python [60].

Para establecer la comunicación entre GNU Radio y el dispositivo HackRF One, se utiliza el bloque predefinido `osmocom source`, como se muestra en la Figura 2.10. Este bloque permite definir el tipo de dispositivo SDR que se utilizará, además de modificar parámetros clave como la frecuencia de muestreo y la frecuencia portadora.

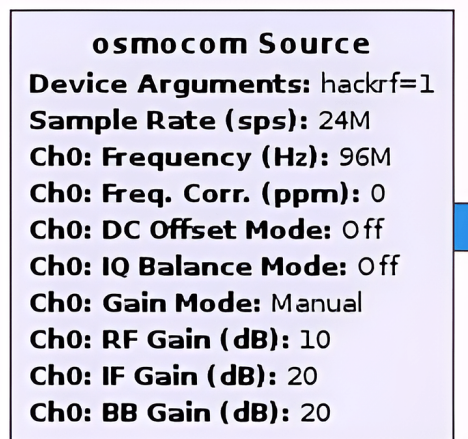


Figura 2.10: Bloque Osmocom de GNU-RADIO. Fuente: [8]

2.3. Field Programmable Gate Array

Un *Field Programmable Gate Array* (FPGA) es un circuito integrado que permite al diseñador de hardware configurar la lógica y las interconexiones sin necesidad de intervención del fabricante, facilitando así la implementación de una variedad de circuitos digitales [61]. A diferencia de los Circuitos Integrados de Aplicación Específica (ASICs), las FPGAs ofrecen un ciclo de diseño más sencillo, permiten una creación de diseños más rápida mediante el uso de Lenguajes de Descripción de Hardware (HDL), y tienen tiempos de fabricación más cortos lo que facilita el prototipado. Estas características las hacen altamente flexibles y les proporcionan

altas velocidades de procesamiento [62].

Las FPGAs están formadas por bloques lógicos configurables organizados en una matriz, con funciones lógicas y conexiones definidas por celdas SRAM. Debido a la naturaleza volátil de la SRAM, estas celdas requieren reconfiguración cada vez que el dispositivo se enciende [63].

Las plataformas de desarrollo FPGA disponibles en la Universidad de Cuenca son: Spartan 3E, Basys 3 y STEMLab 125-14. A continuación, se presentan las características técnicas más relevantes de cada una.

2.3.1. Características técnicas de los dispositivos FPGA

La información relevante sobre las especificaciones técnicas de las plataformas FPGA disponibles en la Universidad de Cuenca se presenta en la Tabla 2.1. Las propiedades evaluadas con “Si/No” indican la presencia o ausencia de la característica específica en las tarjetas FPGA.

Tabla 2.1: Características técnicas de los dispositivos FPGA. Fuentes: [9] [19] [20] [10] [11] [21]

Tarjeta FPGA	Hardware RF	Internal Clock	DAC	Laboratorio	LUTs	Flip Flop	IDE
Spartan 3E	No	50 MHz	12	No	9312	9312	ISE Design
Basys 3	No	100 MHz	No	No	20800	41600	Vivado
STEMLab 125-14	Si	125 MHz	14	Si	17600	35200	Vivado

2.3.2. Spartan 3E

La tarjeta Spartan 3E presentada en la Figura 2.11 es un dispositivo FPGA programable de tipo XC3S500E que cuenta con 500,000 puertas lógicas y un procesador RISC de 32 bits [9]. La tarjeta utiliza el IDE ISE Design Suite para realizar su programación de descripción de hardware. Sin embargo, ISE ya no cuenta con soporte del fabricante Xilinx (hoy bajo AMD). Sus características específicas se presentan en la Tabla 2.1.

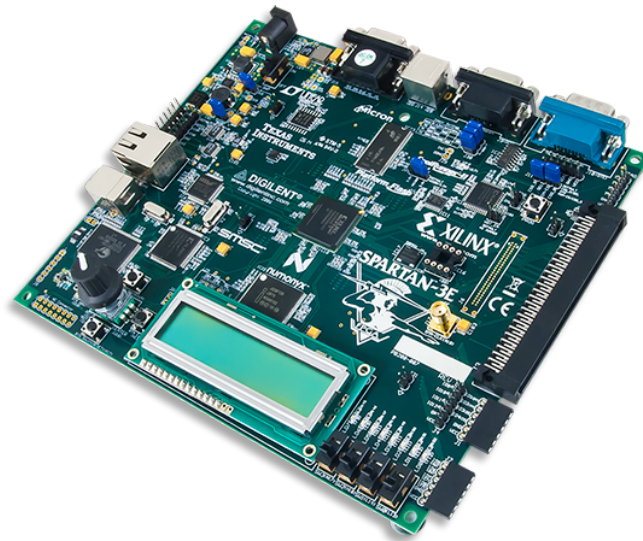


Figura 2.11: Dispositivo FPGA Spartan 3E. Fuente: [9]

2.3.3. Basys 3

La tarjeta Basys 3 mostrada en la Figura 2.12 es una herramienta de desarrollo muy útil para el aprendizaje de las FPGA, ya que cuenta con cuatro puertos de 16 pines digitales cada uno. Está equipada con una FPGA de alta capacidad XC7A35T-1CPG236C de Xilinx basada en la FPGA Artix 7 [10], lo que proporciona una arquitectura poderosa para la programación de circuitos complejos y su respectiva depuración. La Tabla 2.1 presenta un resumen de las características técnicas de la tarjeta Basys 3.

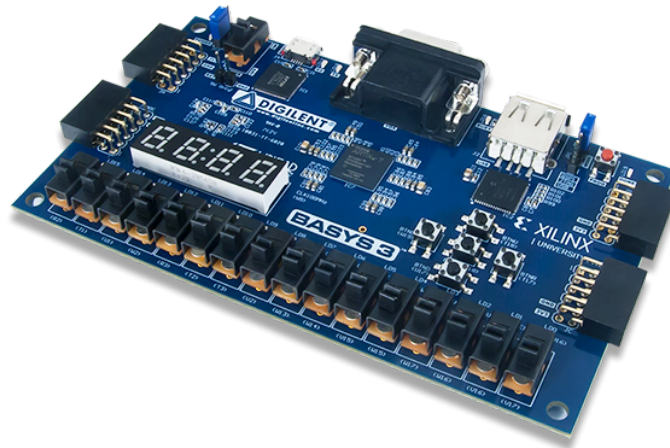


Figura 2.12: Dispositivo FPGA Basys 3. Fuente: [10]

2.3.4. STEMLab 125-14

El dispositivo STEMLab 125-14 de la empresa Red Pitaya presentada en la Figura 2.13, es una tarjeta con mejores prestaciones en comparación con otras tarjetas FPGA como las mencionadas arriba. Además de contar con una FPGA Xilinx Zynq 7010, también incluye un sistema operativo Linux que se ejecuta sobre un procesador Dual-Core ARM Cortex-A9 MPCore [11]. Esto permite interactuar con herramientas de laboratorio, como un osciloscopio y un analizador de espectros, que están integradas en el dispositivo. Un resumen de las características de la tarjeta FPGA se muestran en la Tabla 2.1.

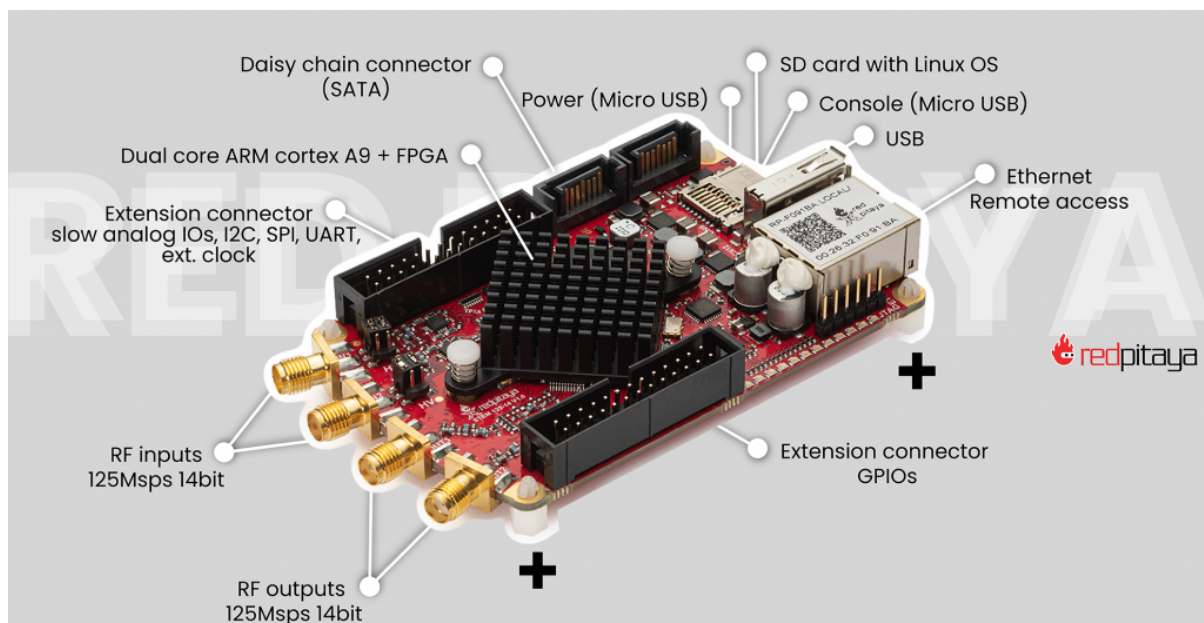


Figura 2.13: Dispositivo FPGA STEMLab 125-14. Fuente: [11]

2.4. Método Analytic Hierarchy Process

Para seleccionar la tarjeta FPGA adecuada para el desarrollo del trabajo de integración curricular, se utilizará el método *Analytic Hierarchy Process* (AHP) de Thomas Saaty. Este método permitirá comparar las tres opciones de tarjetas FPGA y tomar una decisión basada en siete criterios de selección.

El profesor Thomas L. Saaty creó el Proceso Analítico Jerárquico (*Analytic Hierarchy Process*, AHP) en los años 70 y 80 como técnica de toma de decisiones multicriterio, donde se utilizan comparaciones por pares para determinar escalas de prioridad [64] [65]. El AHP es útil en tareas de toma de decisiones y clasificación porque transforma las evaluaciones subjetivas en ponderaciones, lo que ayuda a elegir soluciones para situaciones complicadas [66]. Debido a la naturaleza humana, permite la derivación de escalas de relación a partir de comparaciones por pares, teniendo en cuenta pequeñas discrepancias en los juicios. Como marco para la resolución de problemas, el AHP divide las tareas en segmentos manejables y utiliza comparaciones directas por pares para determinar el orden de importancia dentro de cada jerarquía [67]. La claridad y practicidad de este enfoque metódico quedan demostradas por su valor en la toma de decisiones que abarcan costes, oportunidades, riesgos y recompensas.

2.4.1. Metodología del Análisis Jerárquico AHP

El *Analytical Hierarchy Process* (AHP) utiliza una escala de juicios absolutos para realizar la comparación entre opciones, permitiendo tratar las inconsistencias en los juicios para mejorar la coherencia. La metodología se puede utilizar para tomar decisiones complejas en una variedad de áreas, desde la administración pública hasta la distribución de recursos en grandes empresas [12].

Para realizar un análisis AHP de manera efectiva, Thomas Saaty en [12] sugiere seguir los siguientes pasos:

1. **Definición del problema y planteamiento del objetivo:** Definir claramente el problema que se pretende abordar y el alcance del análisis AHP, se considera como el objetivo a la solución del problema de decisión, por tanto debe ser específico, medible y alcanzable.
2. **Estructuración de la jerarquía de decisión:** La jerarquía debe descomponerse en niveles jerárquicos claros, empezando con el objetivo general, luego los criterios y subcriterios hasta terminar con las posibles opciones de decisión.

3. Comparaciones pareadas

- **Escala para las comparaciones pareadas:** Las comparaciones en pares deben realizarse utilizando una escala de comparación estándar, como la escala fundamental de Saaty, la cual se presenta con mayor detalle en la Sección 2.4.3.
 - **Consistencia:** Verificar la consistencia de las comparaciones utilizando métodos como el índice de consistencia (CI) y la razón de consistencia (CR).
4. **Determinación de las escalas de prioridad:** A cada criterio y subcriterio se le debe asignar un valor numérico que represente su importancia relativa dentro de su nivel jerárquico.
5. **Priorización de las alternativas:** Cálculo de la ponderación final al realizar la combinación del valor numérico de cada criterio o subcriterio en función de cada alternativa de solución.

2.4.2. Estructuración de los niveles de jerarquía

El proceso AHP se descompone el problema en una jerarquía de decisiones que consiste en:

- **Objetivo principal:** La meta última que se desea alcanzar.
- **Criterios:** Factores o atributos que influirán en la decisión.
- **Subcriterios (Opcional):** Desgloses más detallados de los criterios principales.
- **Alternativas:** Opciones entre las cuales se debe elegir.

La Figura 2.14 muestra gráficamente el resultado de la descomposición en jerarquía del problema a analizar.

2.4.3. Escala fundamental de Saaty

Tras establecer la jerarquía de decisión, se procede a realizar comparaciones pareadas entre los elementos de cada nivel. Para ello, se utiliza la escala de importancia desarrollada por Thomas Saaty, la cual asigna valores numéricos del 1 al 9 para representar la intensidad de la importancia relativa entre dos elementos. En esta escala, el valor 1 indica que los elementos son de igual importancia, mientras que el valor 9 representa la máxima diferencia en importancia entre dos elementos. La Tabla 2.2, que se presenta a continuación, muestra la escala fundamental de Comparación de Saaty.

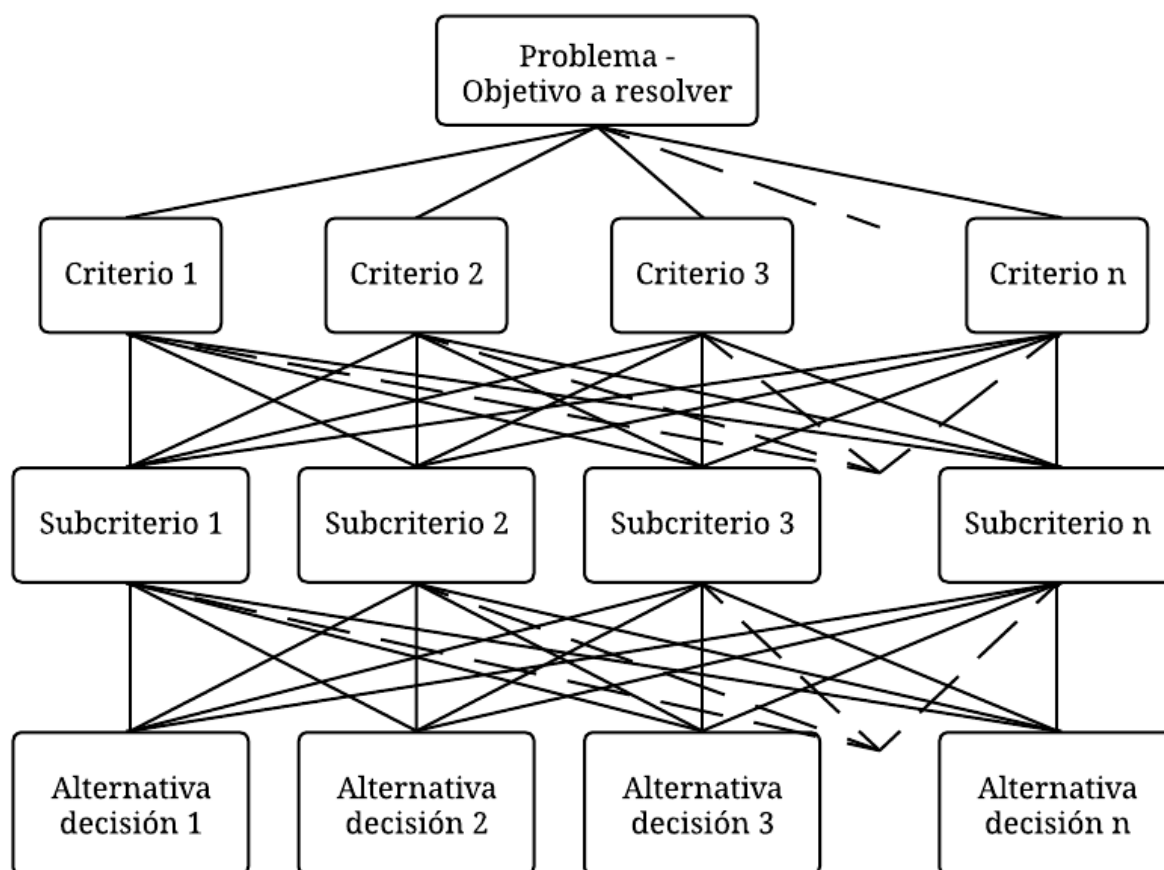


Figura 2.14: Estructura jerárquica del método AHP. Fuente: [12]

Tabla 2.2: Escala fundamental de Saaty. Fuente: [22]

Nivel de Importancia	Definición	Explicación
1	Igualmente importante	Ambos criterios poseen igual importancia
3	Moderadamente más importante	Un criterio es ligeramente más importante que otro
5	Fuertemente más importante	Considerable importancia de un criterio sobre otro
7	Extremadamente más importante	Evidencia extrema importancia de un criterio sobre otro
9	Absolutamente más importante	Orden de importancia con el valor más alto posible
2,4,6,8	Valores intermedios	Importancia media entre niveles adyacentes
$a_{ij} = 1/a_{ji}$	Recíprocos	Cuando el valor del criterio en la fila i en relación con el criterio en la columna j es igual a un entero de los mencionados anteriormente, entonces el valor de j en relación con i es igual a $1/a$.

2.4.4. Matrices de comparación y cálculo de prioridades

Cada elemento de la matriz cuadrada de comparaciones pareadas representa una comparación individual entre dos criterios o alternativas del mismo nivel jerárquico. Cada elemento de la matriz tiene un valor numérico que muestra la intensidad de la preferencia relativa entre los dos elementos que se comparan. La Tabla 2.2 muestra la escala Saaty que sirve como base para estos valores. Posteriormente, se calculan los vectores de prioridad utilizando la información de las matrices de comparaciones pareadas. Estos vectores asignan un peso ponderado a cada criterio o alternativa, representando la importancia relativa de cada elemento en su nivel jerárquico. La Tabla 2.3 a continuación muestra un ejemplo de aplicación.

Tabla 2.3: Ejemplo de Matriz de Comparación Pareada.

	Criterio 1	Criterio 2	Criterio 3	...	Criterio n
Criterio 1	1	a	b	...	c
Criterio 2	$1/a$	1	$1/d$	...	$1/e$
Criterio 3	$1/b$	d	1	...	$1/f$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Criterio n	$1/c$	e	f	...	1

2.4.5. Consistencia del Análisis

En el contexto de la toma de decisiones multicriterio mediante el Proceso Analítico Jerárquico (AHP), es fundamental asegurar la coherencia lógica en la jerarquía de prioridades, lo cual se logra a través de un análisis de consistencia [68]. El método AHP utiliza dos herramientas matemáticas, el Índice de Consistencia (CI) y el índice de Consistencia Aleatoria (RC), para evaluar y mejorar la consistencia en las comparaciones por pares [69] [70].

El CI es un indicador numérico que mide el nivel de inconsistencia en las comparaciones por pares y el tamaño de la jerarquía [71]. Para determinar el índice de consistencia CI es necesario definir la matriz cuadrada de n criterios $\mathbf{M}$ como:

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nn} \end{bmatrix} \quad (2.2)$$

Luego, es necesario determinar el valor del vector de ponderación correspondiente a la valoración de cada criterio, donde se requiere calcular la matriz normalizada E de valores, nor-

malizando cada columna de la matriz de criterios. El proceso de normalización se lleva a cabo dividiendo cada elemento de una columna por la suma de todos los elementos de dicha columna.

Formalmente, si M_{ij} representa el elemento en la fila i y columna j de la matriz original M , entonces el elemento e_{ij} de la matriz normalizada E se calcula como:

$$e_{ij} = \frac{M_{ij}}{\sum_{k=1}^n M_{kj}} \quad (2.3)$$

Donde, $\sum_{k=1}^n M_{kj}$ es la suma de todos los elementos de la columna j . Los elementos e_{ij} forman la matriz normalizada E .

$$E = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{nn} \end{bmatrix} \quad (2.4)$$

Ahora para determinar el vector de valores ponderados es necesario calcular el promedio por filas de la matriz normalizada, y como resultado se obtiene el vector de pesos $\vec{b}$.

$$\vec{b}_i = \frac{1}{n} \sum_{j=1}^n e_{ij} \quad (2.5)$$

Una vez se posee la matriz de cuadrada de criterios M y el vector de ponderación $\vec{b}$, entonces es posible calcular el índice de consistencia (CI), que corresponde a:

$$CI = \frac{n_{max} - n}{n - 1} \quad (2.6)$$

Donde:

- n_{max} : Es el resultado de la combinación lineal de la matriz M por el vector de ponderación $\vec{b}$.
- n : Cantidad de criterios utilizados por el método AHP.

Por otra parte, la relación de consistencia (CR) compara el valor del índice de consistencia (CI) con un Índice Aleatorio (RI) obtenido de tablas predefinidas, lo que permite a los responsables de la toma de decisiones evaluar la consistencia de las comparaciones y tomar decisiones bien fundamentadas [72]. La Tabla 2.4 muestra la referencia según [23] que será utilizada para determinar el valor del índice RI.

Tabla 2.4: Tabla de referencia para el índice de consistencia aleatorio. Fuente: [23]

n	1	2	3	4	5	6	7	8	9	10
Índice de Consistencia Aleatorio (RI)	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

El valor de **n** en la Tabla 2.4 corresponde al número de criterios que será utilizados por el análisis AHP.

Por último, se calcula la relación de consistencia CR de la forma:

$$CR = \frac{CI}{RI} \quad (2.7)$$

Para que una relación de consistencia sea considerada aceptable es necesario que cumpla la condición de $CR \leq 0.1$.

2.4.6. Síntesis de Prioridades

La síntesis que corresponde al último paso de un proceso AHP, consiste en combinar las prioridades locales (ponderaciones de criterios y subcriterios) para obtener las prioridades globales de las alternativas. Este proceso se realiza mediante una sencilla multiplicación, donde la prioridad global de una alternativa se calcula como la suma de los productos entre su desempeño en cada criterio y la ponderación correspondiente de ese criterio.

2.5. Trabajos Relacionados

En este apartado se revisan trabajos e investigaciones previas sobre la implementación de sistemas de transmisión multiportadora en plataformas FPGA. El objetivo es analizar el estado del arte, identificar las técnicas y metodologías más utilizadas, y comprender los retos abordados en estudios anteriores. Cabe mencionar el trabajo presentado en [73] y desarrollado en la Universidad de Cuenca, en el que se utiliza una plataforma FPGA basada en un SPARTAN-3E de Xilinx, para el desarrollo de un sistema de comunicaciones digitales. En ese trabajo se explora diferentes técnicas de modulación digital de portadora simple, y que en parte ha motivado el desarrollo del presente trabajo de integración curricular. A continuación mencionaremos algunos trabajos relacionados en el contexto de sistemas multiportadora.

En [74], se llevó a cabo la implementación de un sistema de comunicación OFDM mediante el uso de una plataforma FPGA con un modelo de punto-fijo. Este sistema se desarrolló mediante la construcción de bloques funcionales basados en el modelo matemático OFDM utilizando una modelización inicial del sistema en la plataforma Matlab y, posteriormente, la implementación apoyada de Xilinx System Generator for DSP (XSG) y Simulink. El diseño se fundamentó en el estándar IEEE 802.11a y se implementó en una tarjeta de desarrollo Xilinx Virtex-5. Para verificar el desempeño del sistema OFDM, se empleó la técnica de co-simulación de hardware, que ofrece una forma eficiente de validar y evaluar su rendimiento. El sistema se sometió a pruebas utilizando un enfoque de hardware en bucle, permitiendo el cálculo y la comparación de la tasa de error de bits (BER) en relación a los resultados del modelo realizado en Matlab. Los resultados demuestran la correcta implementación del sistema OFDM en banda base, ya que el transmisor requiere $15.6\mu s$ para generar las primeras muestras de la señal OFDM. Sin embargo, se observa una degradación del BER debido a que, con un SNR alto, se pierde precisión al representar niveles de ruido débil en el modelo de punto fijo implementado en la FPGA.

Por otra parte, en [75] se presenta el diseño y desarrollo de un sistema OFDM utilizando el lenguaje de programación VHDL. La implementación de este sistema se llevó a cabo en una tarjeta de desarrollo FPGA Virtex-6. El proceso de diseño involucró la creación y simulación del sistema OFDM con modulación 64-QAM. Su implementación se fundamentó en el procesamiento de señales digitales de alta velocidad, haciendo uso de módulos IFFT/FFT. Para verificar la precisión de los resultados, se realizaron simulaciones en MATLAB, lo que permitió la comparación entre los resultados obtenidos en dichas simulaciones y aquellos generados por Xilinx ISE. Estos resultados demuestran la correcta implementación de OFDM en FPGA, ya que la señal generada se asemeja a la simulación realizada en MATLAB en el dominio de la frecuencia. La verificación y validación del hardware se completaron a través del empleo del kit FPGA Xilinx Virtex-6.

El modelo matemático y la operación base de GFDM se presentan en [76] y [77]. En el estudio [76], se implementan los esquemas de modulación y demodulación GFDM en el simulador Vienna 5G. Su aporte consiste en desarrollar un ecualizador en el dominio del tiempo enfocado en canales desvanecimiento (Fading), cuya idea principal es dividir el canal en sub-bloques para luego promediarlos. Para encontrar el número de sub-bloques que minimizan el *Bit Error Rate* (BER), se utilizan algoritmos genéticos para optimizar los modelos de regresión que predicen el BER del sistema. Los resultados demuestran que el esquema GFDM implementado y su ecualización superan la tasa de error de otras modulaciones, tales como OFDM. Por otra

parte, en [77] se proponen técnicas de multiplexación espacial para GFDM mediante el uso de un modelo lineal acotado en el que solo se considera la influencia del ruido Gaussiano. Los resultados demuestran que GFDM no es un reemplazo general para OFDM, dado que presenta una mayor complejidad y un rendimiento variable en la demodulación de GFDM. Por tanto, se propone que GFDM puede ser utilizado en escenarios con selectividad en frecuencia, siendo apto para aplicaciones especializadas de radio móvil (SMR), radio cognitiva (CR) y comunicación ultra confiable de baja latencia (URLLC).

En [78], se realiza una implementación de GFDM sobre una plataforma FPGA. Esta implementación utiliza el software NI PXI y el módulo LabView FPGA con el objetivo de realizar una prueba de concepto de un transmisor GFDM. Por lo tanto, sus resultados son preliminares, ya que se limitan a mostrar la forma de onda y el espectro de potencia de la señal GFDM generada por la FPGA, sin realizar ningún tipo de verificación adicional.

Finalmente, una implementación de GFDM se encuentra disponible en GNU Radio [79]. Esta representa una implementación de radio definida por software (SDR), la cual proporciona información detallada sobre cómo aprovechar la flexibilidad de SDR para simulaciones y pruebas en campo. Esta implementación también permite la integración de interfaces con otros lenguajes de programación. Además, este estudio realizó una prueba de campo que consistió en una transmisión inalámbrica, donde el procesamiento de señales se llevó a cabo en una computadora portátil con un hardware Intel Core i7-6700HQ y 16 GB de RAM. Los resultados de esta transmisión demuestran la correcta modulación y demodulación de GFDM, utilizando aproximadamente el 90% de un núcleo del *Central Processing Unit* (CPU) con una frecuencia de muestreo de 3.125MHz . Sin embargo, al aumentar la frecuencia de muestreo de la señal, se requiere el uso de más de un núcleo del CPU, llegando a necesitar aproximadamente 3 núcleos con una frecuencia de muestreo de 25MHz .

Desarrollo del Sistema

En este capítulo se realiza el análisis AHP para seleccionar la tarjeta más adecuada para el proyecto. Se describe la implementación del sistema de transmisión multiportadora, detallando el funcionamiento de cada uno de sus bloques. Finalmente, se desarrolla el módulo receptor utilizado para evaluar el sistema de transmisión.

3.1. Análisis de los dispositivos FPGA disponibles en la Universidad de Cuenca

La Universidad de Cuenca proporcionó el acceso a tres tarjetas FPGA para el desarrollo de este trabajo de integración curricular, que corresponden a: **Spartan 3E**, **Basys 3** de la empresa Digilent y **STEMlab 125-14** de la empresa Red Pitaya. Con el objetivo de iniciar el desarrollo de la programación de descripción de hardware y enfocados en abarcar el primer objetivo específico de este proyecto, se llevó a cabo una evaluación comparativa de las tarjetas FPGA disponibles utilizando el método Analytic Hierarchy Process para la toma de decisiones descrito en la Sección 2.4 del Marco Teórico.

La selección de la tarjeta FPGA no se basó exclusivamente en la presencia un solo criterio de evaluación como por ejemplo el DAC, ya que otros aspectos técnicos como la frecuencia del reloj interno, la cantidad de LTUs, y Flip Flops, y el entorno de desarrollo integrado (IDE) también son determinantes. Para una evaluación objetiva, se aplicó el método de Proceso de Análisis Jerárquico (AHP), que permitió ponderar siete criterios relevantes. Esta metodología aseguró que la decisión incluyera una combinación de características técnicas importantes de cada tarjeta, permitiendo así una selección justificada de la tarjeta FPGA para el desarrollo del trabajo de integración curricular.

3.1.1. Definición el Problema

El objetivo del análisis es seleccionar la tarjeta FPGA adecuada que pueda sintetizar la descripción de hardware para una modulación multiportadora y que además tenga la capacidad de generar una señal analógica apta para ser radiada de forma inalámbrica.

3.1.2. Diseño de la estructura jerárquica con AHP para la selección de la tarjeta FPGA

El análisis AHP considera siete criterios de evaluación para las tres opciones de tarjetas FPGA disponibles para ser utilizadas en el desarrollo del trabajo de integración curricular. Para

seleccionar una sola tarjeta, se presenta a continuación la descomposición del modelo jerárquico de acuerdo con los siguientes parámetros: Problema, Criterios, Opciones. El esquema de jerarquía se muestra en la Figura 3.1.

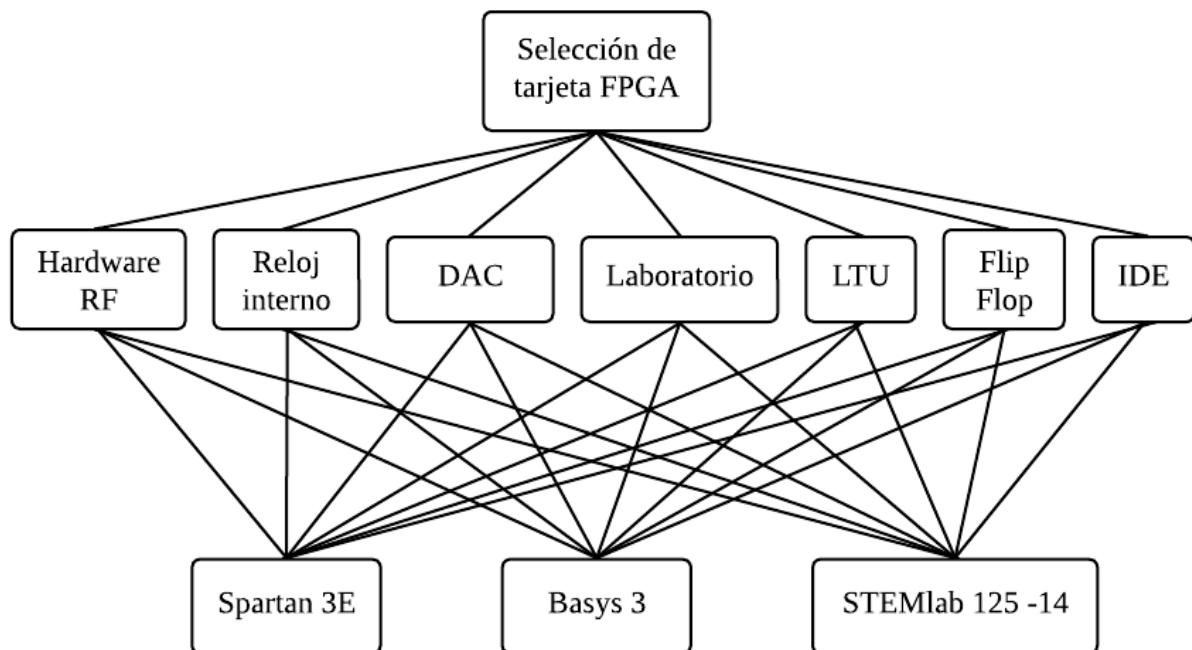


Figura 3.1: Esquema Jerárquico para la selección de la tarjeta FPGA.

En la Figura 3.1 se observa que el modelo de estructura jerárquica tiene tres niveles. El primer nivel de la estructura jerárquica define el objetivo: seleccionar la tarjeta FPGA adecuada para el desarrollo del trabajo de integración curricular. El segundo nivel de jerarquía está compuesto por los siete criterios de evaluación detallados en la Subsección 3.1.3. Finalmente, el tercer nivel incluye las tres opciones de tarjetas FPGA disponibles en la Universidad de Cuenca para el desarrollo del trabajo de integración curricular.

3.1.3. Criterios de evaluación para el análisis del Proceso de Jerarquía Analítica (AHP)

El análisis AHP emplea siete criterios para determinar la selección óptima de la tarjeta FPGA para el desarrollo de este trabajo de integración curricular. Estos criterios son: Hardware RF, Reloj Interno, Resolución del DAC, Laboratorio, LUTs, Flip Flop e IDE. Los criterios se establecieron en función de los recursos y herramientas de programación e implementación utilizados en el desarrollo del trabajo de integración curricular. A continuación, se proporciona una descripción de cada uno de los criterios:

- **Hardware RF:** Considera la implementación de circuitería de Radio Frecuencia presente

en la tarjeta FPGA, que será utilizado para transmitir la señal al medio inalámbrico utilizado en la transmisión de información de la modulación multiportadora hacia el hardware de recepción.

- **Reloj Interno:** Este criterio se relaciona directamente con la frecuencia del reloj interno de cada tarjeta analizada, sin incluir la posibilidad de uso de relojes externos.
- **Resolución del DAC:** Este parámetro hace referencia a la presencia de un conversor Digital a Analógico (DAC) en la tarjeta FPGA, ya que, la inclusión de esta característica permitiría evitar dificultades en el diseño y construcción de circuitos adicionales.
- **Laboratorio:** La tarjeta STEMLab 125-14 de la empresa Red Pitaya incluye entradas y salidas para señales analógicas que, mediante su sistema operativo interno, permiten visualizar el espectro y el comportamiento temporal de dichas señales. Por esta razón, la presencia de este elemento se considera relevante para la depuración y verificación de la descripción de hardware.
- **LUTs:** El número de LUTs (*Look-Up Tables*) determina la cantidad de entradas que pueden ser asignadas a la salida para un bloque de memoria de solo lectura [80], es un indicador importante de la capacidad de la FPGA para manejar entradas y salidas.
- **Flip Flops:** La cantidad de Flip Flops está asociada a la capacidad de diferenciar entre flancos de reloj, así como al establecimiento o reseteo síncrono o asíncrono dentro de procesos en VHDL [81]. Por ello, su cantidad es relevante para la descripción de hardware.
- **IDE:** Existen dos posibles entornos de desarrollo para realizar la descripción de hardware en el desarrollo de este trabajo de integración curricular:
 - La FPGA Spartan 3E, utiliza **ISE Design Suite** que es un software, desarrollado por Xilinx, pero sin soporte oficial desde octubre de 2013.
 - Las tarjetas Basys y STEMLab 125-14 utilizan **Vivado** que es la suite de diseño actual de Xilinx.

3.1.4. Comparación de los criterios

La Tabla 3.1 presenta la matriz cuadrada de comparación de criterios, en la cual se ha establecido un orden jerárquico para los criterios. Los elementos de la matriz han sido rellenados de acuerdo con la escala de Saaty 2.2.

Tabla 3.1: Matriz de comparación de criterios.

Criterios	Hardware RF	Reloj interno	DAC	Laboratorio	LUTs	Flip Flop	IDE
Hardware RF	1	3	5	7	8	7	9
Reloj interno	1/3	1	5	3	5	5	7
DAC	1/5	1/5	1	3	3	3	5
Laboratorio	1/7	1/3	1/3	1	1	2	4
LUTs	1/8	1/5	1/3	1	1	1	5
Flip Flop	1/7	1/5	1/3	1/2	1	1	3
IDE	1/9	1/7	1/5	1/4	1/5	1/3	1

Una vez establecida la matriz de comparación entre pares, se procede a calcular el vector de ponderación según el procedimiento descrito en la subsección 2.4.5, referente a la consistencia del análisis. Para ello, comenzamos calculando la matriz normalizada E .

Tabla 3.2: Matriz de comparación de criterios normalizada.

Criterios	Hardware RF	Reloj interno	DAC	Laboratorio	LUTs	Flip Flop	IDE
Hardware RF	0.487	0.591	0.410	0.444	0.417	0.362	0.265
Reloj interno	0.162	0.197	0.410	0.190	0.260	0.259	0.206
DAC	0.097	0.039	0.082	0.190	0.156	0.155	0.147
Laboratorio	0.070	0.066	0.027	0.063	0.052	0.103	0.118
LUTs	0.061	0.039	0.027	0.063	0.052	0.052	0.147
Flip Flop	0.070	0.039	0.027	0.032	0.052	0.052	0.088
IDE	0.054	0.028	0.016	0.016	0.010	0.017	0.029

El vector de ponderación se obtiene al calcular el promedio de los elementos de cada fila en la matriz normalizada E presentada en la Tabla 3.2.

$$\text{Vector de ponderación} = \vec{b} = [0.4250 \quad 0.2406 \quad 0.1239 \quad 0.0713 \quad 0.0631 \quad 0.0514 \quad 0.0245]^T \quad (3.1)$$

Verificación de la razón de consistencia (CR)

La razón de consistencia del análisis AHP está definida por la relación:

$$CR = \frac{CI}{RI}$$

Donde, el índice de consistencia se calcula utilizando:

$$CI = \frac{n_{max} - n}{n - 1} \quad (3.2)$$

El valor n_{max} se obtiene al multiplicar la matriz de comparación de criterios $\mathbf{M}$, presentada en la Tabla 3.1, por el vector de valores ponderados $\vec{b}$, mostrado en (3.1). A continuación, se presenta su cálculo:

$$x = \mathbf{M} \times \vec{b} = \begin{bmatrix} 3.3515 \\ 1.9603 \\ 0.9372 \\ 0.5176 \\ 0.4510 \\ 0.3739 \\ 0.1785 \end{bmatrix}$$

Y por tanto el n_{max} será:

$$n_{max} = \sum_{j=1}^n x_j = 7.76996 \quad (3.3)$$

El índice de consistencia CI es:

$$CI = \frac{n_{max} - n}{n - 1} = \frac{7.76996 - 7}{7 - 1} = 0.128327 \quad (3.4)$$

Considerando la Tabla 2.4 correspondiente a los valores de RI. El valor del índice aleatorio es: $RI = 1.32$ ya que $n = 7$.

Entonces la relación de consistencia tiene un valor de:

$$CR = \frac{CI}{RI} = \frac{0.128327}{1.32} = 0,09722 \quad (3.5)$$

Se puede concluir que la escala de jerarquía establecida para los criterios posee consistencia lógica, dado que el índice de consistencia (CR) es menor que 0.1.

3.1.5. Comparación por pares para cada criterio

Para la comparación pareada entre opciones, se utiliza la misma lógica descrita en la subsección 3.1.4. Sin embargo, en este caso, la matriz cuadrada será evaluada en relación con las opciones de elección. En el presente trabajo, las opciones se refieren a los dispositivos FPGA disponibles para el desarrollo de este trabajo de integración curricular.

La comparación entre pares se realiza para cada criterio, y la matriz de comparación se llena de acuerdo con la escala fundamental de Saaty, presentada en la Tabla 2.2.

3.1.5.1. Hardware RF

La característica de hardware RF se consideró de mayor prioridad con respecto a los otros seis criterios evaluados. La razón principal es que, al contar con un hardware RF integrado, la implementación del trabajo de integración curricular se centraría directamente en el diseño de un sistema de transmisión multiportadora y su descripción en hardware, en lugar de enfocarse en el diseño de circuitos de RF.

Tabla 3.3: Matriz de comparación para el criterio: Hardware RF

Alternativas	Spartan 3E	Basys 3	STEMlab 125-14	Promedio
Spartan 3E	1	1	1/9	0.090909091
Basys 3	1	1	1/9	0.090909091
STEMlab 125-14	9	9	1	0.818181818

Los elementos de la Tabla 3.3 reflejan directamente si una tarjeta cuenta o no con la característica de hardware RF.

3.1.5.2. Reloj Interno

La escala de Saaty empleada para evaluar la característica del reloj interno se basa en comparar directamente las frecuencias de reloj de cada tarjeta. Dado que la tarjeta STEMlab posee el reloj con mayor frecuencia, se le asigna una prioridad superior respecto a las otras tarjetas, como se muestra en la Tabla 3.4 a continuación.

Tabla 3.4: Matriz de comparación para el criterio: Reloj Interno

Alternativas	Spartan 3E	Basys 3	STEMlab 125-14	Promedio
Spartan 3E	1	1/9	1/3	0.101316701
Basys 3	9	1	1/3	0.378516979
STEMlab 125-14	3	3	1	0.52016632

3.1.5.3. Conversor digital analógico (DAC)

La evaluación del criterio del conversor DAC se basa en la presencia o ausencia de esta característica en la tarjeta. Al estar presente en las tarjetas Spartan 3E y STEMLab, la evaluación en la escala de Saaty se realiza considerando la resolución del DAC, donde la tarjeta STEMLab supera a la Spartan 3E, como se aprecia en la Tabla 3.5.

Tabla 3.5: Matriz de comparación para el criterio: DAC

Alternativas	Spartan 3E	Basys 3	STEMlab 125-14	Promedio
Spartan 3E	1	9	1/3	0.315898895
Basys 3	1/9	1	1/9	0.052193894
STEMlab 125-14	3	9	1	0.631907211

3.1.5.4. Laboratorio

Para la característica de laboratorio, se asigna un valor de 9 a la tarjeta STEMLab, ya que posee esta característica. En contraste, a las demás tarjetas se les asigna un valor de 1.

Tabla 3.6: Matriz de comparación para el criterio: Laboratorio

Alternativas	Spartan 3E	Basys 3	STEMlab 125-14	Promedio
Spartan 3E	1	1	1/9	0.090909091
Basys 3	1	1	1/9	0.090909091
STEMlab 125-14	9	9	1	0.818181818

3.1.5.5. Look-Up Tables (LUT)

La evaluación del criterio LUT se realiza de forma proporcional al número de LUT que posee cada tarjeta. En este sentido, el dispositivo Basys 3 obtiene la mayor puntuación, seguido de la tarjeta STEMLab y, finalmente, la Spartan 3E. La asignación específica de cada prioridad se puede observar en la Tabla 3.7 a continuación.

Tabla 3.7: Matriz de comparación para el criterio: LUTs

Alternativas	Spartan 3E	Basys 3	STEMlab 125-14	Promedio
Spartan 3E	1	1/3	1/2	0.153660349
Basys 3	3	1	5	0.640468227
STEMlab 125-14	2	1/5	1	0.205871423

La Tabla 3.7 muestra una prioridad para la tarjeta Basys 3, ya que posee el mayor número de LUT.

3.1.5.6. Flip Flop

Al igual que en el caso de las LUT, la tarjeta Basys 3 destaca por poseer la mayor cantidad de Flip Flops, lo que le otorga una clara ventaja sobre los demás dispositivos. No obstante, cabe mencionar que la prioridad de este criterio en el diseño no es la más alta, razón por la cual su valor máximo en la escala de Saaty se limita a 5, como se puede apreciar en la Tabla 3.8.

Tabla 3.8: Matriz de comparación para el criterio: Flip Flop

Alternativas	Spartan 3E	Basys 3	STEMlab 125-14	Promedio
Spartan 3E	1	1/5	1/4	0.096419437
Basys 3	5	1	3	0.619352089
STEMlab 125-14	4	1/3	1	0.284228474

3.1.5.7. Entornos de desarrollo (IDE)

La evaluación del entorno de desarrollo se centró en la comparación de los IDEs y sus versiones utilizadas. Dado que la tarjeta Basys 3 puede ser programada con la versión más actualizada de Vivado, se le asigna una valoración de 7, indicando la mayor prioridad. Sin embargo, la tarjeta STEMlab también puede ser programada con Vivado, aunque con la versión 2020, por lo que recibe una valoración de 5 según la escala de Saaty. La tarjeta Spartan, al poder ser programada solo en ISE, recibe una valoración basada en los recíprocos de las otras tarjetas. Esto se detalla en la Tabla 3.9 a continuación.

Tabla 3.9: Matriz de comparación para el criterio: IDE

Alternativas	Spartan 3E	Basys 3	STEMlab 125-14	Promedio
Spartan 3E	1	1/7	1/5	0.073772106
Basys 3	7	1	3	0.643388869
STEMlab 125-14	5	1/3	1	0.282839025

3.1.6. Matriz de prioridades de criterios y opciones

La Tabla 3.10 muestra los valores promedios resultantes de la comparación entre opciones de dispositivos para cada criterio. Estos valores forman las columnas de la matriz. Por otra parte, la fila denominada “Ponderación” contiene cada uno de los valores de ponderación obtenidos de evaluar los criterios entre pares. Para obtener los resultados de la columna de priorización, se realiza una combinación lineal de cada una de las filas, que representan los valores de cada tarjeta, con los valores de la fila de ponderación.

Tabla 3.10: Matriz de resultados del análisis AHP.

	Hardware RF	Reloj interno	DAC	Laboratorio	LUTs	Flip Flop	IDE	Priorización
Spartan 3E	0.0909	0.1013	0.3159	0.0909	0.1537	0.0964	0.0738	0.1251
Basys 3	0.0909	0.3785	0.0522	0.0909	0.6405	0.6194	0.6434	0.2307
STEMlab 125-14	0.8182	0.5202	0.6319	0.8182	0.2059	0.2842	0.2828	0.6441
Ponderación	0.4250	0.2406	0.1239	0.0713	0.0631	0.0514	0.0245	1

En consecuencia, al analizar la columna de priorización, se observa que la tarjeta STEMLab obtiene la mayor valoración. Este resultado responde a al objetivo del análisis, ya que la tarjeta STEMLab presenta la combinación adecuada de características técnicas y compatibilidad con los requisitos del trabajo de integración curricular.

3.1.7. Interpretación de los Resultados

El Proceso de Jerarquía Analítica (AHP) finaliza con la presentación de un diagrama de barras mostrado en la Figura 3.2, el cual es una representación gráfica de los resultados mostrados en la Tabla 3.10. De acuerdo al análisis AHP y el gráfico resultante, la tarjeta FPGA STEMLab 125-14 se perfila como la alternativa idónea para el desarrollo del trabajo de integración curricular. Esta selección se sustenta en su conjunto superior de características y funcionalidades adicionales para la depuración y transmisión de la señal procesada.

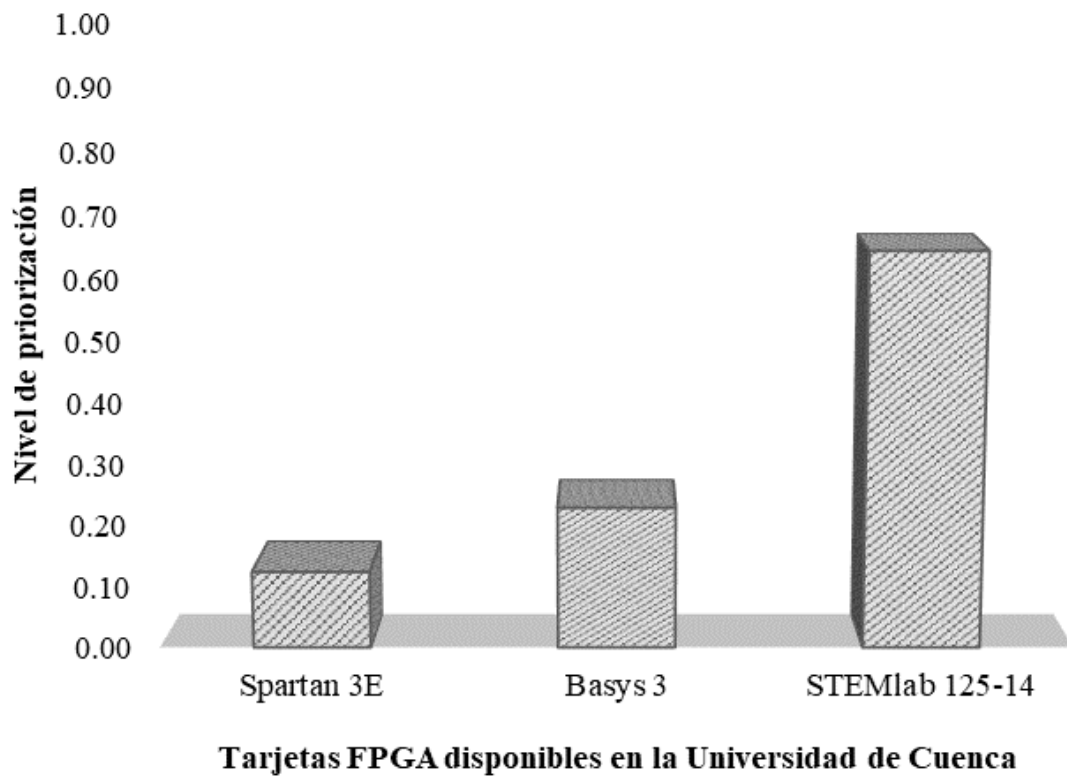


Figura 3.2: Resultado del análisis AHP.

3.2. Esquema del sistema de comunicación

La Figura 3.3 muestra la visión global del sistema de comunicación implementado en este trabajo de integración curricular. En ella, se destaca que el transmisor está conformado por un PC y la tarjeta STEMLab 125-14, mientras que el receptor está compuesto por el HackRF One y GNU Radio. Las funciones de cada uno de estos componentes se describen a continuación:

- **PC-TX:** Este computador se encarga de cargar el diseño de hardware descrito en *Hardware Description Language* (VHDL) a través de una conexión *Secure Shell* (SSH).
- **STEMLab 125-14:** Su función es ejecutar el sistema de transmisión multiportadora (*Orthogonal Frequency Division Multiplexing* (OFDM)/*Generalized Frequency Division Multiplexing* (GFDM)) descrito en VHDL. Esto se realiza mediante el comando: `redpitaya>cat /opt/redpitaya/fpga/bitstream_OFDM/GFDM.bit > /dev/xdevcfg.`

La señal de salida de este dispositivo corresponde a una señal multiportadora en pasa banda.

- **HackRF One:** Este dispositivo se encarga de convertir la señal multiportadora en pasa

banda capturada a una señal en banda base.

- **GNU Radio:** Su función consiste en demodular la señal multiportadora recibida, a partir de las muestras capturadas por el HackRF One.

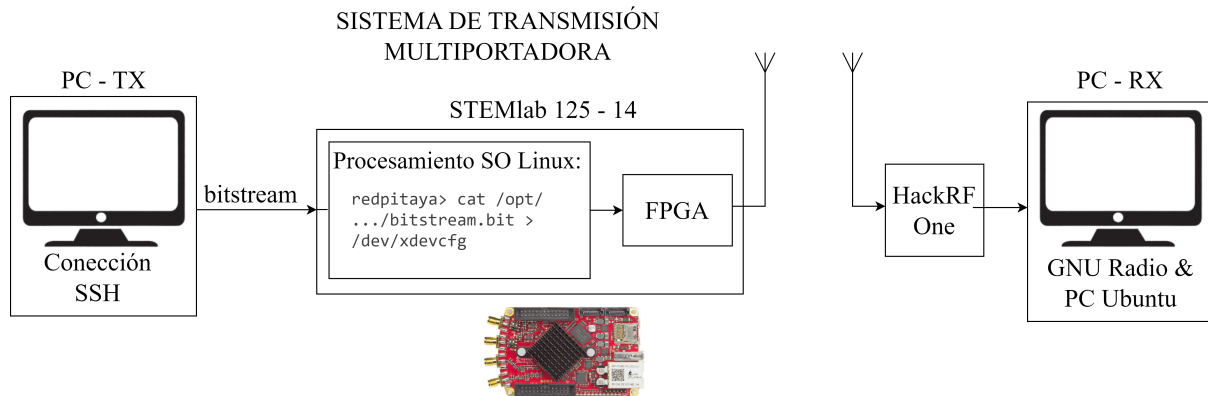


Figura 3.3: Visión global del sistema de comunicaciones implementado

3.3. Implementación del Sistema de Transmisión sobre la FPGA

Un aspecto fundamental para la implementación de un sistema de transmisión multiportadora es la capacidad de realizar operaciones aritméticas básicas con valores en punto flotante. Afortunadamente, Xilinx ha implementado estas operaciones bajo el nombre de IP-CORE Floating Point. Esta implementación permite llevar a cabo una variedad de operaciones aritméticas con valores en punto flotante conforme al estándar IEEE-754 dentro de una *Field Programmable Gate Array* (FPGA). Cada operación se especifica al generar el núcleo mediante una interfaz común. Los IP-CORE Floating Point V.7.1 pueden generarse utilizando Vivado Design Suite 2020, y son los utilizados en el presente trabajo de integración curricular.

La Figura 3.4 ilustra las diferentes entradas y salidas del IP Core Floating Point. En ella se observa que el IP Core requiere una señal de reloj, lo que implica que el resultado de cada operación se obtiene en cada ciclo de reloj. El número de ciclos de reloj necesarios para obtener el resultado de la operación se especifica durante la preconfiguración del IP Core y se denota como N .

Cada valor flotante incluye señales de control denominadas *tvalid* y *tready*. En las entradas, *tvalid* en estado ALTO indica que el valor de entrada es válido para ejecutar la operación configurada en el IP Core, mientras que, en las salidas *tready* en estado ALTO indica que el IP Core está listo para proporcionar el valor de salida calculado.

Para los valores flotantes de salida, *tready* se activa cuando el valor de salida está disponi-

ble, mientras que, en las entradas *tvalid* en ALTO indica que el valor de entrada calculado es correcto. Es importante mencionar que tanto los valores de entrada como de salida son representados como vectores binarios en VHDL, cuya estructura y longitud corresponden al estándar IEEE-754.

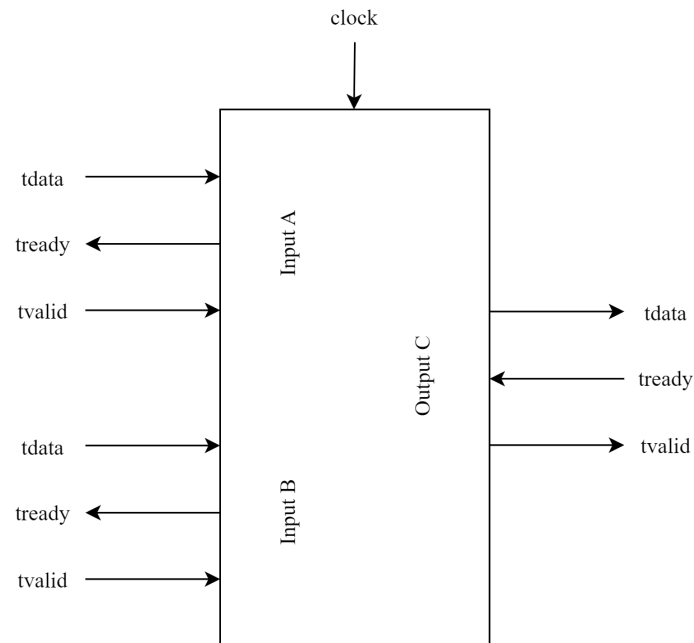


Figura 3.4: IP Core Floating Point

Uno de los modos de operación del IP Core Floating Point es el Non-Blocking. En este modo, la operación matemática se realiza en cada ciclo de reloj, produciendo un resultado independientemente de las señales de control. En consecuencia, las señales de control del valor flotante de salida del IP Core indican que dicho valor corresponde al resultado de los valores de entrada marcados como válidos ($tvalid = 1$). Es importante destacar que, para facilitar el diseño en VHDL se asume que cada valor válido se obtiene en cada ciclo de reloj, manteniendo las señales de control *tvalid* en estado ALTO durante todo el proceso de ejecución.

Después de establecer la capacidad de realizar operaciones aritméticas en una FPGA, la Figura 3.5 muestra un diagrama de bloques del diseño del sistema de transmisión implementado en el STEMLab 125-14. Cada bloque representa un componente diseñado en VHDL, donde los componentes denominados como *Muestras Multiprotadora*, *Mezclador de Cuadratura* y *Cuantización* hacen uso de los IP Core Floating Point de Xilinx. La función y el fundamento matemático de cada componente se detallarán en secciones posteriores.

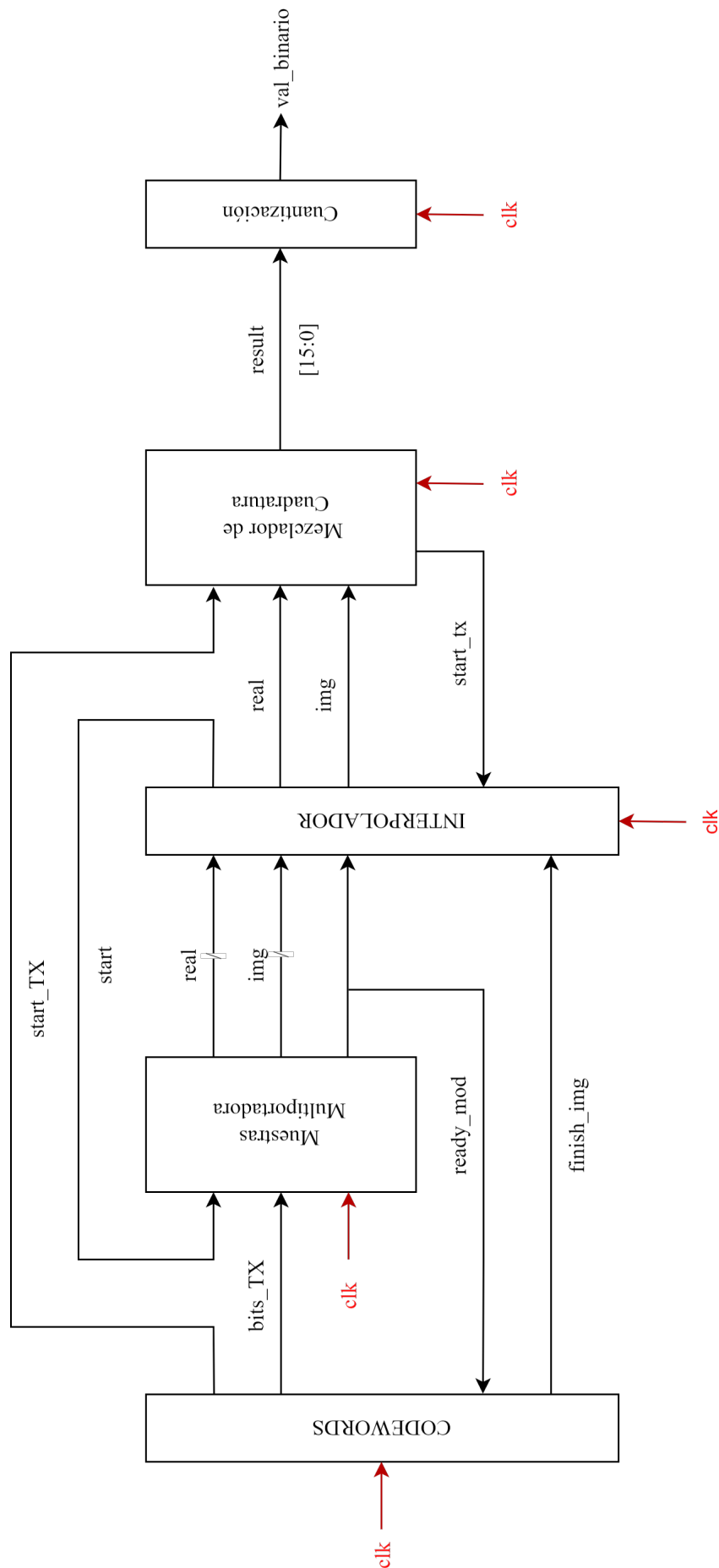


Figura 3.5: Diagrama de bloques del sistema transmisor implementado sobre la FPGA STEMLab 125-14

3.3.1. Codewords

La Figura 3.6 muestra el diagrama de bloques correspondiente a la generación de codewords binarias (`config_ofdm`) utilizados como información a enviar en cada modulación multiportadora. A continuación, se presenta una descripción de cada sub-bloque, así como las entradas y salidas que conforman este bloque generador.

- **clock:** La señal de entrada clock corresponde a la señal de reloj de la tarjeta STEM-lab 125-14, con una frecuencia interna de 125 MHz. Esta entrada funciona como un reloj interno para controlar la salida de bits **bits_tx** y aumentar el contador de la señal intermedia **cont_sal**. Esta señal de reloj tiene valores que alternan entre 0 y 1.
- **ready_mod:** Corresponde a una señal de control que determina el tiempo que dura la solicitud realizada por el modulador para entregar una codeword utilizando la salida **bits_tx**. Esta entrada funciona como una señal de reloj, donde su tiempo en alto determina el periodo para cargar una palabra en su salida.
- **start_tx:** Es una señal binaria que determina el tiempo en alto de toda una trama OFDM. Mientras esta señal esté en alto, el bloque **config_ofdm** estará entregando codewords válidos por la salida correspondiente.
- **finish_img:** Es una salida que indica cuándo se ha subido la última codeword en la trama multiportadora. Permanece en bajo hasta que se va a transmitir la última codeword de una trama OFDM. Luego, se mantiene en alto mientras se transmite la última codeword, y tras dicha transmisión, vuelve a su valor predeterminado en bajo.

La salida `finish_img` es controlada por el bloque **control_memoria**, que coloca el valor de la salida en 1 cada 3 ciclos la señal `ready_mod`, lo que corresponde a la variable $N_{TX} = 3$.

- **bits_tx:** Es una salida de 12 bits que cambia de valor con cada ciclo de reloj de la señal de entrada `clock`. El vector será dividido en grupos de 2 bits que serán modulados utilizando 4-QAM. Sus valores se cargan mediante un proceso selector.
- **cont_sal:** Es una señal contadora de 6 bits que toma el valor de un contador que varía desde 1 hasta 63 en binario. Su cambio es controlado por el bloque **control_memoria**.
- **control_memoria:** El proceso de control de memoria gestiona las señales de salida `start_tx`, `selection_tx` y `cont_sal`. Este bloque es responsable de generar el valor correcto para la salida `cont_sal`, la cual controla el proceso de selección de la salida de la memo-

ria.

El bloque funciona con base en 2 contadores. El primer contador corresponde a un selector que es un vector de 6 bits, el cual determina la cantidad de palabras en el componente memoria, es decir, la memoria contendrá $2^{\text{contador}_1} - 1 = 2^6 - 1 = 63$ palabras. Actualmente, la memoria posee 63 palabras o *codewords* (término que se utilizará para definir los bits correspondientes a un mensaje a ser enviado). El segundo contador especifica el tiempo de espera una vez finalizada la transmisión de las 63 palabras. Su valor es guardado en un vector de 26 bits que puede generar un tiempo de espera entre transmisiones de hasta 1 segundo, sin embargo, su valor se configuró en 1_2 (1 en binario) debido a que la evaluación del sistema se realiza de forma continua y no requiere un tiempo de espera entre las transmisiones.

- **Selección de bits:** Es un proceso selector que toma como entrada un contador de 8 bits y entrega a su salida un vector de 12 bits hacia la salida *bits_tx*. El proceso se actualiza en función de las señales *cont_sal* y *clock*.

El sistema generador de codewords utiliza un algoritmo en Python para generar 60 palabras de 12 bits. Estos bits son entregados por el proceso de selección de bits a la salida *bits_tx*. Los codewords utilizados para la evaluación del sistema de transmisión se generan de forma aleatoria con el comando `join(random.choice('01') for _ in range(12))` de Python.

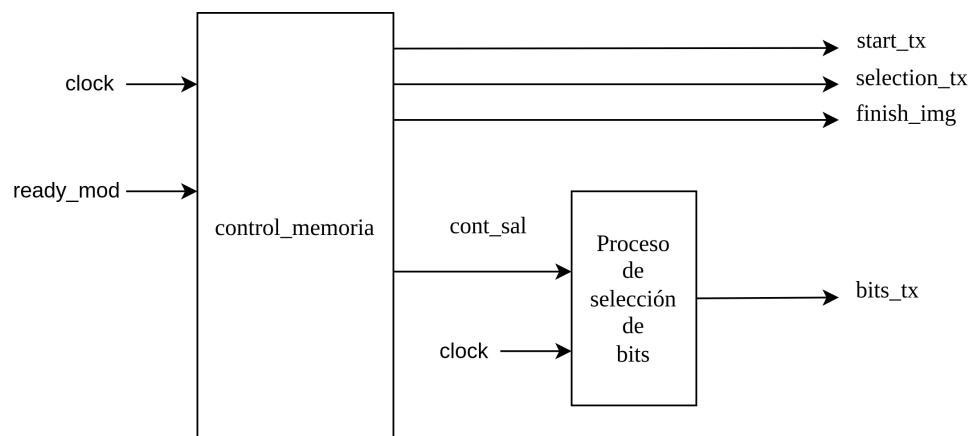


Figura 3.6: Diagrama de bloques del sistema generador de codewords.

La memoria tiene cargadas 63 palabras de 12 bits, de las cuales 60 son generadas de forma aleatoria utilizando Python, y las últimas 3 palabras corresponden a vectores de 12 bits con todos sus valores en alto, es decir, con todos sus bits iguales a 1. Estas últimas 3 tramas se utilizan para reconocer el inicio y el fin de cada transmisión multiportadora. El reconocimiento del inicio y del final de la transmisión se realiza mediante un filtro acoplado (matched filter) en

el lado del receptor.

3.3.2. Generación de muestras multiportadora (OFDM/GFDM)

En esta Sección se describe la implementación de las modulaciones OFDM y GFDM en la FPGA STEMLab 125-14. En la Sección 3.3.2.1 se presenta el modelo matemático base para la generación de muestras multiportadora. En la Sección 3.3.2.2 se detalla su implementación en la FPGA mediante diagramas de bloques. Finalmente, en las Secciones 3.3.2.3 y 3.3.2.4 se indican los parámetros de configuración necesarios para generar una modulación OFDM y GFDM, respectivamente.

3.3.2.1. Modelo Matemático de Sistemas Multiportadoras

Una onda multiportadora se define como la superposición de varias señales con una única portadora. La Ecuación (3.6) representa esta forma de onda multiportadora, donde $d_{k,i}$ corresponde a símbolos de la constelación 4-QAM, modulados utilizando un filtro formador de pulso $g_{k,m}(t)$ para la k -ésima subportadora y el m -ésimo sub-símbolo [82]. Por lo tanto, la cantidad de símbolos modulados (N) está determinada por el producto entre el número total de subportadoras (K) y el número de sub-símbolos (M) utilizados en el sistema multiportadora.

$$x_{MC}(t) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} d_{k,m} g_{k,m}(t) \quad (3.6)$$

Los filtros formadores de pulsos $g_{k,m}(t)$ deben ser diseñados de manera que ocupen bandas de frecuencia estrechas y disjuntas, evitando el solapamiento entre subportadoras. Esto se logra mediante desplazamientos de frecuencia de un pulso prototipo $g(t)$. En la práctica, este pulso prototipo es finito y causal, lo que implica que la definición de $g_{k,m}(t)$ está dada por (3.7). Esta ecuación establece que el filtro formador de pulsos se conforma mediante la repetición periódica del pulso prototipo $g(t)$ dentro de un intervalo $[0, T]$, limitado por una ventana rectangular $u_T(t) = 1$ en dicho intervalo [82].

$$g_{k,m}(t) = u_T(t) g_T(t - m\Delta T) e^{j2\pi k \Delta f (t - m\Delta T)} \quad (3.7)$$

Para realizar el procesamiento digital, es necesario discretizar la onda multiportadora definida por (3.6). Para ello, consideramos $N = MK$ muestras, haciendo que el tiempo de símbolo sea $T = N/F_s$, donde F_s es la frecuencia de muestreo. Según este enfoque, obtenemos (3.8), donde

$P = \Delta T \cdot F_s$, $Q = \frac{\Delta f}{F_s}$ y $N = \Delta f \cdot T$. Aquí, P representa el número de muestras por sub-símbolo y Q es el número de muestras por cambio de frecuencia Δf , siendo una muestra de frecuencia equivalente a F_s/N [82].

$$x_{MC}[n] = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} d_{k,m} g_T[n - mP] e^{j2\pi k \frac{nQ}{N}} e^{-j2\pi \frac{kmPQ}{N}} \quad (3.8)$$

Para facilitar la implementación del sistema de modulación multiportadora en una tarjeta FPGA, es necesario representar esta modulación en forma vectorial, como $\vec{b} = \mathbf{A} \vec{x}$, donde $\vec{b}[n_0] = x_{MC}[n_0]$. En este contexto, el tamaño del vector $\vec{b}$ debe igualar el número de muestras de la señal discretizada, lo que implica que la matriz $\mathbf{A}$ tiene dimensiones $N \times N$. Cada elemento del vector resultante del producto matriz-vector es la suma del producto de los elementos del vector $\vec{x}$ con la i -ésima fila de la matriz $\mathbf{A}$.

Según esta premisa, se define (3.9), donde se destaca que la cantidad de términos sumados es igual, lo cual establece la relación $i = m + M \cdot k$. Esta relación asegura un mapeo directo entre cada posición de la matriz $\mathbf{A}$ y el vector $\vec{x}$, correspondiente a las muestras del filtro formador de pulsos $g_{k,m}[n]$ y los símbolos QAM $d_{k,m}$, respectivamente. Este mapeo se ilustra en (3.10) y (3.11).

$$x_{MC}[n_0] = \sum_{i=0}^{N-1} A[n_0, i] x[i] = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} d_{k,m} g_{k,m}[n] \quad (3.9)$$

$$A[n, m + M \cdot k] = g_{k,m}[n] = g_T[n - mP] e^{j2\pi k \frac{nQ}{N}} e^{-j2\pi \frac{kmPQ}{N}} \quad (3.10)$$

$$x[m + M \cdot k] = d_{k,m} \quad (3.11)$$

$$\Delta_f \Delta_T = 1$$

3.3.2.2. Diseño del producto matriz-vector en VHDL

En la Sección 3.3.2.1 se enfatizó que la base de la modulación multiportadora puede ser representada mediante un producto matriz-vector. Las muestras de la señal multiportadora corresponde a valores complejos. Esto introduce cierta complejidad en la implementación sobre la FPGA, ya que los IP-CORE Floating Point de Xilinx están optimizados para operaciones con valores reales. Para abordar esta dificultad, se separan las partes real e imaginaria de (3.9), como se muestra en (3.12), donde $A[n_0, i] = A_R[n_0, i] + jA_I[n_0, i]$ y $x[i] = x_R[i] + jx_I[i]$ siendo

$i = m + M \cdot k$. Cabe mencionar que, en el contexto de las modulaciones multiportadoras, n_0 representa un instante discreto de la señal multiportadora, mientras que m y k representan el íesimo subsímbolo y la íesima subportadora, respectivamente.

$$x_{MC}[n_0] = \sum_{i=0}^{N-1} A_R[n_0, i]x_R[i] - A_I[n_0, i]x_I[i] + j \sum_{i=0}^{N-1} A_I[n_0, i]x_R[i] - A_R[n_0, i]x_I[i] \quad (3.12)$$

De acuerdo a (3.12), se observa la posibilidad de ejecutar en paralelo la suma de la parte real e imaginaria en la implementación en VHDL. Esto se debe a que la combinación en serie de un componente multiplicador y un acumulador resulta en la suma representada en (3.12), donde los valores flotantes de entrada del componente multiplicador se iteran conforme a un ciclo de reloj y según lo indicado en la ecuación mencionada. Sin embargo, la ecuación indica que durante una iteración de la suma se generan dos elementos sumatorios. Para abordar esta cuestión, se propone el enfoque ilustrado en la Figura 3.7, donde la iteración de los valores complejos asociados al filtro formador de pulsos y los símbolos 4-QAM se realiza a la mitad de la frecuencia de reloj disponible en la FPGA. En cada ciclo de reloj, los valores reales e imaginarios se entrelazan de acuerdo con el valor complejo presente en ese instante. Esta estrategia reduce la cantidad de IP Cores necesarios, optimizando los recursos de la FPGA y aprovechando el procesamiento en paralelo. Es fundamental introducir un desfase de medio ciclo de reloj entre los valores complejos y los valores entrelazados reales e imaginarios (ver Figura 3.7) para evitar problemas de sincronización negativa (*Negative Time Slacks*). En la práctica, las señales en VHDL que tienen asignado un tipo de variable `std_logic_vector` requieren tiempo para procesarse entre bloques. Este retardo puede causar que las señales no estén listas en cada ciclo de reloj, provocando *Negative Time Slacks*. El desfase de medio ciclo de reloj garantiza que se eviten errores acumulativos entre las operaciones aritméticas (producto y acumulador).

La Figura 3.8 muestra el diagrama de bloques que representa la implementación de (3.12), cuyo comportamiento central se describió anteriormente. Cada bloque representa un componente implementado en VHDL, con las siguientes funciones:

- **Mapeo de bits a símbolos:** Este componente toma el codeword de entrada (*bits_Tx*) y realiza el mapeo de dos bits de cada palabra a un símbolo 4-QAM a una frecuencia de 62.5 MHz, comenzando desde los bits menos significativos. Después de N ciclos (donde N representa el número de ciclos de reloj de latencia, provocado por los componentes IP Core) de haber completado el mapeo del codeword de entrada, se coloca la señal de

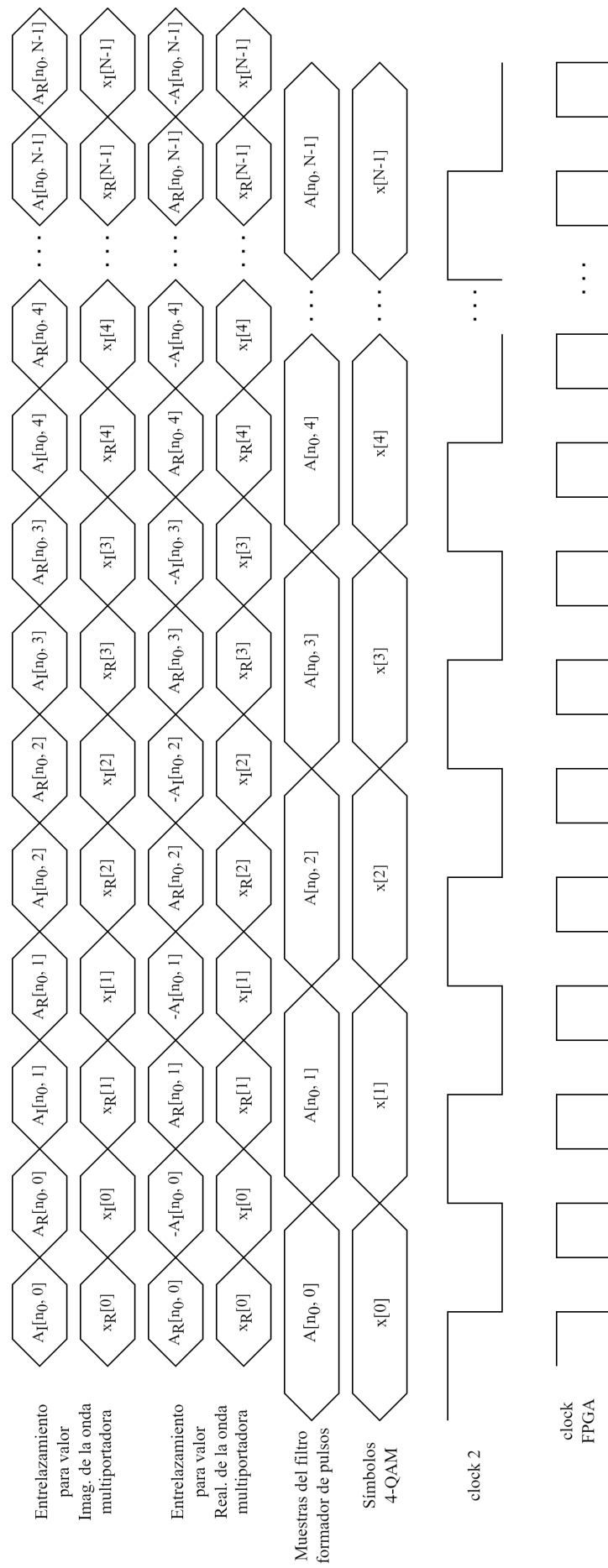


Figura 3.7: Comportamiento de la interacción entre los valores flotantes pertenecientes a las muestras del filtro formador de pulsos y los símbolos 4-QAM

salida *reset_acumulador* en BAJO para borrar los valores anteriores del acumulador de la modulación anterior.

- **Muestras del filtro:** Este componente itera cada una de las muestras del filtro formador de pulso ($A[n_0, i]$) a una frecuencia de 62.5 MHz. Después de completar la iteración de cada muestra, devuelve un valor de cero para mantener intacto el resultado del acumulador hasta la siguiente transmisión de otro codeword.
- **Entrelazador:** Este bloque toma las muestras del filtro formador de pulsos y los símbolos 4-QAM para entrelazarlos según lo especificado en (3.12).

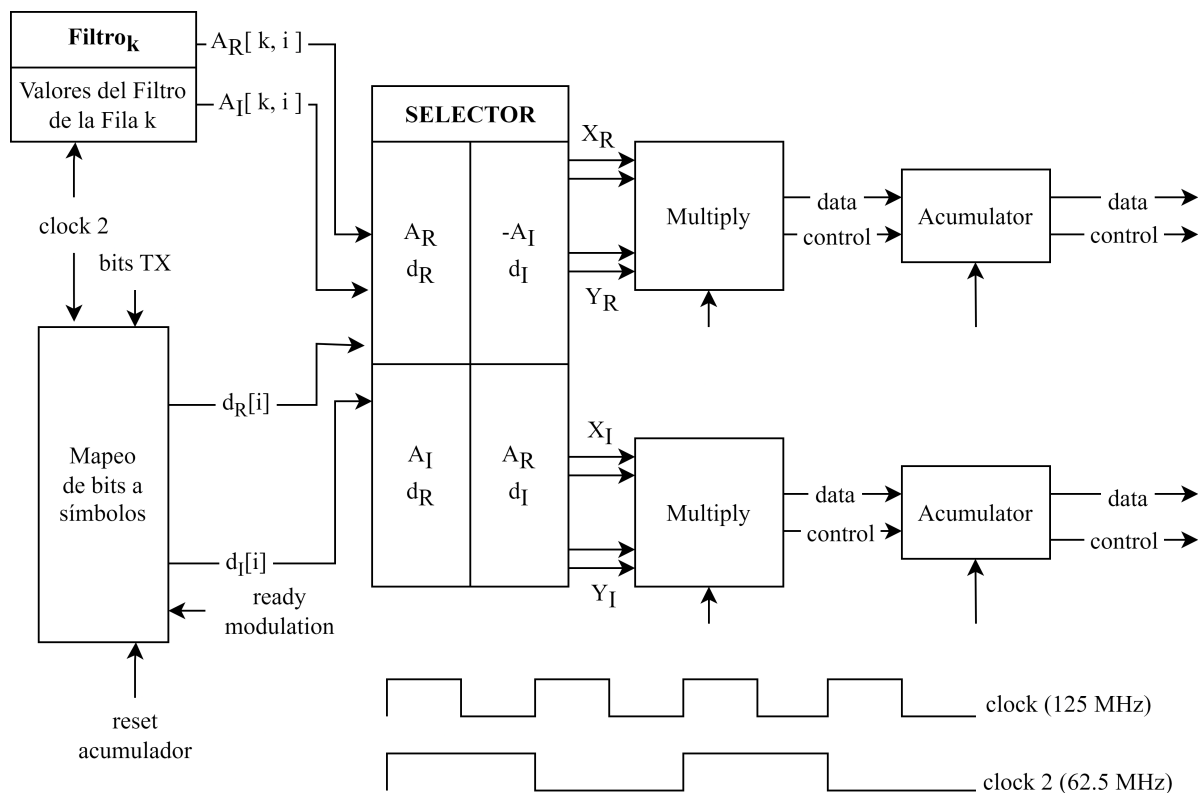


Figura 3.8: Diagrama de bloques para la obtención de una muestra de la onda multiportadora

Como se ha mencionado, el diseño anterior proporciona únicamente una muestra de la señal multiportadora. Por lo tanto, se propone implementar múltiples instancias de este diseño en paralelo sobre la FPGA, como se ilustra en la Figura 3.9. El objetivo es reducir el tiempo necesario para obtener todas las muestras de la señal multiportadora. Sin embargo, en la programación paralela, la sincronización es fundamental, ya que un componente de la Figura 3.9 podría completar su tarea antes que otros. Para abordar esta situación, se emplea una compuerta lógica AND, cuyas entradas están vinculadas a la salida denominada *ready_modulation* de la Figura 3.8. Esto indica a componentes externos que todas las muestras de la señal multiportadora están listas para ser procesadas.

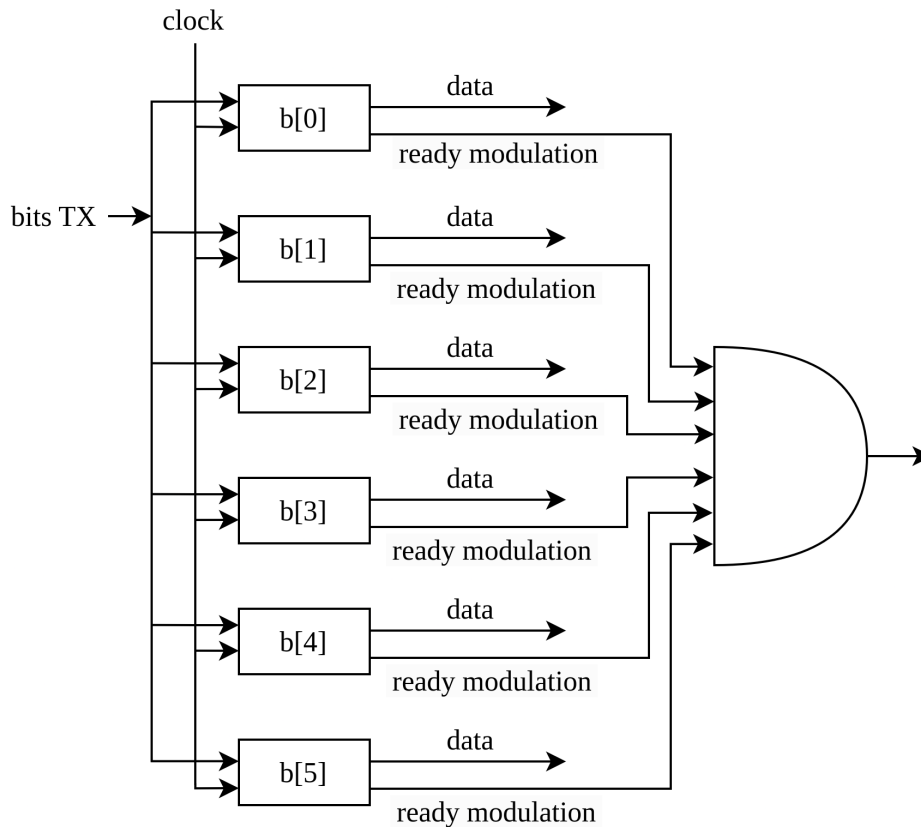


Figura 3.9: Diagrama de bloques simplificado de la implementación del producto matriz-vector

Basado en el diseño propuesto, se estima que se necesitarán aproximadamente $4N$ IP Cores Floating Point. Una característica destacada de estos núcleos es que la cantidad de LUTs que requieren está definida por su configuración previa. Para optimizar el uso de recursos en la FPGA, se ha decidido modular seis símbolos 4-QAM. Además, los valores flotantes tendrán una precisión de 16 bits, que es la menor resolución disponible en la norma IEEE-754. Esta elección no debería representar un problema, ya que los valores flotantes de 16 bits mantienen una precisión adecuada cuando están normalizados. Por lo tanto, las muestras del filtro formador de pulsos deben diseñarse de manera que el valor de cada muestra de la señal multiportadora sea menor que uno.

Es importante mencionar que cada núcleo multiplicador presenta una latencia de dos ciclos de reloj, mientras que cada núcleo acumulador tiene una latencia de ocho ciclos de reloj. Esta configuración se ha establecido para evitar problemas de sincronización negativa (*Negative Time Slacks*). Como resultado, las muestras de la señal multiportadora estarán listas aproximadamente en 22 ciclos de reloj.

Tanto los valores de las muestras del filtro formador de pulsos como los símbolos 4QAM son colocadas dentro de VHDL como vectores binarios, en función del estándar IEEE-754 de 16

bits. Para optimizar la colocación de estos valores en VHDL se diseñó un programa en Python que automatiza su colocación. El valor de la muestras del filtro formador de pulsos para OFDM y GFDM se muestran en las Secciones 3.3.2.3 y 3.3.2.4, respectivamente.

3.3.2.3. Implementación de OFDM en VHDL

En secciones anteriores se destacó que una modulación multiportadora depende del filtro formador de pulsos, el número sub-símbolos y el número de portadoras. En [82] se destaca que el número de sub-símbolos para OFDM es igual a uno ($M = 1$), por lo tanto, el número de símbolos a transmitir es igual al número de subportadoras $N = K$. Esto implica que el espaciamiento de tiempo entre sub-símbolos es igual al tiempo de símbolo de la modulación OFDM $\Delta T = T$, resultando en que el número de muestras por símbolo sea igual al número de subportadoras $P = K$. Los símbolos 4-QAM, al ser modulados únicamente por subportadoras, hacen que el número de muestras por saltos en frecuencias sea igual a uno $Q = 1$. Dado estos parámetros, el filtro formador de pulsos se define como se muestra en (3.13).

$$g_{k,0}[n] = g[n]e^{-j\frac{2\pi kn}{K}} \quad (3.13)$$

En [82] el filtro base $g[n]$ corresponde a un pulso rectangular con magnitud unitaria. Esto lleva a una definición similar a la Transformada Inversa de Fourier Discreta (IDFT, por sus siglas en inglés), como se presenta en (3.14).

$$x_{OFDM}[n] = \sum_{k=0}^{K-1} d_k e^{j2\pi k \frac{n}{K}} \quad (3.14)$$

En la Sección 3.3.2.2 se mencionó que la magnitud de las muestras de la señal multiportadora debe estar normalizada, con el fin de no perder precisión en la representación del valor flotante de 16 bits. Para cumplir con este propósito, se propone normalizar (3.14) con respecto a la máxima energía que puede contener la señal OFDM. De acuerdo con el Teorema de Parseval, la energía de la señal OFDM es igual a 2, si se modifica (3.14) para que cumpla con la definición de una IDFT. Esto último es cierto, porque tanto la parte real como imaginario de un símbolo 4-QAM puede contener un valor 1 o -1 , siendo que sin importar el símbolo que se tome su magnitud será la misma $\sqrt{2}$. La Ecuación (3.15) agrega el factor $1/K\sqrt{2}$ con respecto a (3.14), con el objeto de cumplir el concepto de la *Inverse Discrete Fourier Transform* (IDFT)

y que la energía de la señal multiportadora (3.14) sea unitaria.

$$x_{OFDM}[n] = \frac{1}{K\sqrt{2}} \sum_{k=0}^{K-1} d_k e^{j2\pi k \frac{n}{K}} \quad (3.15)$$

Para demostrar que se cumple con la normalización de la señal se utiliza el método por reducción de lo absurdo. Por tanto, se considera que $|x_{OFDM}[n_0]|^2 > 1$, mientras que $|x_{OFDM}[n]|^2 \neq 0, n \neq n_0$. Al considerar que la energía de la señal multiportadora fue normalizada se tiene el siguiente desarrollo

$$\begin{aligned} |x_{OFDM}[n_0]|^2 &> 1 \\ \sum_{n=0}^{N-1} |x_{OFDM}[n]|^2 &= 1 \\ |x_{OFDM}[n_0]|^2 + \sum_{n=0, n \neq n_0}^{N-1} |x_{OFDM}[n]|^2 &= 1 \\ |x_{OFDM}[n_0]|^2 &= 1 - \sum_{n=0, n \neq n_0}^{N-1} |x_{OFDM}[n]|^2 \\ 1 - \sum_{n=0, n \neq n_0}^{N-1} |x_{OFDM}[n]|^2 &> 1 \\ \sum_{n=0, n \neq n_0}^{N-1} |x_{OFDM}[n]|^2 &< 0 \end{aligned} \quad (3.16)$$

El resultado de (3.16) representa una contradicción, dado que $|x_{OFDM}[n]|^2 > 0$. Esto conlleva que ninguna muestra de la magnitud de la señal multiplicadora contenga un valor mayor a uno. Por ende, tanto la parte real como imaginaria de la señal multiplicadora también deben ser menores a uno.

En consecuencia, el mapeo uno a uno entre los elementos de la matriz A y el vector x entre las muestras del filtro formador de pulso y los símbolos 4-QAM, respectivamente, se detalla en las Ecuaciones 3.17 y 3.18. Según lo planteado, la matriz $\mathbf{A}$ debe tener un tamaño de 6×6 , utilizando seis subportadoras para OFDM.

$$A[n, k] = g_{k,0}[n] = \frac{1}{K\sqrt{2}} e^{j2\pi k \frac{n}{K}} \quad (3.17)$$

$$x[k] = d_{k,0} \quad (3.18)$$

3.3.2.4. Implementación de GFDM en VHDL

En GFDM, el número de muestras por tiempo de sub-símbolo es igual al número de subportadoras $P = K$, mientras que el número de muestras entre saltos de frecuencias es igual al número de sub-símbolos $Q = M$ [82]. Esto implica que el producto entre el tiempo de sub-símbolo y los saltos de frecuencias sea igual a uno $\Delta T \Delta f = 1$. Según las relaciones mostradas en la Sección 3.3.2.1, se define que el tiempo de símbolo de la modulación equivale a $T_{simb} = M\Delta T$, mientras que su frecuencia de muestreo es $F_s = K\Delta f$. Con estos parámetros, el filtro formador de pulsos se muestra en (3.19).

$$g_{k,m}[n] = g_T[n - mK]e^{-j\frac{2\pi kn}{K}} \quad (3.19)$$

En [4] se muestran diferentes filtros de pulso base para GFDM, tales como el filtro de coseno elevado (RC, por sus siglas en inglés) y el filtro de raíz de coseno elevado (RRC, por sus siglas en inglés), entre otros. En este trabajo de integración curricular se utiliza el filtro de coseno elevado, dado que es comúnmente empleado en diversos estudios [51], cuya definición en tiempo continuo se muestra en (3.20), siendo que el filtro pulso periódico se define de la forma $g_T(t) = g(t)$, para, $[-M\Delta T/2, M\Delta T/2]$.

$$g(t) = \text{sinc}(\Delta f t) \frac{\cos(\pi \Delta f \alpha t)}{1 - (2\alpha \Delta f t)^2} \quad (3.20)$$

Al igual que OFDM se debe normalizar la señal multiportadora. En GFDM durante un tiempo de sub-símbolo ΔT se modulan K símbolos 4-QAM, siendo que la energía de la señal este definido por los símbolos 4-QAM que estén presentes en ese instante. Con ello, por el teorema de Parseval la energía de la señal es igual 2 (Se sigue el mismo concepto presentado en la Sección 3.3.2.3). En base a lo anterior, se define (3.21).

$$x_{GFDM}[n] = \frac{1}{K\sqrt{2}} \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} d_{k,m} g[\langle n - mK \rangle_N] e^{j2\pi k \frac{n}{K}} \quad (3.21)$$

Una característica de GFDM es que agrega interferencia intersimbólica, siendo que se pierde ortogonalidad entre sus subportadoras. Esta interferencia provoca que parte de la energía de la señal del sub-símbolo m_0 se comparta a sus alrededores, perdiendo así la normalización de la señal durante el instante ΔT . Sin embargo, esta interferencia no es significativa, provocando que el cambio de magnitud de cada muestra de la señal GFDM sea insignificante.

En consecuencia de la relación anterior, el mapeo uno a uno entre los elementos de la matriz A y el vector x con respecto a las muestras del filtro formador de pulso y los símbolos 4-QAM, respectivamente, se detalla en (3.22) y (3.23). La matriz A debe tener un tamaño de 6×6 , siendo que los posibles valores para K y M sean 2 o 3, respectivamente. Para esta implementación, se define que el número de subportadoras a utilizar es $K = 2$, mientras que el número de símbolos es $M = 3$. Además, el valor de α del filtro formador de pulsos se definirá con un valor de 0.5.

$$A[n, k] = g_{k,m}[n] = \frac{1}{K\sqrt{2}} g[\langle n - mK \rangle_N] e^{j2\pi k \frac{n}{K}} \quad (3.22)$$

$$x[m + M \cdot k] = dk, m \quad (3.23)$$

3.3.3. Interpolador

3.3.3.1. Descripción Matemática

Para convertir una señal de banda base a pasabanda en tiempo continuo, se requiere una portadora cuya frecuencia sea mucho mayor que el ancho de banda de la señal en banda base. En el dominio discreto, esto implica que la señal de banda base debe ser muestreada, como mínimo, a la frecuencia de muestreo de la señal portadora. Sin embargo, este método aumenta el tamaño de la matriz A en función de la frecuencia de la portadora, agotando los recursos de la FPGA. Para solucionar este problema, se optó por usar un interpolador simple, que consiste en mantener una muestra de la señal de banda base durante L muestras. Este método es fácil de implementar en una FPGA, ya que no requiere diseñar una arquitectura compleja para determinar las $L - 1$ muestras adicionales a partir de más de una muestra original de la señal de banda base.

El diseño de un interpolador consta de dos fases. La primera fase consiste en expandir la frecuencia de muestreo de la señal multiportadora de la forma $F_{s1} = L \cdot F_s$, donde $x_1[n] = x_{MC}[n/L]$, $n = 0, \pm L, \pm 2L, \dots$. Esta operación crea múltiples copias de la señal de banda base en el dominio de la frecuencia discreta. La segunda fase implica aplicar un filtro paso bajo con una frecuencia de corte de π/L . Por lo tanto, la eficiencia de un interpolador está directamente relacionada con el filtro paso bajo. La señal resultante tras aplicar el interpolador simple propuesto se muestra en (3.24).

$$x'_{MC}[n] = \sum_{k_1=0}^{L-1} \sum_{k_2=0}^N x_{MC}[k] \delta[n - k_1 - k_2L] \quad (3.24)$$

Al aplicar una Transformada Discreta de Fourier, se obtiene (3.25), que muestra el efecto de aumentar la frecuencia de muestreo de la señal multiportadora y la aplicación del filtro. La respuesta al impulso del filtro paso bajo se muestra en la Figura 3.10, destacando su ineficiencia, ya que modifica la magnitud del espectro de la señal en banda base.

$$\begin{aligned}
 X'_{MC}(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} \left(\sum_{k_1=0}^{L-1} \sum_{k=0}^N x_{MC}[k] \delta[n - k_1 - kL] \right) e^{-j\omega n} \\
 X'_{MC}(e^{j\omega}) &= \sum_{k_1=0}^{L-1} \sum_{k=0}^N \sum_{n=-\infty}^{\infty} x_{MC}[k] e^{-j\omega(k_1 - kL)} \delta[n - k_1 - kL] \\
 X'_{MC}(e^{j\omega}) &= \sum_{k_1=0}^{L-1} \sum_{k=0}^N x_{MC}[k] e^{-j\omega(k_1 - kL)} \sum_{n=-\infty}^{\infty} \delta[n - k_1 - kL] \\
 X'_{MC}(e^{j\omega}) &= \sum_{k_1=0}^{L-1} e^{-j\omega k_1} \sum_{k=0}^N x_{MC}[k] e^{-j\omega kL} \\
 X'_{MC}(e^{j\omega}) &= \sum_{k_1=0}^{L-1} e^{-j\omega k_1} X(e^{j\omega L}) \\
 X'_{MC}(e^{j\omega}) &= X_{MC}(e^{j\omega L}) \sum_{k_1=0}^{L-1} e^{-j\omega k_1} \\
 X'_{MC}(e^{j\omega}) &= X_{MC}(e^{j\omega L}) \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} \quad (3.25)
 \end{aligned}$$

En la Ecuación (3.26) se observa la magnitud del filtro aplicado, cuya frecuencia de corte es $2\pi/L$ (primer punto en el que la magnitud del filtro es cero). Habitualmente, para corregir estas deficiencias, se debe aplicar un filtro pasa-bajo externo a la FPGA para limitar el ancho de banda. Sin embargo, en este trabajo de integración curricular, el ancho de banda utilizado no está limitado, por lo que no es necesario implementar el filtro pasa-bajo mencionado. Como consecuencia, el receptor debe incluir la función adicional de eliminar el efecto del filtro reflejado en (3.25) para demodular correctamente los símbolos 4-QAM.

$$|X'_{MC}(e^{j\omega})| = |X(e^{j\omega L})| \cdot \sqrt{\frac{1 - \cos(\omega L)}{1 - \cos(\omega)}} \quad (3.26)$$

3.3.3.2. Diseño en VHDL

La función principal del componente interpolador, mostrado en la Figura 3.5, es convertir las muestras de la señal multiportadora de paralelo a serial. Esto permite aplicar el método de interpolación mencionado en la Sección 3.3.3.1, de manera que la frecuencia de muestreo expandida de la señal de banda base sea igual a la frecuencia de reloj del STEMLab 125-14,

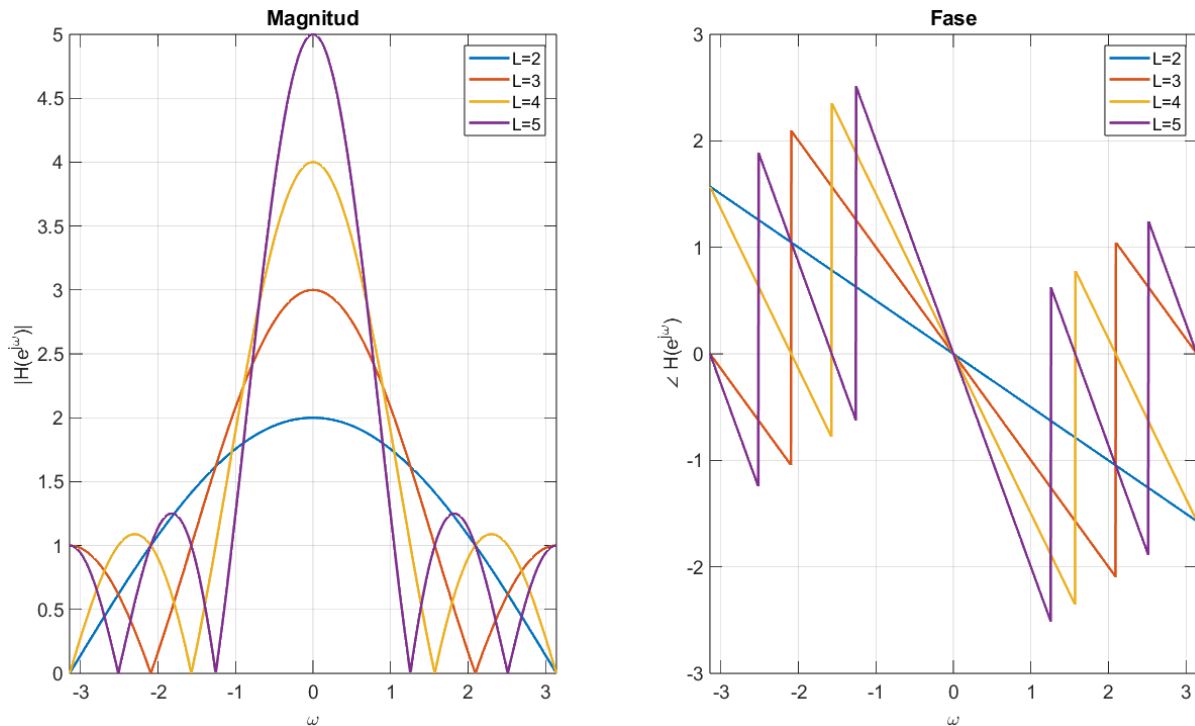


Figura 3.10: Respuesta al Impulso del filtro pasa bajo en el dominio de la frecuencia

es decir, $L \cdot F_s = 125 \text{ MHz}$.

Otra función del componente interpolador es controlar la generación de muestras de la señal multiportadora. De esta forma, en paralelo, se genera la señal interpolada mientras se generan las siguientes muestras de la señal multiportadora. Este diseño asegura que la generación de la señal interpolada no contenga pausas, lo que permite que la transmisión se realice de manera efectiva.

La Figura 3.11 muestra el diseño del componente interpolador, el cual está conformado por los siguientes componentes y fue diseñado en VHDL.

- Control:** Este componente es el encargado de generar la señal interpolada a partir del resto de componentes mostrados en la Figura 3.11. Su funcionamiento se compone de tres fases. La primera fase consiste en aguardar en un estado de espera hasta que la entrada *startTx* contenga un valor ALTO. La segunda fase genera un pulso de sincronización, tanto en la parte real como imaginaria de la señal resultante, para ayudar al receptor a identificar cuándo empieza una trama OFDM o GFDM. La duración del pulso de sincronización es de aproximadamente un tiempo de símbolo más el tiempo de muestro de la señal multiportadora (T_s), de modo que $x_{MC}[n] = 0.9 + 0.9j$, $[0, T_{Simb} + T_s]$, se coloca este valor para facilitar la sincronización en el receptor. La tercera fase consiste en interpolar la señal multiportadora y agregar el prefijo cíclico, el cual es el 33 %

del tiempo de símbolo (equivalente a dos muestras de la señal multiportadora). Dentro de esta fase, cuando se determina que la señal de banda base ya fue interpolada, se verifica el estado de la entrada *finish_img*. Si esta se encuentra en ALTO, el componente regresa a su estado inicial; sin embargo, si está en BAJO, se activa la salida *band_alm*, indicando al bloque *Almacenamiento* que guarde las muestras de la siguiente señal multiportadora a transmitir. Cabe mencionar que la salida denominada *selector* es la encargada de especificar el valor de salida del interpolador a partir del componente denominado *Salida*.

Internamente, el componente *Control* cuenta con un contador binario que es el responsable de expandir la frecuencia de muestreo de la señal multiportadora. Esta expansión de frecuencia se encuentra definida por (3.27), siendo n_c el número de bits del contador.

$$2^{n_c} F_s = 125 \text{ MHz} \quad (3.27)$$

- **Almacenamiento:** Este bloque es responsable de que la señal interpolada no contenga pausas de procesamiento, dado que se encarga de almacenar las muestras de la señal multiportadora. Una vez almacenadas dichas muestras, se comienza con las siguientes muestras de la señal multiportadora tras colocar la salida *start_mod* en ALTO. Este comportamiento se da siempre y cuando las entradas *band_alm* y *ready_mod* estén en ALTO.
- **Salida:** Este componente se encarga de seleccionar el valor de salida en función de su entrada *selector*. Los valores de salida de este componente se pueden clasificar en tres tipos: muestras de la señal en banda base, pulso de sincronización y un valor nulo. Este último se utiliza cuando el componente *Codewords* (Sección 3.3.1) no requiere ejecutar alguna transmisión. En la Figura 3.11, se destaca que este componente tiene como entrada la cuarta muestra de la señal multiportadora. Esto se debe a que el almacenamiento de las muestras de la señal multiportadora requiere más de un ciclo de reloj; por tanto, para no pausar la generación de la señal interpolada, se toman directamente las salidas del componente *Generación de Muestras Multiportadora*, que corresponde a la cuarta muestra de la señal en banda base. De esta forma, se evitan los Negative Time Slack.

La frecuencia de muestreo de la señal multiportadora depende directamente del número de bits del contador del interpolador. Para determinar el número de bits del contador, se define

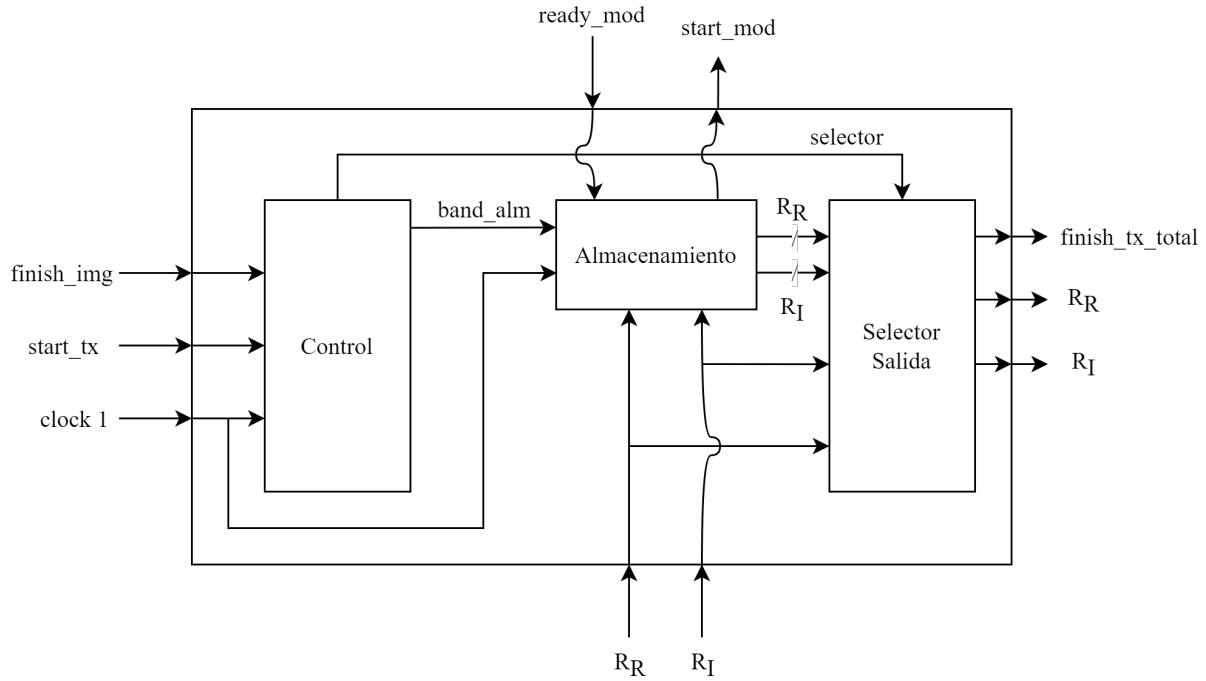


Figura 3.11: Diagrama de Bloques del componente Interpolador

(3.28), donde F'_s es la frecuencia de muestreo deseada de la señal multiportadora. En este trabajo de integración curricular, se pretende que el ancho de banda de las señales multiportadora sea de 1 MHz, por lo que se necesitan 7 bits para el contador del componente interpolador. Esto conlleva que el ancho de banda real de las señales multiportadora sea 984.375 kHz.

$$n_c = \text{Round} \left(\log_2 \left(\frac{125 \text{ MHz}}{F'_s} \right) \right) \quad (3.28)$$

$$n_c = \text{Round} \left(\log_2 \left(\frac{125 \text{ MHz}}{1 \text{ MHz}} \right) \right) = 7$$

Tras definir la frecuencia de muestreo de la señal multiportadora, se pueden obtener el tiempo de símbolo de la señal multiportadora OFDM, así como los saltos de frecuencia y tiempo de la señal multiportadora GFDM. Los parámetros de cada señal multiportadora se muestran en la Tabla 3.11.

En base al funcionamiento especificado y a los parámetros propuestos en la Tabla 3.11, la salida del componente **Interpolador** en tiempo discreto corresponde a la Ecuación 3.29, donde $(0.9 + 0.9j)\mu_R[n]$ corresponde al pulso de sincronización tal que $\mu_R[n] = 1$ $[0, 7 \cdot 2^{n_c}]$ y $x_{MC_{CP}}[n]$ corresponde a la señal multiportadora interpolada con prefijo cíclico.

$$x_{MC_I}[n] = (0.9 + 0.9j)\mu_R[n] + \sum_{b=0}^{N_{TX}-1} x_{MC_{CP}}[n - 15 \cdot 2^{n_c} b] \quad (3.29)$$

Tabla 3.11: Parámetros de señales multiportadoras

Parámetros	OFDM	GFDM
Ancho de Banda	976.56 KHz	976.56 KHz
Tiempo de Símbolo	6.095 μs	6.095 μs
Tiempo de Sub-símbolo	6.095 μs	2.032 μs
Salto de Frecuencia	164.063 KHz	492.188 KHz
Número de Subportadoras	6	2
Número de Sub-símbolos	1	3

3.3.4. Mezclador de Cuadratura

Para poder transmitir la señal multiportadora es necesario un mezclador de cuadratura, dado que esta es una señal con valores complejos. La señal resultante del mezclador de cuadratura en tiempo discreto es el que se muestra en (3.30), siendo F_p la frecuencia de la portadora y F_{s1} la frecuencia de muestreo, donde para esta implementación es igual a la frecuencia de reloj del STEMLab 125-14 (125 MHz).

$$x_o[n] = \text{Real}\{x_{MC_I}[n]\} \cos\left(2\pi \frac{F_p}{F_{s1}} n\right) - \text{Imag}\{x_{MC_I}[n]\} \sin\left(2\pi \frac{F_p}{F_{s1}} n\right) \quad (3.30)$$

Con base en (3.30) se pretende emular el funcionamiento de los *Software Defined Radio* (SDR) disponibles en el mercado, tal como el HackRF One. Para ello, se diseñó osciladores discretos en VHDL de fase y cuadratura, de modo que por cada ciclo de reloj se obtenga una muestra de las portadoras. Para llevar la implementación de (3.30) es necesario que las muestras de las portadoras correspondan a valores flotantes de 16 bits, siendo que el producto entre componentes de cuadratura y fase este dado por IP Cores Floating Point.

Dentro del diseño del oscilador discreto las muestras de las portadoras cambian en función de un contador. Para facilitar el diseño de implementación se impone que el número de muestras de la portadora es igual a la capacidad del contador. Con ello, se define (3.31), donde n_p es el número de bits del contador del oscilador. Para que sea congruente con el Teorema de Nyquist $n_p > 0$.

$$\frac{F_p}{F_{s1}} = \frac{1}{2^{n_p}} \quad (3.31)$$

Con base en (3.31) se establece que la máxima frecuencia de la portadora es igual a 62.5 MHz.

Sin embargo, el STEMLab 125-14 cuenta con un filtro paso bajo con frecuencia de corte de 50 MHz [11]. Ante ello, se define que el número de bits del contador sea $n_p = 2$, provocando que la frecuencia de la portadora sea igual a 31.25 MHz , siendo que por cada componente de fase y cuadratura se contenga cuatro muestras flotantes.

El diseño final del componente *Mezclador de Cuadratura* se muestra en la Figura 3.12, donde se destaca que tiene un diseño similar a lo mostrado en la Figura 2.9. Sin embargo, en el diseño propuesto cuenta con un componente denominado *Sincronización*. Este componente tiene el objeto de sincronizar las muestras entre la señal interpolada y la portadora, de modo que se obtenga el resultado esperado conforme a lo estipulado en (3.30). Para conseguir esto, se coloca un retardo en la salida *start_tx* definida en la Sección 3.3.1, de modo que se controle la salida de la señal interpolada. Ante esta implementación no es necesario una sincronización entre el componente *Codewords* y *Mezclador de Cuadratura*.

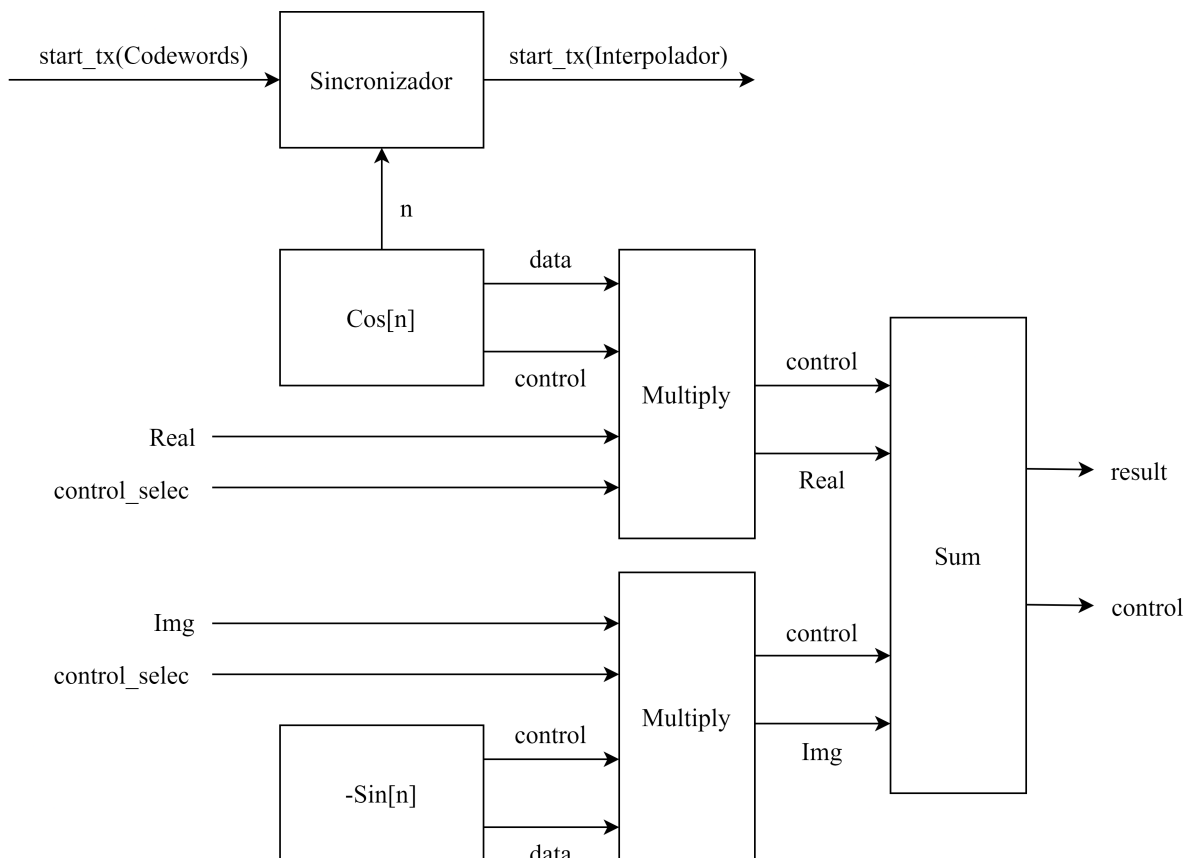


Figura 3.12: Diagrama de Bloques del componente Mezclador de Cuadratura

3.3.5. Cuantización uniforme

3.3.5.1. Descripción Matemática

El proceso de cuantización uniforme, empleado en el presente trabajo de integración curricular, se describe matemáticamente mediante dos etapas:

1. **Adición de offset:** En la primera etapa, se añade un valor de offset v_{min} a la señal discreta original v_a para ajustar su rango al intervalo que el convertidor de valor flotante a digital puede representar. La expresión matemática es:

$$V_{offset} = v_a - v_{min}$$

2. **Conversión a valor binario:** En la segunda etapa, la señal con offset V_{offset} se convierte a un valor binario de 14 bits mediante un proceso de cuantización uniforme. El rango de la señal discreta se divide en 2^{14} niveles digitales, cada uno representado por un código binario único de 14 bits.

El vector de entrada del bloque cuantizador se considera como una representación de una señal discreta que utiliza el estándar IEEE 754 para su formato de formato. Esto se debe a que los valores binarios utilizados en el proceso de modulación representan muestras de una señal multiportadora analógica.

La base de la expresión matemática que modela las dos transformaciones de la señal discreta (adición de offset y conversión a binario) radica en la relación fundamental entre cada valor digital resultante de la conversión y su correspondiente valor flotante. Esta relación se establece mediante la multiplicación de cada valor binario por su respectivo espaciamiento entre niveles, ajustada por un offset. Matemáticamente, se expresa como:

$$\begin{aligned} v_a &= (v_{bin} \cdot \Delta V) + v_{min} \\ v_{bin} \cdot \Delta V &= v_a - v_{min} \\ v_{bin} \cdot \frac{v_{max} - v_{min}}{2^N - 1} &= v_a - v_{min} \\ v_{bin} &= \frac{v_a - v_{min}}{v_{max} - v_{min}} \cdot 2^N - 1 \end{aligned}$$

El modelo matemático utilizado para el proceso de cuantización se presenta en (3.32) a con-

tinuación.

$$v_{bin} = \frac{v_a - v_{min}}{v_{max} - v_{min}} \cdot (2^N - 1) \quad (3.32)$$

Donde,

- v_{bin} : Representa el vector binario resultado de la conversión del valor flotante de entrada v_a a un número binario de 14 bits.
- v_a : Corresponde a una muestra de la señal modulada en formato IEEE 754.
- v_{min} : Corresponde al número más pequeño que puede ser entregado por el proceso de modulación. En este desarrollo, se considera que su valor es -1 .
- v_{max} : Representa la máxima magnitud que puede alcanzar un símbolo modulado. A través del proceso de análisis de energía utilizando el teorema de Parseval, se determinó que su valor corresponde a 1 en formato IEEE 754.
- N : Es el número de bits. La tarjeta STEMLab 125-14, posee un conversor DAC de 14 bits, por tanto, la conversión a binario también utiliza $N = 14$ bits.

3.3.5.2. Diseño en VHDL

El proceso de cuantización tiene como objetivo convertir la señal modulada discreta en una señal digital de 14 bits.

La modulación se ha construido utilizando bloques de punto flotante de 16 bits. Sin embargo, para una conversión de valores flotantes a datos digitales más precisos, se decidió incluir una etapa de conversión de valores flotantes de 16 bits a 32 bits, pasando del estándar IEEE 754 de media precisión (16 bits) a precisión simple (32 bits). Una vez definido este ajuste en los tamaños de transformación, se procedió a modelar (3.32) que resultó en el circuito cuantizador, que se visualiza en la Figura 3.13.

La Tabla 3.12 detalla las señales y bloques que forman parte de la descripción de hardware del circuito cuantizador, el cual incluye las etapas de conversión, resta, multiplicación, conversión a binario y redondeo.

Tabla 3.12: Descripción de los bloques y señales en el proceso de cuantización uniforme

Señal/Bloque	Descripción
clock	Utilizan el reloj de 125 MHz de la tarjeta STEMLab 125-14.
resultado sdr	Resultado de la modulación multiportadora tras el mezclador en cuadratura, se representa en formato IEEE 754 de 16 bits.
data_ieee	Es el resultado de la conversión del formato IEEE 754 de media precisión (16 bits) a precisión simple (32 bits).
ieee_valid	Indica el estado de la salida data_ieee. Un valor de 1 indica que data_ieee se ha cargado correctamente, mientras que un valor de 0 indica que aún no se ha procesado.
val_min	Representa el valor mínimo que puede alcanzar la señal modulada. Su valor es -1 .
val_min_tvalid	Controla la entrada val_min. Su valor es constante e igual a 1.
res_result	Almacena el resultado del bloque resta y lo conecta con el bloque multiplicador.
res_result_tvalid	Indica si el resultado res_result está disponible para su uso.
factor	Es una constante de valor flotante igual a 8191.5.
factor_tvalid	Indica el estado de la entrada factor. Su valor se establece en 1, dado que factor es una constante.
result_multiplicador	Es el resultado del bloque multiplicador y tiene un valor flotante escalado de 32 bits.
tvalid_multiplicador	Es un 1 bit que indica el estado del resultado del bloque multiplicador.
bin_sin_redond	Es el número binario sin redondear obtenido como resultado de cuantización de la modulación. Tiene una longitud de 14 bits.
t_valid_bin	Es la señal de control que determina el estado de la salida bin_sin_redond.
deci	Salida binaria de 1 bit que indica si es necesario redondear la salida digital.
binario	La señal declarada con el nombre de "binario" es una señal de 14 bits, resultado de un proceso de conversión de valores flotantes pertenecientes a la señal discreta modulada. Este vector se envía a la entrada del DAC de la tarjeta STEMLab 125-14.
my_rest, my_multiplicador	Son bloques IP CORE que implementan las operaciones de resta y multiplicación, respectivamente, utilizando el tipo de dato floating point IEEE 754 de precisión simple.
conv_bin	Recibe un vector de entrada de 32 bits (formato de precisión simple) y genera un vector de salida binario de 14 bits.
redondeo	Es un bloque que realiza la operación de redondeo, sumando el valor de la señal de entrada "deci" (señal de un bit) al valor de la entrada "bin_sin_redond" (señal de 14 bits).

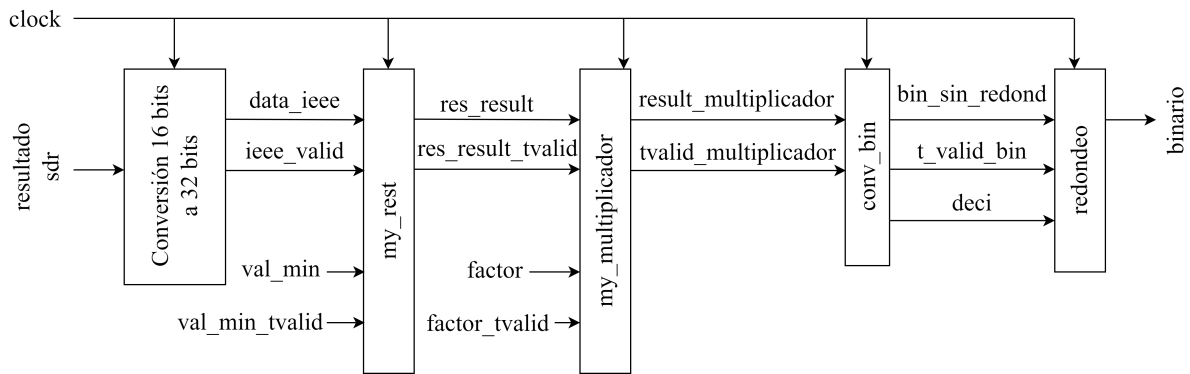


Figura 3.13: Diagrama de Bloques del componente cuantizador.

La salida del cuantizador es un vector `std_logic` de 14 bits que se conecta directamente a las entradas del convertidor DAC de la tarjeta Red Pitaya STEMLab 125-14. El control del módulo DAC se basa en el principio de funcionamiento del ejemplo **Frequency Counter**¹ desarrollado por el fabricante Red Pitaya.

El ejemplo Frequency Counter mide las frecuencias de las entradas analógicas de la tarjeta STEMLab 125-14. En este ejemplo, se utilizan tanto las entradas ADC como las DAC de la tarjeta. Sin embargo, para el desarrollo del trabajo de integración curricular, el bloque de importancia para el control de DAC es el diagrama mostrado en la Figura 3.14, donde se ilustra la conexión de la salida DAC para habilitar las salidas analógicas de la tarjeta.

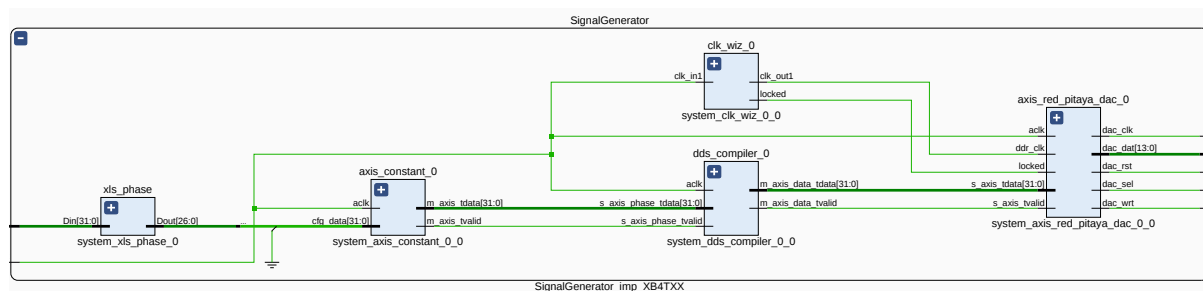


Figura 3.14: Diagrama esquemático del bloque Generador de Señales del ejemplo Frequency Counter. Fuente: [13]

La Figura 3.15 muestra la circuitería de la tarjeta STEMLab 125-14, la cual está compuesta por un DAC con entradas de 14 bits, un amplificador, un filtro paso bajo y puertos SMA de 50Ω. El modelo del DAC es un AD9767, cuya característica principal es que su interfaz digital consta de dos latches de doble buffer y lógica de control, permitiendo escribir datos en dos puertos DAC de forma independiente a través de sus entradas de escritura [83]. Con esta información, se modificó el ejemplo Frequency Counter para utilizar únicamente un puerto

¹Frequency Counter: https://redpitaya-knowledge-base.readthedocs.io/en/latest/learn_fpga/4_lesson_s/FreqCounter.html

DAC (OUT1 según la Figura 3.15). Por otra parte, la frecuencia de corte del filtro paso bajo es de 50 MHz, mientras que el amplificador asegura que las señales de salida tengan una potencia de transmisión de 10 dBm, permitiendo la generación de señales de ± 1 v [14].

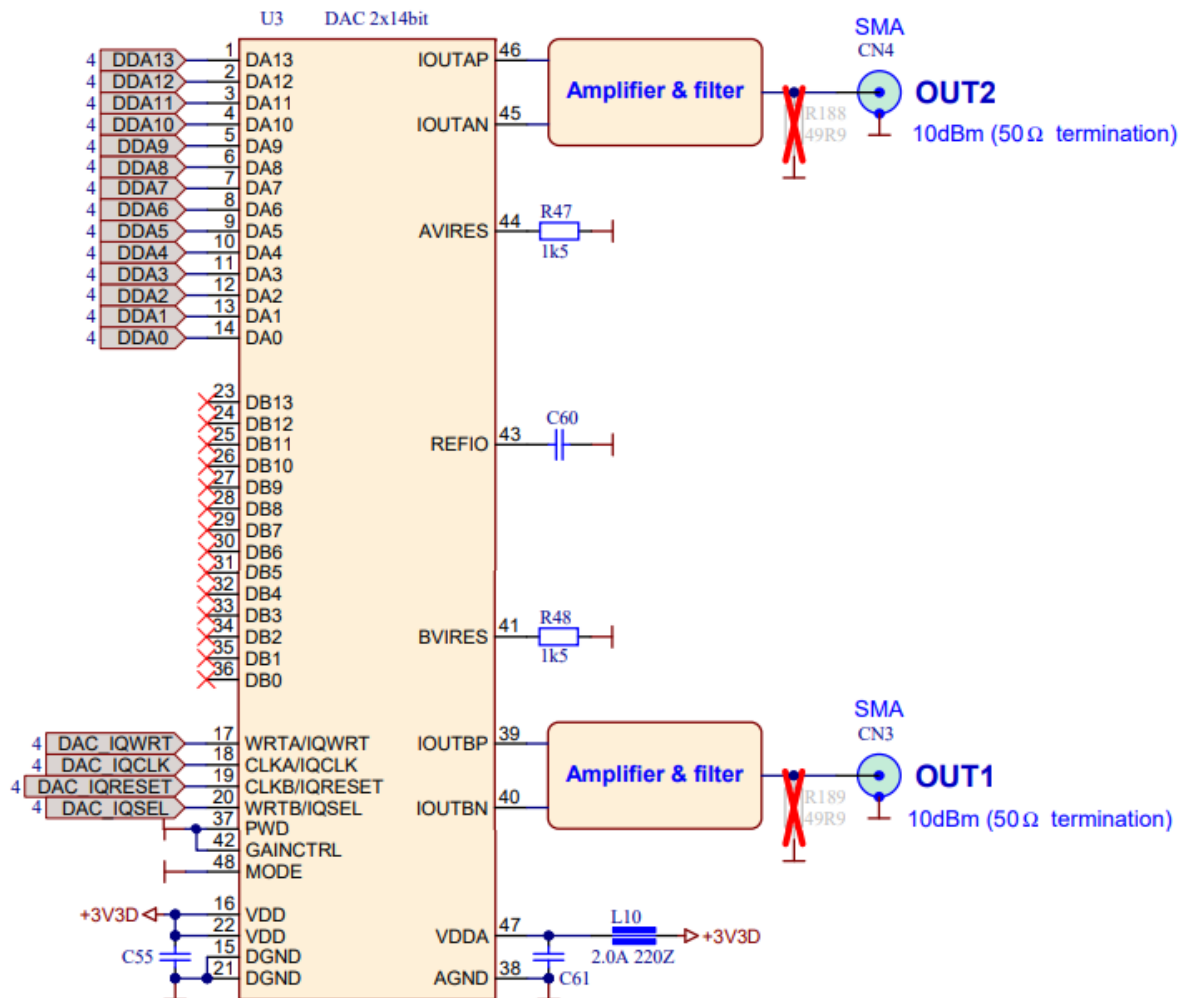


Figura 3.15: Circuitería RF en STEMLab 125-14. Fuente: [14]

3.4. Desarrollo de Receptor

La implementación del módulo receptor tiene como objetivo validar el transmisor implementado sobre la tarjeta FPGA. Para ello, se utilizaron dos componentes principales:

- **Hardware de radiofrecuencia:** Se empleó un dispositivo HackRF One, descrito en la Subsección 2.2.1 del marco teórico, titulado *HackRF One*. El dispositivo tiene la capacidad de recibir señales de radio en el rango de frecuencias de 1 MHz a 6 GHz. Dado que la frecuencia portadora de la señal modulada es de 31.25 MHz.

- **Software GNU Radio:** Esta herramienta de software libre permite interactuar desde un computador con el hardware HackRF One mediante bloques de programación en Python.

3.4.1. Análisis matemático para la implementación del receptor en GNU Radio

Uno de los principales problemas de los sistemas de recepción es la sincronización, lo que puede introducir una mayor cantidad de errores independientemente del efecto del canal. Por ello, para implementar el sistema de recepción en GNU Radio, es necesario corregir dos problemas de sincronización: el desfase entre los relojes del HackRF One y el STEMLab 125-14, y la incertidumbre en el inicio de una trama multiportadora. La solución a cada uno de estos problemas se abordará en las siguientes secciones, donde también se presentará el método de demodulación de señales multiportadoras.

3.4.1.1. Desfases entre los relojes del sistema transmisor y receptor

El dispositivo HackRF One devuelve la señal en banda base, cuya frecuencia central (f_c) y frecuencia de corte del filtro paso bajo se especifican en GNU Radio. Además, debido a los componentes internos del HackRF One, se introduce un componente DC en la señal en banda base, cuya magnitud es variable. A partir de estas características, se puede definir (3.33) como la señal recibida en GNU Radio, considerando su representación unilateral, la atenuación en el espacio libre α y un proceso estocástico Gaussiano $n_0(t)$, ya que tanto el receptor como el transmisor se mantienen fijos, eliminando problemas de desvanecimientos.

$$y(t) = \alpha \cdot (\text{Real}\{x_{MCp}(t)\} + j \cdot \text{Imag}\{x_{MCp}(t)\}) e^{j(2\pi(f_p - f_c)t + \theta)} + C_{DC} + n_0(t) \quad (3.33)$$

Dada (3.33), es necesario eliminar tanto el efecto del componente DC como el desfase dentro de GNU Radio. Para eliminar el componente DC, se optó por el uso de un filtro Notch. Este filtro no altera las subportadoras de la señal multiportadora, ya que la señal recibida cuenta con una portadora $f_p - f_c$. En la práctica, el desfase θ puede ser variable, lo que hace complejo eliminarlo completamente. Por ello, se optó por el uso de un *Phase Locked Loop* (PLL), el cual implementa un sistema de lazo cerrado para detectar el desfase de una señal de entrada. Este sistema de lazo cerrado ya está implementado en GNU Radio, donde se denomina **PLL Carrier Tracking**, obteniéndose a la salida el producto de la señal de entrada con la portadora desfasada. Con base en lo mencionado, se obtiene (3.34), donde aún se mantiene un

desfase θ' , ya que en la práctica se considera que el PLL no es capaz de eliminar completamente el desfase existente entre los relojes del sistema transmisor y receptor. Se asume que su valor para un funcionamiento correcto debe estar dentro del intervalo $[-15^\circ, 15^\circ]$. Sin embargo, en experimentos se observó que en instantes aleatorios el PLL falla significativamente, provocando que θ' se encuentre fuera del rango mencionado.

$$y_{MC}(t) = \alpha (\text{Real}\{x_{MC_{CP}}(t)\} + j \cdot \text{Imag}\{x_{MC_{CP}}(t)\}) e^{j\theta'} + n_0(t) \quad (3.34)$$

3.4.1.2. Detección de inicio de tramas multiportadora

El receptor necesita reconocer la primera muestra perteneciente a la señal multiportadora para demodularla correctamente, por lo que el transmisor agrega un pulso de sincronización junto a la trama multiportadora. Para ello, se opta por la implementación de un filtro acoplado (Matched Filter), cuyo punto de máxima similitud indica el inicio de la trama multiportadora. La respuesta al impulso del filtro acoplado se muestra en (3.35), donde $\mu_R(t)$ corresponde a un pulso rectangular, de modo que $\mu_R(T) = 1; [0, T_{simb} + T_s]$.

$$h(t) = (0.9 + 0.9j)\mu_R(t) \quad (3.35)$$

En la implementación de este trabajo de integración curricular, la señal transmitida no cuenta con símbolos piloto ni con un protocolo para corregir el efecto del PLL en caso de que falle en eliminar el desfase. Esto se debe a que la implementación de dichos mecanismos no son parte de los objetivos del presente trabajo de integración curricular, sino la implementación fundamental de las modulaciones multiportadora en una FPGA STEMLab 125-14.

La falencia del PLL provoca un aumento en el *Bit Error Rate* (BER), impidiendo obtener resultados precisos en la implementación de las modulaciones multiportadoras. Por ello, se opta por descartar aquellas recepciones en las que se producen eventos catastróficos, con el objetivo de evaluar correctamente las señales multiportadora. Para detectar los eventos catastróficos, es necesario considerar la salida del Matched Filter dada por (3.36), cuya entrada corresponde a (3.34), considerando únicamente el pulso de sincronización sin ruido Gaussiano, de modo

que $y_{MC}(t) = \alpha \cdot h(t)e^{j\theta'}$, donde α es la atenuación por el espacio libre.

$$\begin{aligned}
 y_{MF}(t) &= \int_{-\infty}^{\infty} y_{MC}^*(\tau)h(\tau-t)d\tau \\
 y_{MF}(t) &= \int_{-\infty}^{\infty} \alpha e^{-j\theta'}(0.9-0.9j)h(t-\tau)d\tau \\
 y_{MF}(t) &= 2\alpha e^{-j\theta'}(0.9)^2 \int_{-\infty}^{\infty} \mu_T(\tau)\mu_T(t-\tau)d\tau \\
 y_{MF}(t) &= \begin{cases} 1.62\alpha e^{-j\theta'}t & 0 < t < T_{simb} + T_s \\ 1.62\alpha e^{-j\theta'}(2(T_{simb} + T_s) - t) & T_{simb} + T_s < t < 2(T_{simb} + T_s) \end{cases} \quad (3.36)
 \end{aligned}$$

En la Ecuación (3.36), se observa que en el punto de máxima similitud se encuentran componentes de cuadratura y fase, cuyos valores corresponden a $1.62\alpha \cos(\theta')$ y $-1.62\alpha \sin(\theta')$, respectivamente. De esto se desprende que, ante un desfase muy pequeño, el componente de cuadratura tiende a cero. Con esta información, en la práctica se define un rango permitido para los valores de cuadratura en el punto de máxima similitud. Aquellos pulsos de sincronización que contengan un desfase mayor al permitido hacen que las muestras de la señal multiportadora no se tomen en cuenta, ya que se considera que están desfasadas. El rango mencionado se muestra en (3.37), donde PC_{max} y PI_{max} corresponden al punto de máxima similitud en cuadratura y fase capturado, respectivamente, mientras que θ'_0 corresponde al máximo desfase permitido. Para la evaluación de este trabajo de integración curricular, se consideran aquellas transmisiones que contienen un desfase máximo de 15° .

$$-tan(\theta'_0)PI_{max} < PC_{max} < tan(\theta'_0)PI_{max} \quad (3.37)$$

Para facilitar la evaluación de las modulaciones multiportadora es necesario reconocer la primera trama de la señal multiportadora. Para alcanzar este propósito se optó nuevamente por el uso de un Matched Filter, donde el punto de máxima similitud corresponda al inicio de la primera muestra de la primera trama de la señal multiportadora. La respuesta al impulso del Matched Filter corresponde al reflejo temporal de una señal multiportadora, cuyos símbolos 4-QAM es el mismo de tal forma que el mensaje no codificado sea igual a 111111111111₂. Se escogió este mensaje, ya que es muy poco probable que se transmita si se considera que los mensajes son aleatorios. La respuesta al impulso del Matched Filter se muestra en (3.38), siendo que (3.39) corresponda a una representación en tiempo continuo de (3.29).

$$h_1(t) = x_{MCs_{CP}}(5T_{simb} + Ts - t) \quad (3.38)$$

$$x_{MCs_{CP}}(t) = (0,9 + 0,9j)\mu_R(t) + \sum_{b=0}^{N_{TX}-1} x_{MCs_{CP}}\left(t - T_s - \frac{T_{simb}}{3}(3 + 4b)\right) \quad (3.39)$$

La respuesta de un Matched Filter permite reconocer un punto de máxima similitud entre la señal de entrada y el filtro acoplado. Es por ello, que se pueda aplicar la regla mostrada en (3.37), con el objeto de evitar falsos positivos en la detección de la primera trama de la señal multiportadora.

3.4.1.3. Demodulador Multiportadora

Tras identificar las muestras de la señal multiportadora (Sección 3.4.1.2), se definen las siguientes etapas para demodular la señal multiportadora:

- **Remover Prefijo Cíclico:** Se elimina la porción de la señal recibida perteneciente al prefijo cíclico, asumiendo que fue afectada por la interferencia entre símbolos.
- **Demodulación GFDM:** En GFDM existen varias formas de demodular la señal, siendo el Matched Filter (MF) y el Zero-Forcing (ZF) los más utilizados [4]. Sin embargo, para una demodulación general de señales multiportadora se opta por el método Zero Forcing, la cual consiste en eliminar el efecto de la matriz $\mathbf{A}$ (Definido en la Sección 3.3.2.1). Para lograr esto, se considera que la señal recibida puede modelarse como se muestra en (3.40), donde $\vec{w}$ corresponde a muestras de ruido Gaussiano.

$$\vec{y} = \mathbf{A}\vec{x} + \vec{w} \quad (3.40)$$

Así pues, para despejar el vector $\mathbf{x}$ de (3.40) es necesario aplicar un producto matricial $\mathbf{A}^{-1}$. De esta forma, se contiene la Expresión (3.41), siendo que la matriz $\mathbf{A}^{-1}$ pueda amplificar o atenuar el ruido. Ante esta posibilidad, al usar un demodulador Zero Forcing la tasa de error GFDM puede disminuir con respecto a OFDM.

$$\mathbf{A}^{-1}\vec{y} = \vec{x} + \mathbf{A}^{-1}\vec{w} \quad (3.41)$$

- **Máxima Verosimilitud:** Una vez que la señal haya sido demodulada, se aplica la regla de máxima verosimilitud para reducir la probabilidad de error. Esta regla consiste en

escoger aquel símbolo que se encuentra a una menor distancia euclidiana con respecto al símbolo receptado.

3.4.2. Software GNU Radio

GNU Radio es una herramienta open source que permite interactuar con el hardware RF utilizando bloques de Python (software), en este caso, con el dispositivo HackRF One. Para el desarrollo del módulo receptor se utilizó GNU Radio Companion versión 3.10.1.1, instalada en un sistema operativo Ubuntu 22.04 LTS.

El módulo receptor fue implementado utilizando los bloques predefinidos de GNU Radio, como Osmocom, PLL, Interpolating FIR Filter, y multiply. Para la conversión se emplearon los bloques Complex Conjugate, Vector to Stream, y Variable. Sin embargo, para establecer el correcto funcionamiento de los umbrales que determinan los máximos locales de cada matched filter, se utilizaron bloques embebidos de Python, específicamente el bloque de GNU Radio llamado Embedded Python Block, además de importar la librería Numpy utilizando el bloque Import. A continuación, se presenta la configuración cada bloque de GNU Radio implementado para construir el demodulador.

3.4.2.1. Bloque Osmocom source para la recepción de la señal

El bloque Osmocom, del fabricante con el mismo nombre, es una herramienta para recibir señales de radiofrecuencia utilizando el SDR HackRF One. Este bloque proporciona una interfaz intuitiva para conectar los dispositivos SDR con GNU Radio, permitiendo el procesamiento y análisis de señales de radio en tiempo real.

El bloque Osmocom Source actúa como un puente entre el hardware SDR y el entorno de programación GNU Radio. Se encarga de la configuración, sincronización, conversión y entrega de datos desde el SDR a GNU Radio. El bloque se muestra en la Figura 3.16.

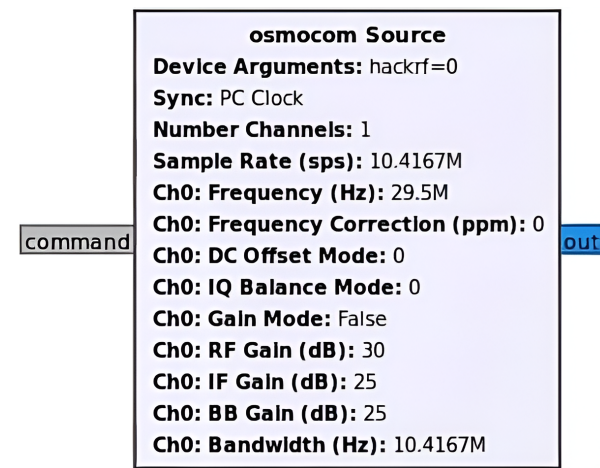


Figura 3.16: Configuración para el bloque Osmocom. Fuente: [8]

La Tabla 3.13 muestra los parámetros de configuración empleados por el bloque Osmocom Source para establecer la comunicación entre el hardware HackRF One y el software GNU Radio.

3.4.2.2. Bloque DC Blocker

El módulo bloqueador DC está diseñado para eliminar la componente DC de una señal al implementar un filtro paso alto que atenúa las frecuencias cercanas cero. La longitud de la línea de retardo (parámetro Length) determina la anchura de la ranura alrededor de la componente DC.

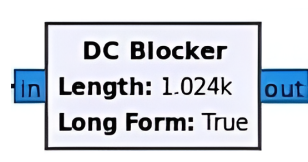


Figura 3.17: Configuración para el bloque DC Blocker. Fuente: [15].

- **Length:** El parámetro especifica la longitud del retardo en muestras utilizada por el DC Blocker, su valor se configura en 1024 muestras.
- **Long form:** El valor TRUE indica que se utiliza la versión extendida del algoritmo DC Blocker, que elimina de manera más completa el componente DC a costa de mayores recursos computacionales.

Tabla 3.13: Parámetros de configuración del bloque Osmocom Source de GNU Radio.

Parámetro	Configuración
Device Arguments	El dispositivo utilizado HackRF One se define utilizando el comando hackrf=0
Sync	El parámetro especifica la fuente de sincronización del bloque Osmocom, donde se seleccionó el reloj del PC.
Number Channel	El bloque operará en un único canal.
Sample Rate (sps)	Los datos del bloque Osmocom serán procesados a una frecuencia de muestreo igual a 10.4167 MHz.
Frequency (Hz)	La frecuencia central del canal 0 para el bloque Osmocom es igual a 29.5 MHz.
Frequency Correction (ppm)	No se aplica corrección de frecuencia en partes por millón (ppm).
DC Offset Mode	El parámetro controla el modo DC Offset para el canal y su valor se establece en 0, lo que indica que el offset DC se ajusta automáticamente.
IQ Balance Mode	El balance IQ está desactivada.
Gain Mode	Este parámetro indica si la ganancia se controla manualmente, el valor de 0 indica que la ganancia se ajusta automáticamente.
RF Gain (dB)	Es la ganancia RF en decibelios (dB), su valor se configura a 30 dB.
IF Gain (dB)	Es la ganancia IF, su valor se configura a 25 veces la intensidad de la señal original.
BB Gain (dB)	Es la ganancia de banda base y su valor se establece en 25 dB.
Bandwidth (Hz)	El ancho de banda del canal se establece en 10.4167 MHz, este valor fue calculado mediante una operación diezmado por un factor 12, con el objetivo de capturar componentes espectrales suficientes haciendo que se recepte la señal multiportadora interpolada.

3.4.2.3. Bloque PLL Carrier Tracking corrector de fase

El bloque PLL Carrier Tracking implementa un bucle de sincronización de fase (PLL) para realizar un seguimiento a la frecuencia portadora de una señal modulada. Un control de bucle de enganche de fase (PLL) es un sistema de realimentación de bucle cerrado que sincroniza su señal de salida en frecuencia y fase con una señal de entrada, de uso común en diversas aplicaciones como sistemas conectados a la red y circuitos de RF [84] [85]. En este contexto, el objetivo del PLL es corregir el error de fase entre el clock del HackRF y la señal de reloj de la señal modulada. La Figura 3.18 a continuación muestra el bloque PLL de GNU Radio.

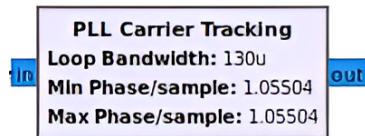


Figura 3.18: Configuración para el bloque PLL Carrier Tracking. Fuente: [16]

- **Loop Bandwidth:** Establece el ancho de banda del bucle del PLL, que determina el balance entre la velocidad de seguimiento y la estabilidad, su valor se configura en 130 μs para proporcionar una mayor estabilidad.
- **Min Phase/sample:** Determina el incremento mínimo de fase por muestra para el PLL y su valor se configura de acuerdo a la ecuación:

$$\text{Radianes por muestra} = 2 \cdot \pi \cdot \frac{31.25 \text{ MHz} - \text{central_freq}}{\text{samp_rate}} = 1.05504 \quad (3.42)$$

La Ecuación (3.42) fue obtenida de la documentación oficial del bloque PLL Carrier Tracking [16].

- **Max Phase/sample:** Determina el incremento de fase máximo por muestra para el PLL y su valor se configura de acuerdo a la Ecuación:

$$\text{Radianes por muestra} = 2 \cdot \pi \cdot \frac{31.25 \text{ MHz} - \text{central_freq}}{\text{samp_rate}} = 1.05504 \quad (3.43)$$

Para obtener un desfase cercano a cero se establece el mismo valor para bloque PLL en sus parámetros Min Phase/sample y Max Phase/sample.

3.4.2.4. Bloque de filtrado Interpolating FIR Filter

El bloque Interpolating FIR Filter es un filtro de respuesta al impulso finito (FIR) utilizado para la interpolación de la señal modulada.

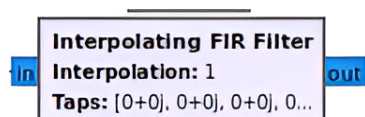


Figura 3.19: Configuración para el bloque Interpolating FIR Filter. Fuente: [15]

- **Interpolation:** Especifica el factor de interpolación, que determina la cantidad por la que se incrementa la frecuencia de muestreo.

- **Taps:** Representa los taps del filtro, que definen la respuesta al impulso del filtro FIR.

3.4.2.5. Bloque Embedded Python Block para la adición de código en Python

Es un bloque que permite implementar algoritmos para el procesamiento de señales dentro de un diagrama de flujo de GNU Radio. Al ser un bloque embebido en Python, puede utilizarse sin necesidad de crear un Módulo Fuera del Árbol (OOT). Sin embargo, su funcionamiento tiene un alcance local, es decir, solo dentro del proyecto en el que se creó, y no puede ser buscado en el apartado de componentes para usarse en otros proyectos.

3.4.2.6. Bloques de conversión

Complex Conjugate: El bloque calcula el complejo conjugado de la señal de entrada, es decir cambia de signo a la parte compleja y deja la parte real sin cambios.

Vector to Stream: El bloque convierte un vector de datos en un flujo de muestras.

Float to Complex: Convierte un conjunto de números de coma flotante en un flujo de números complejos, donde la parte real del número complejo es igual al valor de entrada flotante y la parte imaginaria es cero.

3.4.2.7. Bloques de visualización de resultados

GUI QT Time Sink: Es un bloque que utiliza el framework Qt para visualizar las gráficas en tiempo real formas de onda de señales en el dominio del tiempo.

3.4.2.8. Bloques de procesamiento de resultados

File Sink: Se utiliza para guardar la salida de un diagrama de flujo de GNU Radio para su posterior análisis. El bloque acepta un flujo de muestras como entrada y las escribe en el archivo .dat especificado en formato binario, el cual puede ser cargado en software como C, Python y MATLAB, su diagrama de bloques se puede observar en la Figura 3.20 a continuación.



Figura 3.20: Configuración para el bloque File Sink. Fuente: [17]

File Source: Es una fuente de datos pregrabados que lee un flujo de muestras de un archivo .dat especificado en el disco. Su configuración se establece de la siguiente manera: no realiza repeticiones, no añade ninguna etiqueta al flujo de datos de salida, la lectura comienza a 3.5 megabytes (MB) dentro del archivo, y el parámetro $Length = 0$ indica que el bloque leerá todo el archivo, su bloque se presenta en la Figura 3.21.

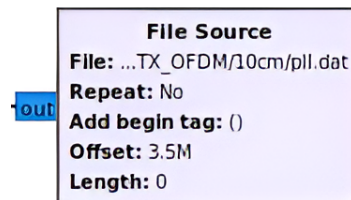


Figura 3.21: Configuración para el bloque File Source. Fuente: [18]

3.4.3. Implementación del sistema de recepción multiportadora en GNU Radio

3.4.3.1. Receptor RF

La Figura 3.22 muestra la implementación del receptor RF implementado en GNU Radio. La función principal de este receptor es disminuir el desfase existente entre los relojes del STEMLab 125-14 y el HackRF One a partir del uso de un PLL.

Sobre la Figura 3.22 se visualiza que la señal de entrada del receptor corresponde a la salida del bloque Osmocom, cuya frecuencia central es igual 29.5MHz . Esto conlleva que la señal de salida de este bloque corresponda a una señal multiportadora desfasada con portadora de 1.75MHz , ya que la frecuencia portadora original de la señal multiportadora es igual a 31.25MHz . Esta configuración permite que ninguna subportadora sea afectada por el filtro Notch (DC Bloquer, según la Figura 3.22), dado que estas se encuentran centradas en una frecuencia de 1.75MHz . El bloque PLL Carrier Tracking se encarga de pasar la señal pasabanda a banda base y disminuir el desfase existente. Sin embargo, en experimentos se observó que la señal resultante del bloque PLL Carrier Tracking tiende a un desfase de 90° , por lo que para corregirlo se optó por el producto de una constante compleja $-1j$. Al final de todo este proceso la señal resultante multiportadora es almacenada en un archivo binario para posterior aplicación del Demodulador Multiportadora.

La Figura 3.22 muestra la implementación del receptor RF en GNU Radio. La función principal de este receptor es reducir el desfase existente entre los relojes del STEMLab 125-14 y el HackRF One utilizando un PLL.

En la Figura 3.22 se observa que la señal de entrada del receptor proviene del bloque Osmocom, cuya frecuencia central es de 29.5 MHz. Esto implica que la señal de salida de este bloque es una señal multiportadora desfasada con una portadora de 1.75 MHz, dado que la frecuencia portadora original de la señal multiportadora es de 31.25 MHz. Esta configuración asegura que ninguna subportadora sea afectada por el filtro Notch (DC Blocker, según la Figura 3.22), ya que estas están centradas en una frecuencia de 1.75 MHz. El bloque PLL Carrier Tracking se encarga de convertir la señal de paso banda a banda base y de reducir el desfase existente. Sin embargo, en experimentos se observó que se mantiene la forma de onda de la señal multiportadora recibida cuando a la salida del bloque PLL Carrier Tracking tiende a un desfase de 90° entre la parte real e imaginaria de la señal recibida. Así pues, para corregirlo se optó por multiplicar la señal por una constante compleja $-1j$. Al final de este proceso, la señal multiportadora resultante se almacena en un archivo binario para su posterior aplicación en el Demodulador Multiportadora.

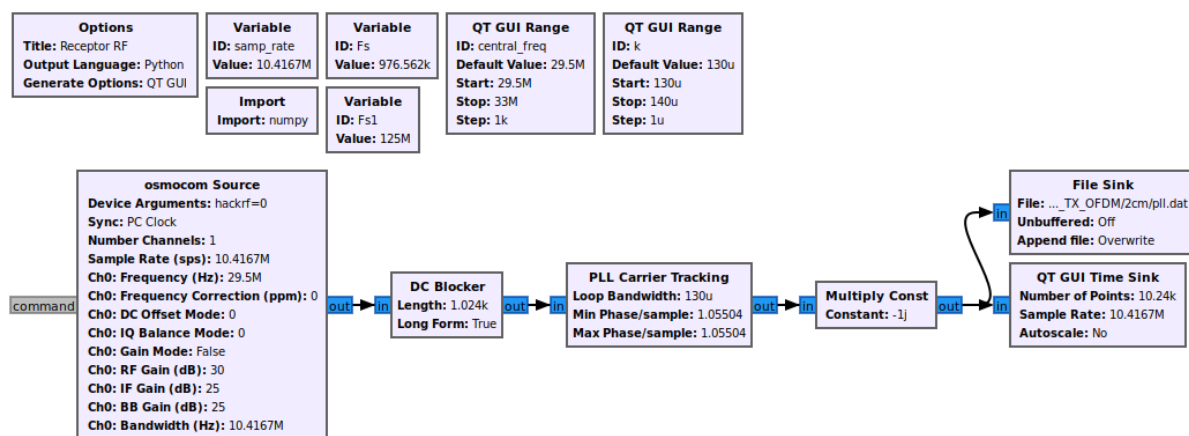


Figura 3.22: Receptor RF implementado en GNU Radio

3.4.3.2. Demodulador Multiportadora

La Figura 3.23 muestra la implementación del demodulador multiportadora en GNU Radio, con bloques diseñados mayoritariamente en Python, conforme a lo especificado en la Sección 3.4.1. A continuación, se describe la función de cada bloque:

- **Interpolating FIR Filter:** Este bloque aplica el filtro acoplado conforme al conjugado de la señal multiportadora en banda base.
- **Máxima Similitud - Pulso de Sincronización:** Este bloque encuentra el instante en que comienza una trama multiportadora a partir de la señal resultante del filtro acoplado. Su funcionamiento se divide en tres fases:

1. Encontrar los instantes de máxima similitud a partir del componente en fase de la señal resultante del filtro acoplado.
 2. Evaluar los posibles instantes de máxima similitud conforme a (3.37).
 3. Formar la señal de salida del bloque, donde los instantes de máxima similitud corresponden a la muestra de la señal de entrada, y los demás instantes toman un valor nulo.
- **Máxima Similitud - Trama de Sincronización:** Este bloque encuentra el instante en que comienza la primera trama multiportadora a partir de la señal resultante del filtro acoplado. Su funcionamiento es similar al bloque **Máxima Similitud - Pulso de Sincronización**.
 - **Determinación Borde Ascendente:** Este bloque forma un pulso rectangular unitario cuando su señal de entrada es diferente de cero. La duración del pulso (**Num_samp**) depende de la aplicación. Para determinar el inicio de la trama multiportadora, se utiliza una duración de 5 muestras, de manera que su flanco de bajada indique el inicio del muestreo de la señal de entrada. Para la detección de la primera trama multiportadora, se opta por una duración equivalente al tiempo de transmisión de todos los mensajes cargados en la FPGA (sin contar los mensajes de sincronización). El producto entre las señales mencionadas (Bloque Multiply de la Figura 3.23) indica los instantes en que debe iniciarse el muestreo de la señal de entrada, evitando así los mensajes de sincronización.
 - **Muestreo de señal multiportadora:** Este bloque muestrea la señal de entrada conforme a los flancos de bajada de la salida del bloque Multiply. La frecuencia de muestreo es igual a la frecuencia de la señal multiportadora antes de su interpolación. Además, toma $8 \cdot N_{TX}$ muestras, donde N_{TX} es el número de tramas multiportadoras acompañadas del pulso de sincronización. Una vez tomada esa cantidad de muestras, se espera el siguiente flanco de bajada.
 - **OFDM / GFDM:** Este bloque toma las muestras de la señal multiportadora para aplicar el método mencionado en la Sección 3.4.1.3.

Evaluación del Sistema

4.1. Metodología para realizar las pruebas y mediciones

La evaluación del sistema de transmisión multiportadora se realiza en dos etapas de experimentación. El primer experimento evalúa la respuesta del receptor tras su proceso de sincronización al utilizar el controlador PLL. Para este experimento, se coloca el transmisor a una distancia fija de 10 cm (en la que se cuenta con niveles apropiados de SNR) y se analiza y evalúa su proceso de sincronización y el correcto funcionamiento del demodulador implementado. El segundo experimento consiste en variar la distancia entre el transmisor y el receptor a una tasa de 4 cm por vez, comenzando en 10 cm hasta llegar a 26 cm. Como resultado, se evaluará la respuesta del Bit Error Rate (BER) en cada una de las distancias mencionadas.

La distancia máxima de 26 cm se estableció debido a las características técnicas de la tarjeta *Field Programmable Gate Array* (FPGA) STEMLab 125-14, cuya máxima potencia de salida es de 10dBm (0.01 W), lo que impide realizar una transmisión a largas distancias. Además, el transmisor utiliza una antena telescópica diseñada para operar en un rango de frecuencias de 75 MHz a 1 GHz [86], pero la frecuencia portadora de la señal modulada está limitada por la frecuencia de reloj de la FPGA STEMLab 125-14. Esta situación reduce la eficiencia de radiación de la antena, provocando que la potencia radiada sea menor que la potencia suministrada por la propia tarjeta FPGA, es decir, el coeficiente de reflexión es diferente de cero. Estas limitaciones podrían mejorarse mediante la integración de hardware externo en el sistema de transmisión. Sin embargo, el objetivo general del trabajo de integración curricular es implementar un sistema de transmisión multiportadora sobre una plataforma FPGA. Por este motivo, el diseño no incluyó hardware RF externo, ya que no es objeto de estudio en este trabajo de integración curricular.

El protocolo para la realización de pruebas y mediciones se aplica de la misma manera para ambos moduladores multiportadora implementados (OFDM y GFDM). La medición del BER que se realiza en el segundo experimento se basa en la expresión dada por (4.1) a continuación.

$$\text{BER} = \frac{\text{numero total de bits erroneos}}{\text{numero total de bits}} \quad (4.1)$$

La información utilizada para el proceso de evaluación del sistema de transmisión consiste en 60 palabras de 12 bits generadas de forma aleatoria utilizando el lenguaje de programación Python. Aunque la matriz completa de información transmitida está formada por 63 palabras de 12 bits, las últimas 3 codewords no son aleatorias, sino que, forman una trama de fin de

transmisión que se utiliza para identificar el inicio y fin de cada transmisión en el lado del receptor. De esta manera, es posible ejecutar el algoritmo de pruebas varias veces, identificando cada una de las transmisiones realizadas.

Para las mediciones se desarrolló un algoritmo en Python para automatizar la conexión SSH y la ejecución del programa de transmisión y recepción de datos. Este algoritmo, basado en las librerías Paramiko, sched, time, os y subprocess, permite establecer una conexión segura con el dispositivo remoto, programar eventos para la ejecución del programa de transmisión y recepción en intervalos de tiempo específicos, y ejecutar comandos de consola en el sistema operativo Ubuntu para la transferencia de datos.

El experimento 1 incluye un proceso de sintonización manual para la primera sintonización del PLL. En el experimento 2, las variaciones no automatizadas se refieren a las variaciones de distancia entre transmisor y receptor como parámetros de ajuste mecánico.

4.1.1. Metodología para realizar el experimento 1

El primer experimento tiene como objetivo evaluar el funcionamiento de la modulación multiportadora utilizando GNU Radio a una distancia fija de 10 cm. El procedimiento seguido para realizar el primer experimento es el siguiente:

1. Seleccionar el tipo de modulación multiportadora a evaluar.
2. Establecer una conexión *Secure Shell* (SSH) desde el computador con Ubuntu 22.04 hacia la tarjeta STEMLab 125-14.
3. Copiar el archivo bitstream correspondiente a la modulación seleccionada desde el computador con Ubuntu 22.04 hacia la tarjeta STEMLab 125-14.
4. Posicionar la tarjeta STEMLab 125-14 a una distancia de 10 cm del receptor HackRF One.
5. Ejecutar el algoritmo receptor programado en GNU Radio para comenzar a recibir señales RF.
6. Subir el programa bitstream a la tarjeta FPGA STEMLab 125-14 utilizando el comando:
redpitaya>cat /opt/redpitaya/fpga/bitstream_OFDM/GFDM.bit > /dev/xdevcfg
7. Ejecutar el algoritmo en la tarjeta FPGA de forma continua hasta sintonizar correctamente los parámetros de control del PLL en GNU Radio.

8. Una vez sintonizados correctamente los parámetros del controlador PLL, se procede a ejecutar por 12 segundos controlados por un script de Python, el programa bitstream y evaluar el comportamiento del transmisor para la distancia de 10 cm de separación.

4.1.2. Metodología para realizar el experimento 2

El experimento 2 se inicia una vez establecidos los parámetros óptimos de sintonización del controlador PLL. Tras sincronizar la señal modulada de transmisión de la tarjeta FPGA con el reloj interno definido por software, se procede con la siguiente metodología para capturar la información a distintas distancias.

1. Situar las tarjetas a una distancia inicial de 10 cm entre el transmisor y el receptor.
2. Encender el programa receptor GNU Radio y comenzar a recibir las señales RF.
3. Transmitir la señal multiportadora modulada durante 12 segundos.
4. Guardar la información recibida para su posterior análisis.
5. Apagar el sistema de transmisión y recepción después de 12 segundos.
6. Aumentar la distancia entre los dispositivos en 4 cm, es decir, de 10 cm a 14 cm, y repetir hasta alcanzar 26 cm en cinco incrementos. Regresar al paso 2 para la recolección de información.

Los experimentos 1 y 2 comparten la misma metodología base para capturar información, pero el experimento 1 emplea una distancia fija de 10 cm, mientras que el experimento 2 introduce una variable de distancia, variando entre 10 cm y 26 cm con incrementos de 4 cm. Esta variación permite evaluar el comportamiento del sistema en diferentes escenarios de alcance lo que posibilita evaluar el sistema para diferentes valores de SNR.

4.2. Resultados de la evaluación experimental

4.2.1. Evaluación del experimento 1

El experimento 1 tiene como objetivo verificar el funcionamiento tanto del sistema transmisor como del receptor. Para el transmisor, se necesita confirmar que su señal de salida corresponda al modelo matemático implementado en la FPGA. Para el receptor, es necesario comprobar que se tiene una menor cantidad de errores en mejores condiciones de canal. Para evaluar

su funcionamiento, se toma la salida del receptor RF dada por (3.4.3.1) y del Demodulador Multiportadora, dada por (3.4.3.2), para su procesamiento en MATLAB.

A partir de las muestras capturadas por el receptor RF, se puede encontrar la mejor coincidencia con respecto a una señal multiportadora simulada en MATLAB. Para ello, se utiliza la correlación cruzada, la cual devuelve el instante en que se alcanza la máxima similitud con respecto a la señal capturada. Basándose en el análisis de correlación cruzada, se pretende encontrar la porción en que la señal capturada se asemeje a la señal multiportadora que modula las tramas de sincronización. Cabe mencionar que la señal simulada en MATLAB contiene el mismo modelo matemático que la implementación en el STEMLab 125-13.

Sin embargo, para observar una mejor similitud, es necesario encontrar el factor de atenuación por espacio libre α , cuyo valor puede ser calculado según (4.2), siendo P_{RX} la potencia de la señal multiportadora recibida sin considerar el efecto del ruido y P_{TX} corresponde a la potencia de la señal multiportadora transmitida.

$$\begin{aligned}
 P_{RX} &= \frac{1}{T} \int_0^T \alpha^2 \cdot |x_{MC_{CP}}|^2 dt \\
 P_{RX} &= \alpha^2 \cdot P_{TX} \\
 \alpha &= \sqrt{\frac{P_{RX}}{P_{TX}}}
 \end{aligned} \tag{4.2}$$

A partir de las muestras capturadas es posible calcular su potencia, sin embargo, esta potencia incluye el ruido. Así pues, para usar la ecuación (4.2) es necesario encontrar la potencia de recepción sin el efecto del ruido, la cual se puede aproximar mediante la ecuación (4.3). Cabe mencionar, que es posible calcular la potencia del ruido presente en el canal, dado que los valores muestreados de la señal multiportadora también incluyen muestras que corresponden al ruido Gaussiano.

$$P_{RX} \approx P_{RX+Ruido} - P_{n_0} \tag{4.3}$$

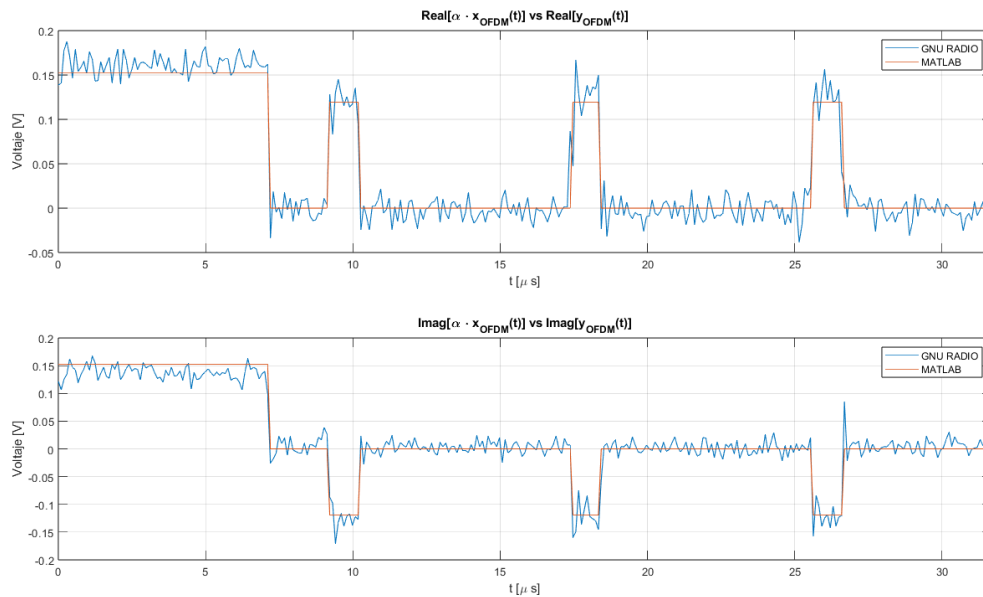


Figura 4.1: Señal OFDM receptada versus señal OFDM simulada

Las Figuras 4.1 y 4.2 muestran la parte real e imaginaria de las señales recibidas y simuladas para las modulaciones *Orthogonal Frequency Division Multiplexing* (OFDM) y *Generalized Frequency Division Multiplexing* (GFDM), respectivamente. Los resultados destacan que la señal recibida por GNU Radio tiende a tener la misma forma que la señal simulada, manteniendo incluso el mismo ancho de pulso para cada muestra interpolada. Esto demuestra que la implementación realizada en el STEMLab 125-14 funciona correctamente, cumpliendo con las predicciones hechas en MATLAB. Además, se confirma el correcto funcionamiento del mezclador de cuadratura implementado en la FPGA, ya que la señal recibida corresponde al paso de pasa banda a banda base. Estos resultados establecen que el transmisor multiportadora funciona correctamente.

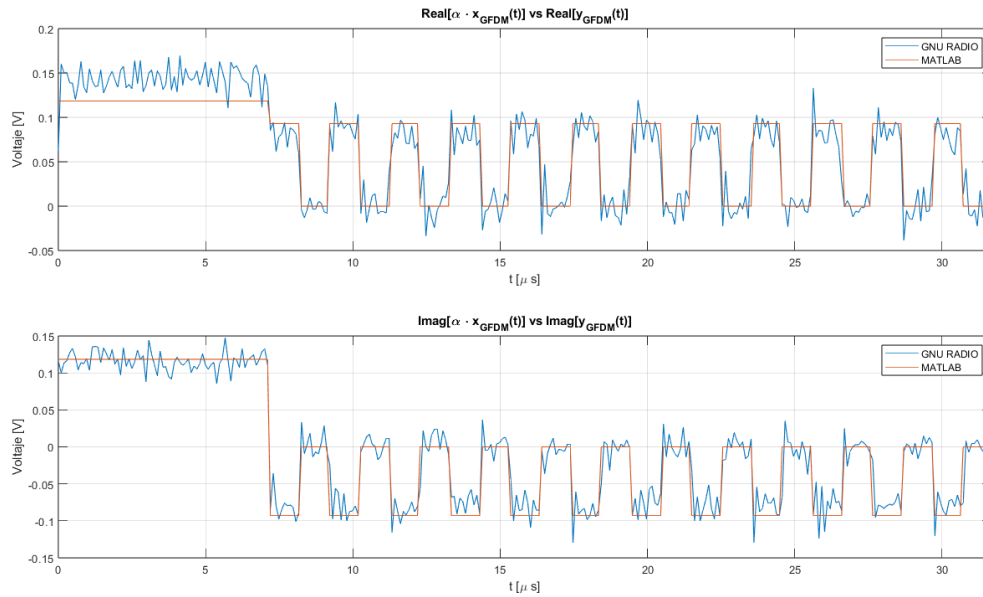


Figura 4.2: Señal GFDM receptada versus señal GFDM simulada

Una forma sencilla de demostrar el funcionamiento del demodulador multiportadora es obtener la constelación de los símbolos 4-QAM en las mejores condiciones de canal. Esto se muestra en la Figura 4.3, donde la agrupación de varios símbolos demodulados forma la constelación característica de los símbolos 4-QAM, independientemente del tipo de modulación utilizado. Estos resultados demuestran el correcto funcionamiento del demodulador multiportadora.

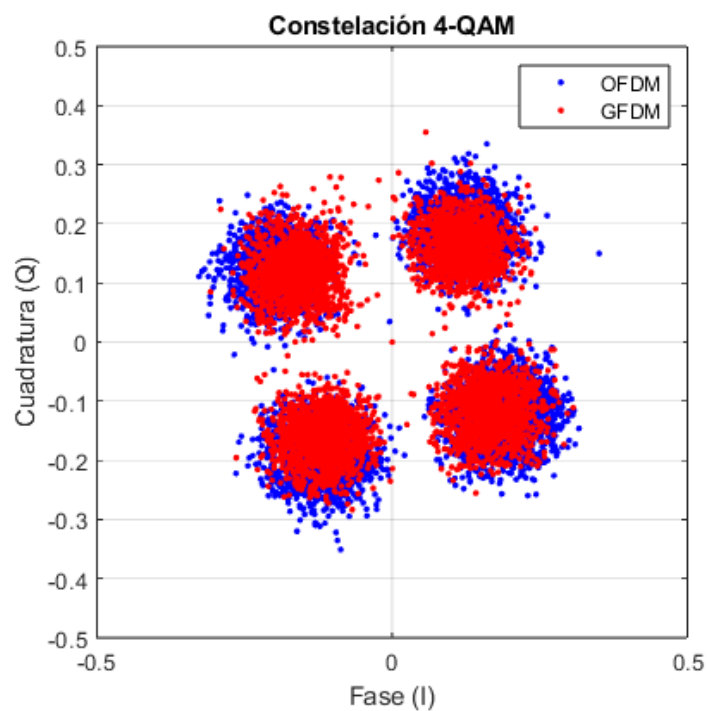


Figura 4.3: Constelaciones 4-QAM para OFDM y GFDM

4.2.2. Evaluación del experimento 2

El experimento 2 evalúa el sistema de transmisión multiportadora utilizando las modulaciones OFDM y GFDM. La Figura 4.4 compara el BER y la distancia para las modulaciones OFDM y GFDM. El gráfico está en escala lineal para los valores de BER y la distancia, que se mide en [cm]. La Tabla 4.1 muestra los resultados obtenidos en el experimento 2. La tabla incluye las distancias de separación en [cm], el valor medio del BER medido para cada distancia de evaluación y el nivel de SNR calculado en [dB].

Los resultados del SNR calculado presentados en la Tabla 4.1 fueron estimados utilizando la ecuación (4.4), donde P_{n_0} corresponde a la potencia del ruido y $P_{RX+Ruido}$ corresponde a la potencia de la señal capturada.

$$SNR[dB] = 10 \log_{10} \left(\frac{P_{RX+Ruido} - P_{n_0}}{P_{n_0}} \right) \quad (4.4)$$

Tabla 4.1: Resultados del experimento 2.

Distancia [cm]	Modulación OFDM		Modulación GFDM	
	SNR [dB]	BER	SNR [dB]	BER
10	15.4	0.000847271	16.01	0.0014536682
14	11.94	0.012156602	12.78	0.0133735165
18	10.6	0.027879918	10.47	0.040057406
22	8.54	0.058608714	9.22	0.06208277
26	6.27	0.11009775	6.83	0.110503726

La evaluación incluye un análisis del intervalo de confianza del 95% para ambas modulaciones, con el objetivo de verificar la consistencia de los resultados del BER. Este intervalo proporciona una medida cuantitativa de la precisión y fiabilidad de la estimación de la media del BER. Para el análisis del intervalo de confianza, se capturaron cientos de transmisiones durante 12 segundos. Algunas transmisiones fueron descartadas por el sistema receptor debido a un desfase mayor al permitido, lo que impedía la correcta identificación de los bits modulados.

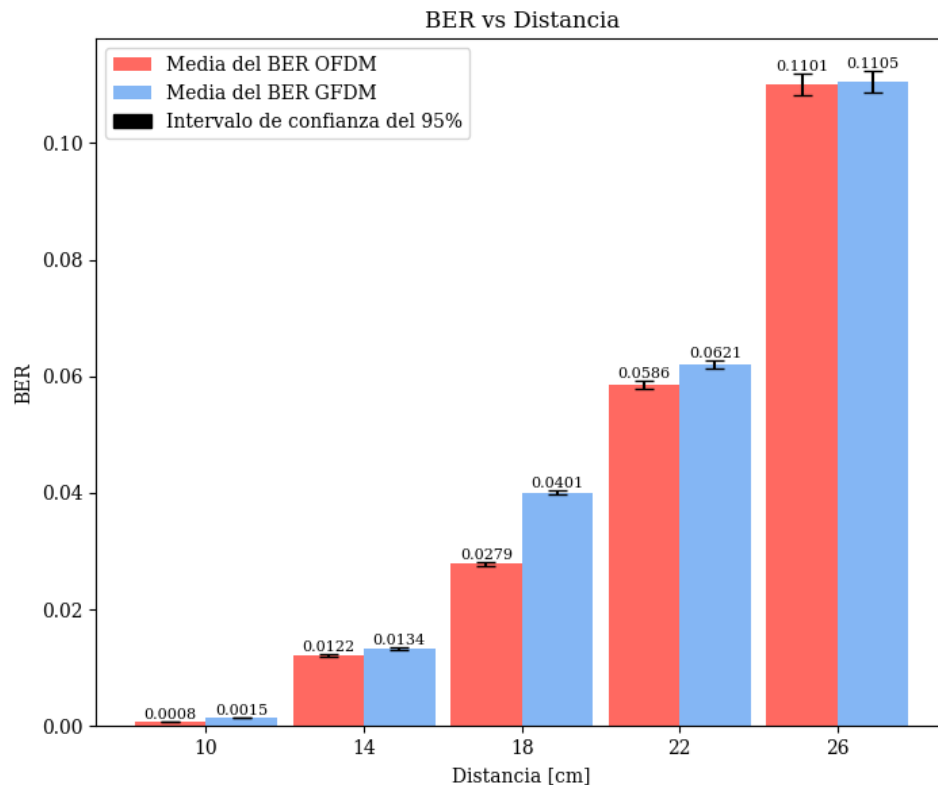


Figura 4.4: Comparación del BER vs Distancia para modulaciones OFDM y GFDM.

La interpretación de la Figura 4.4 demuestra que los valores del BER para la modulación OFDM son menores que para la modulación GFDM. Esto indica que el rendimiento general de la modulación OFDM es superior al de GFDM en términos de BER, ya que su tasa de errores por millón es menor.

El intervalo de confianza calculado en los puntos de medición tiene un rango pequeño de variación para las distancias más cortas entre los módulos transmisor y receptor. Sin embargo, este rango aumenta proporcionalmente con la distancia. A pesar de este comportamiento, el rango sigue siendo pequeño, lo que implica que la media real no está muy alejada de la media estimada y que posee un 95% de certeza de que cualquier valor medido de BER se encontrará dentro del intervalo de confianza, al igual que la media real. Este resultado proporciona un respaldo fiable para las medidas del experimento realizado.

La Figura 4.5 evalúa el rendimiento de cada modulación multiportadora, al visualizar la relación entre BER y SNR. El eje vertical está en escala logarítmica para facilitar la interpretación del BER, mientras que el eje horizontal se presenta en unidades de decibelios.

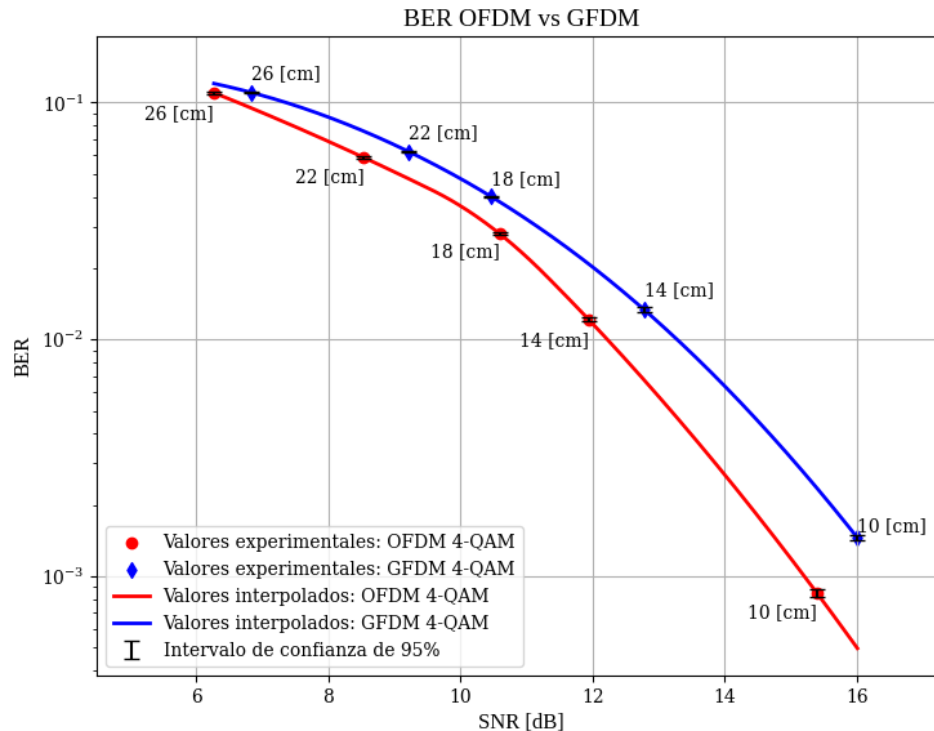


Figura 4.5: Comparación del BER vs SNR para modulaciones OFDM y GFDM.

La Figura 4.5 demuestra que la modulación OFDM requiere un nivel de SNR menor que GFDM para obtener el mismo nivel de BER. Esto se puede comprobar al analizar el punto medido a 14 cm mostrado en la Tabla 4.1, ya que en esa distancia las medidas de BER son similares, pero el nivel de SNR necesario para GFDM es aproximadamente 1.0704 veces (o un 7% más alto) que el necesario en OFDM para obtener un BER similar.

$$\text{Relacion}_{14 \text{ [cm]}} = \frac{12.78}{11.94}$$

$$\text{Relacion}_{14 \text{ [cm]}} = 1.0704$$

Basándonos en el comportamiento de las curvas mostradas en la Figura 4.5 es congruente afirmar que la modulación OFDM presenta una mejor proporción entre BER medido y SNR calculado. Implicando que la modulación OFDM tiene una mayor eficiencia en la transmisión de datos y una mejor resistencia al ruido y a las interferencias. Esto se traduce en una superior fiabilidad y calidad en la comunicación utilizando la modulación OFDM en comparación con GFDM.

Conclusiones, Desafíos, Recomendaciones y Trabajos Futuros

5.1. Conclusiones

En el presente trabajo de integración curricular se evaluaron diferentes *Field Programmable Gate Array* (FPGA) disponibles en la Universidad de Cuenca, determinándose que la FPGA STEMLab 125-14 es la mejor opción según los criterios de selección planteados por el Método Analytic Hierarchy Process. Sin embargo, esto no implica que el resto de las FPGA disponibles no sean aptas para la implementación de un sistema de transmisión multiportadora. Por ejemplo, la tarjeta de desarrollo Basys 3 posee una mayor cantidad de recursos en comparación con la FPGA STEMLab 125-14, lo que hace que el sistema de transmisión multiportadora implementado en la FPGA STEMLab 125-14 sea compatible con la Basys 3. No obstante, para este último es necesario agregar componentes RF para que el mezclador de cuadratura transmita a altas frecuencias, e incluso se requiere un filtro pasa-bajo para limitar el ancho de banda, con el fin de obtener un funcionamiento similar al presentado en este trabajo de integración curricular. Cabe mencionar que, en el caso de la FPGA Spartan 3E, no se podría implementar el sistema de transmisión multiportadora, dado que los IP Core Floating Point de Xilinx (parte central en la implementación del sistemas de transmisión multiportadora) no están disponibles en su entorno de descripción de hardware *ISEDesign*. Ante estas características, se define que la implementación de sistemas multiportadora presentada en este trabajo de integración curricular puede ser realizada en cualquier FPGA disponible en Vivado, cuyos recursos sean iguales o superiores a los de la FPGA STEMLab 125-14.

Por otra parte, la representación vectorial de la modulación multiportadora permitió disminuir la complejidad de implementación sobre una FPGA. Esto se debe a que el problema central para obtener muestras de señales multiportadoras equivale a un producto matriz-vector. Además, debido a la flexibilidad de las plataformas FPGA para ejecutar procesos en paralelo, la obtención de las muestras de una señal multiportadora es mucho más rápida en comparación con un microcontrolador con un reloj igual al disponible en una FPGA. Sin embargo, es necesario mencionar que a medida que aumenta la complejidad matemática, se requiere una mayor cantidad de recursos. Tal es el caso del filtro paso bajo utilizado para la interpolación, ya que, si se opta por la implementación de un filtro FIR es necesario la adición de un mayor número de coeficientes, agotando exponencialmente los recursos disponibles de la FPGA. Por ello, en la implementación presentada se optó por el uso de un interpolador, cuya complejidad matemática es significativamente inferior a la aplicación de un filtro FIR. Ante esto, se establece que para aprovechar el máximo potencial de un sistema de transmisión implementado

en una FPGA es necesario definir modelos matemáticos que requieran una menor cantidad de recursos para su implementación. Como resultado, se implementaron dos sistemas de modulación multiportadora: OFDM y GFDM. Los esquemas de modulación fueron evaluados en función de su BER, revelando que la modulación GFDM en el escenario propuesto requiere un 7% más de SNR para obtener un nivel de BER similar al de OFDM. Esto implica que la modulación OFDM es superior a GFDM en la relación BER vs SNR, ya que OFDM necesita un menor nivel de SNR para alcanzar niveles similares de BER, sin embargo es necesario continuar el estudio de GFDM debido a su flexibilidad de ancho de banda, que permite un uso eficiente y configurable del espectro radioeléctrico.

Finalmente, el sistema de recepción implementado en GNU Radio tiene dos funciones. La primera consiste en detectar la primera muestra correspondiente a la señal multiportadora a partir de la respuesta de un Matched Filter. La segunda fase radica en aplicar el método de demodulación multiportadora, siendo el utilizado en este trabajo de integración curricular el Zero Forcing. Sin embargo, el diseño del receptor implementado en GNU Radio sirve exclusivamente para fines evaluativos del sistema de transmisión multiportadora implementado en la FPGA STEMLab 125-14. Esto se debe a que descarta aquellas señales que sobrepasen los 15° de desfase, con el objeto de capturar muestras de la señal multiportadora que fueron afectadas en su mayor proporción por ruido Gaussiano. Este diseño de receptor no es aplicable en la práctica, ya que descarta información que puede ser crucial. Por ello, en sistemas de comunicaciones se utilizan símbolos pilotos y codificación de canal para corregir aquellos eventos catastróficos en los que se pierde la sincronización, o en el peor de los casos, se implementa un protocolo para retransmitir el mensaje que fue afectado por un evento catastrófico.

5.2. Desafíos en la Implementación

Los desafíos encontrados a lo largo del desarrollo del presente trabajo de integración curricular son los siguientes:

- Implementar conceptos matemáticos como la Transformada Inversa de Fourier Discreta, filtros paso bajo discretos, mezcladores de cuadratura y operaciones con valores complejos en una descripción de hardware. Para abordar esta problemática, se redujo la complejidad de la implementación, identificando modelos matemáticos simples equivalentes a los conceptos matemáticos originales.
- Implementar un sistema de transmisión en la FPGA independiente de elementos exter-

nos, como programas en lenguajes de alto nivel como Python. Para superar este desafío, se decidió implementar señales de control entre los componentes descritos en VHDL, lo que permite mantener una sincronización adecuada entre los componentes diseñados para ejecutar correctamente el modelo matemático propuesto.

- Entender todas las implicaciones prácticas de una descripción de hardware es crucial, ya que los diseños pueden funcionar correctamente en simulaciones del entorno Vivado, pero fallar en una implementación práctica. Para solucionar esto, es necesario diseñar descripciones de hardware que eviten los Negative Time Slacks y realizar pruebas de funcionamiento de cada componente mediante el mapeo de señales de control a salidas digitales.

La implementación de un sistema de transmisión multiportadora en una plataforma FPGA presenta la dificultad de trasladar conceptos matemáticos complejos al nivel de hardware. Esto implica que, a partir de operaciones lógicas simples, se debe representar un concepto matemático avanzado. Para lograr la convergencia del concepto matemático aplicado, es esencial dominar el entorno de descripción de hardware, lo cual permite resolver problemas de síntesis y retardos de tiempo. Finalmente, la plataforma FPGA debe operar como un sistema de transmisión independiente, sin recurrir a hardware externo.

5.3. Recomendaciones

Para replicar el presente trabajo de integración curricular, se recomienda seguir las siguientes consideraciones:

- Revisar el funcionamiento de los componentes adicionales de la tarjeta STEMLab 125-14, ya que el uso del DAC interno de la FPGA requirió la adición un bloque de interpolación para su funcionamiento.
- En la descripción de hardware, se recomienda evitar el uso de tipos de datos no sintetizables o no estandarizados, como “Natural”, “Array” y “Real”. Es preferible utilizar tipos de datos estandarizados, específicamente “std_logic”.
- Tener en cuenta que la programación en VHDL se realiza mediante descripción de hardware y, por lo tanto, no debe utilizar la lógica de programación convencional de los lenguajes secuenciales tradicionales, ya que el sintetizador de Vivado no podrá sintetizar esta lógica.

- Emplear tarjetas FPGA de menor complejidad, como la Basys 3, para facilitar el proceso de depuración. La tarjeta Basys 3 dispone de cuatro puertos de 16 pines que permiten visualizar el comportamiento de bits individuales, lo cual resulta ventajoso para la identificación y resolución de errores en la descripción de hardware.
- Evitar la creación de retardos negativos o negative time slacks en el diseño final descrito en VHDL, ya que estos podrían impedir que la tarjeta FPGA sintetice el diseño final. Para ello, se recomienda incluir retardos en el diseño VHDL, ya sea en los bloques IP Core o en el desarrollo del propio circuito descrito en VHDL.

5.4. Trabajos Futuros

Actualmente los sistemas de comunicación móvil utilizan esquemas de modulación multiportadora como OFDM, por tanto, la implementación de sistemas de transmisión multiportadora sobre una plataforma FPGA establece un precedente para el desarrollo de diversas líneas de investigación futuras que podrían ampliar significativamente el alcance y la aplicabilidad del presente trabajo de integración curricular. Con base en los resultados y las limitaciones identificadas en esta tesis, se proponen varias líneas de trabajo para futuras investigaciones:

Se sugiere utilizar hardware más actualizado para mejorar la resolución y precisión de los resultados, ya que la tarjeta STEMLab empleada en esta tesis fue fabricada en 2013 (se utilizó dada su disponibilidad en la Universidad) y presenta algunas limitaciones en sus características técnicas.

Para mejorar el sistema de transmisión presentado en este trabajo de integración curricular se recomienda, la implementación de técnicas de estimación de canal y ecualización, puesto que, podría mejorar significativamente el desempeño del sistema multiportadora. Estas técnicas pueden ser implementadas mediante la inserción de portadoras piloto en el diseño de VHDL.

También se recomienda la adición de esquemas de codificación de canal, como los códigos convolucionales o la codificación Reed-Solomon, que se utiliza actualmente en el estándar LTE de redes móviles 4G.

Finalmente, se recomienda experimentar con la configuración de la modulación GFDM mediante el uso otros tipos de filtros formadores de pulso. Estos filtros tienen el potencial de mejorar los resultados actuales de GFDM en comparación con OFDM.

Referencias

- [1] E. Dahlman, S. Parkvall, y J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2011, num. 978-0-12-385489-6. [En línea]. Disponible: <https://www.sciencedirect.com/book/9780123854896/4g-lte-lte-advanced-for-mobile-broadband>
- [2] H. Bogucka, A. Kliks, y P. Kryszkiewicz, *Advanced Multicarrier Technologies for Future Radio Communication: 5G and Beyond*, 1ra ed. John Wiley & Sons, 2017.
- [3] M. Agiwal, A. Roy, y N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, num. 3, pp. 1617–1655, 2016.
- [4] N. Michailow, M. Matthé, I. S. Gaspar, A. N. Caldevilla, L. L. Mendes, A. Festag, y G. Fettweis, "Generalized Frequency Division Multiplexing for 5th Generation Cellular Networks," *IEEE Transactions on Communications*, vol. 62, num. 9, pp. 3045–3061, 2014.
- [5] J. Machado Fernández, "Software Defined Radio: Basic Principles and Applications," *Revista Facultad de Ingeniería*, vol. 24, pp. 79–96, 01 2015.
- [6] Godai Group LLC, "HackRF One bundle," 2024, accessed: 06/18/2024. [En línea]. Disponible: <https://hackerwarehouse.com/product/hackrf-one-kit/>
- [7] M. Perotoni y K. Marconi, "SDR-based spectrum analyzer based in open-source GNU Radio," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 20, pp. 542–555, 09 2021.
- [8] Osmocom, "RTL-SDR," <https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr>, 2024, Accessed: 06/06/2024.
- [9] Digilent, "Spartan-3e," 2023, accessed: 06/10/2024. [En línea]. Disponible: <https://digilent.com/reference/programmable-logic/spartan-3e/start>
- [10] —, "Basys 3," 2023, accessed: 06/10/2024. [En línea]. Disponible: <https://digilent.com/reference/programmable-logic/basys-3/start>
- [11] Red Pitaya, "Stemlab 125-14," 2024, accessed: 06/10/2024. [En línea]. Disponible: <https://redpitaya.com/stemlab-125-14/>
- [12] T. L. Saaty, "Decision making with the analytic hierarchy process," *International Journal of Services Sciences*, vol. 1, pp. 83–98, 2008.
- [13] Red Pitaya, "1.4.2.5. frequency counter," https://redpitaya-knowledge-base.readthedocs.io/en/latest/learn_fpga/4_lessons/FreqCounter.html, 2024, Accessed: 06/18/2024.

- [14] RedPitaya, “Stemlab 125-14,” <https://redpitaya.readthedocs.io/en/latest/developerGuide/hardware/125-14/fastIO.html#analog-outputs>, Accessed: 24/06/2024.
- [15] Radio project, “About GNU Radio,” <https://www.gnuradio.org/about/>, 2024, Accessed: 05/29/2024.
- [16] GNU Radio, “PLL carrier tracking,” https://wiki.gnuradio.org/index.php/PLL_Carrier_Tracking, 2024, Accessed: 06/06/2024.
- [17] —, “File sink,” https://wiki.gnuradio.org/index.php/File_Sink, 2024, Accessed: 06/06/2024.
- [18] —, “File source,” https://wiki.gnuradio.org/index.php/File_Source, 2024, Accessed: 06/06/2024.
- [19] Xilinx, “Spartan-3e FPGA family data sheet,” December 2018, accessed: 06/10/2024. [En línea]. Disponible: <https://docs.amd.com/v/u/en-US/ds312>
- [20] —, “7 series FPGAs data sheet: Overview,” September 2020, accessed: 06/10/2024. [En línea]. Disponible: https://docs.amd.com/v/u/en-US/ds180_7Series_Overview
- [21] L. L. Jiménez, “Decentralized location-aware orchestration of containerized microservice applications: Enabling distributed intelligence at the edge,” Ph.D. dissertation, Luleå University of Technology, 2020.
- [22] E. A. Nantes, “El método analytic hierarchy process para la toma de decisiones. repaso de la metodología y aplicaciones,” *Revista de la Escuela de Perfeccionamiento en Investigación Operativa*, vol. 27, num. 46, pp. 54–73, nov. 2019. [En línea]. Disponible: <https://revistas.unc.edu.ar/index.php/epio/article/view/26474>
- [23] R. Saaty, “The analytic hierarchy processwhat it is and how it is used,” *Mathematical Modelling*, vol. 9, num. 3, pp. 161–176, 1987. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/0270025587904738>
- [24] 3rd Generation Partnership Project (3GPP), “3rd generation partnership project; technical specification group radio access network; feasibility study for OFDM for UTRAN enhancement; (release 6),” 3rd Generation Partnership Project, Valbonne - Francia, Technical Report TR 25.892 V0.2.0, 3 2003. [En línea]. Disponible: https://www.3gpp.org/ftp/Specs/archive/25_series/25.892/25892-020.zip

- [25] N. Pathak, "OFDM (orthogonal frequency division multiplexing) simulation using MATLAB," *International Journal of Engineering Research & Technology*, vol. 1, num. 6, pp. 1–6, 8 2012.
- [26] X. Li, "A study of OFDM transmission channels," in *2023 3rd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*. Los Alamitos, CA, USA: IEEE Computer Society, 2 2023, pp. 659–665. [En línea]. Disponible: <https://doi.ieeecomputersociety.org/10.1109/ACCTCS58815.2023.00125>
- [27] A. Desai, A. Gupta, M. Jambhale, y V. Chavan, "Efficient implementation technique for OFDM on FPGA," *SSRN Electronic Journal*, 5 2021.
- [28] M. Mohamed, A. Samarah, y M. F. Allah, "A novel implementation of OFDM using FPGA," *IJCSNS International Journal of Computer Science and Network Security*, vol. 11, num. 11, 11 2011.
- [29] B. Hu, M. Irshad, S. Noman, X. Tang, C. Song, y S. Haider, *Design and Simulation of Orthogonal Frequency-Division Multiplexing (OFDM) Signaling*. Springer Nature, 01 2021, pp. 187–197.
- [30] R. V. Pacelli, A. M. P. d. Lucena, y S. S. Figueiredo, "Técnica de sincronização de portadora para sistemas de comunicação OFDM / carrier synchronization technique for OFDM communication systems," *Brazilian Journal of Development*, vol. 6, num. 3, p. 1429714305, 3 2020. [En línea]. Disponible: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/7944>
- [31] X. Luo, "The application of OFDM in optical fiber communication systems," *IOP Conference Series: Earth and Environmental Science*, vol. 332, num. 4, p. 042010, oct 2019. [En línea]. Disponible: <https://dx.doi.org/10.1088/1755-1315/332/4/042010>
- [32] A. Ngom y A. D. Kora, "Combination of OFDM and cdma techniques for a high bandwidth optimization and a great improvement of signal quality in OFDM systems," in *Wireless Internet*, D.-J. Deng, A.-C. Pang, y C.-C. Lin, Eds. Cham: Springer International Publishing, 2020, pp. 220–236.
- [33] X. Zhang y G. Tao, "The research of adaptive modulation technology in OFDM system," in *Proceeding of 2021 International Conference on Wireless Communications, Networking and Applications*, Z. Qian, M. Jabbar, y X. Li, Eds. Singapore: Springer Nature Singapore, 2022, pp. 739–748.

- [34] E. Hassan, *Multi-Carrier Communication Systems with Examples in MATLAB A New Perspective*. CRC Press Taylor & Francis Group, 10 2015.
- [35] Z. Pi y F. Khan, "System design and network architecture for a millimeter-wave mobile broadband (mmb) system," in *34th IEEE Sarnoff Symposium*, 2011, pp. 1–6.
- [36] G. Wunder y otros, "5GNOW: Non-orthogonal, asynchronous waveforms for future mobile applications," *IEEE Commun. Mag.*, vol. 52, num. 2, pp. 97–105, 2 2014.
- [37] W. Xiang, K. Zheng, y X. Shen, Eds., *5G Mobile Communications*. Springer International Publishing, 2017.
- [38] G. Fettweis y S. Alamouti, "5G: Personal mobile INTERNET beyond what cellular did to telephony," *IEEE Commun. Mag.*, vol. 52, num. 2, pp. 140–145, 2 2014.
- [39] N. Michailow, I. Gaspar, S. Krone, M. Lentmaier, y G. Fettweis, "Generalized frequency division multiplexing: Analysis of an alternative multi-carrier technique for next generation cellular systems," in *Proc. Int. Symp. Wireless Commun. Syst.*, 2012, pp. 171–175.
- [40] Samsung Electronics Co., "5G vision, white paper," 2015.
- [41] J.-B. Dore, V. Berg, y D. Ktenas, "Performance of FBMC multiple access for relaxed synchronization cellular networks," in *Proc. IEEE GlobeCom Workshop Broadband Wireless Access*, 2014, pp. 983–988.
- [42] B. F. Boroujeny, "Filter bank multicarrier (FBMC): An integrated solution to spectrum sensing and data transmission in cognitive radio networks," Presented at the PHYDYAS Workshop, Perugia, Italy, 2009.
- [43] E. Hamiti y F. Sallahu, "Spectrum comparison between GFDM, OFDM and GFDM behavior in a noise and fading channel," *The International Journal of Electrical and Computer Engineering Systems*, vol. 2, pp. 39–43, 11 2015.
- [44] E. Öztürk, E. Basar, y H. A. Çrpan, "Spatial modulation GFDM: A low complexity MIMO-GFDM system for 5G wireless networks," in *2016 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2016, pp. 1–5.
- [45] N. Chiurtu, B. Rimoldi, y E. Telatar, "On the capacity of multi-antenna Gaussian channels," in *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252)*, 2001, pp. 53–.

- [46] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, num. 8, pp. 1451–1458, 10 1998.
- [47] V. Tarokh, N. Seshadri, y A. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, num. 2, pp. 744–765, 3 1998.
- [48] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, num. 2, pp. 41–59, 9 1996.
- [49] P. Wolniansky, G. Foschini, G. Golden, y R. Valenzuela, "V-blast: An architecture for realizing very high data rates over the rich scattering wireless channel," in *Proc. URSI ISSSE*, 9 1998, pp. 295–300.
- [50] R. Mesleh, H. Haas, S. Sinanovic, C. W. Ahn, y S. Yun, "Spatial modulation," *IEEE Trans. Veh. Tech.*, vol. 57, num. 4, pp. 2228–2241, 7 2008.
- [51] M. Matthe y otros, "Multi-user time-reversal STC-GFDMA for future wireless networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, num. 1, pp. 1–8, 5 2015.
- [52] D. Zhang, L. L. Mendes, M. Matthe, N. Michailow, y G. Fettweis, "Expectation propagation for near-optimum detection of MIMO-GFDM signals," *IEEE Trans. Wireless Commun.*, vol. PP, num. 99, 2015.
- [53] M. Matthe, I. Gaspar, D. Zhang, y G. Fettweis, "Near-ml detection for MIMO-GFDM," in *Proc. 82nd IEEE VTC Fall*, 2015.
- [54] J. Datta, H. P. Lin, y D. B. Lin, "A method to implement interference avoidance based MIMO-GFDM using spatial modulation," in *Proc. IEEE Int. Conf. on Advanced Technologies for Communications*, 2015, pp. 572–577.
- [55] P. P. Vaidya, J. M. Waykule, K. Sardesai, y A. Patil, "Software defined radio: A basic hardware realization of SDR receiver," *International Journal of Innovative Research in Engineering*, pp. 577–579, 6 2023.
- [56] S. Tooprakai, K. Kobthanyakit, T. Suwanmad, y P. Moungnoul, "Design QPSK communication system by software-defined radio (SDR)," in *2023 9th International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*. IEEE, 6 2023, pp. 18–21.

- [57] O. Etrillard, R. Gerzaguet, L. Feichter, M. Mabon, A. Courtay, y O. Berder, "LOLA SDR: Low power low latency software defined radio for broadcast audio applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, pp. 481–485, 2 2023.
- [58] Great Scott Gadgets, "Great scott gadgets HackRF One," 2024, accessed: 06/18/2024. [En línea]. Disponible: <https://greatscottgadgets.com/hackrf/one/>
- [59] L. Harshitha, K. N. D. Sindhu, C. Rekha, S. Ramya, y B. T. V. Murthy, "Implementation of noise cancellation using adaptive algorithms in GNU Radio," in *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*. IEEE, 7 2022, pp. 1–5.
- [60] F. Larroca, P. Bertrand, F. Carrau, y V. Severi, "Gr-Tempest: an open-source GNU Radio implementation of tempest," in *2022 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 12 2022, pp. 1–6.
- [61] M. P. Véstias, *Field-Programmable Gate Array*, 5ésima ed. IGI Global, 2021, pp. 257–270.
- [62] T. Soniya, I. Ragasudha, y P. N. Valli, "Routing architecture and applications of FPGA: A survey," *Journal of Physics: Conference Series*, vol. 1717, p. 012025, 1 2021.
- [63] M. L. Ch., A. B. Raj, y L. Abhikshit, "Design and implementation of a secure physical unclo-nable function in FPGA," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 7 2020, pp. 1083–1089.
- [64] J. Khazaii, *Analytical Hierarchy Process (AHP)*. Cham: Springer International Publishing, 2016, pp. 73–85. [En línea]. Disponible: https://doi.org/10.1007/978-3-319-33328-1_9
- [65] D. L. Olson, *Decision Aids for Selection Problems*. Springer New York, 1996.
- [66] T. L. Saaty, *Analytic Hierarchy Process*. Wiley, 2 2005.
- [67] E. Melese, D. Pickel, D. Soon, J. Mack, y S. L. Tighe, "Analytical hierarchy process as dust palliative selection tool," *International Journal of Pavement Engineering*, vol. 21, num. 7, pp. 908–918, 2020.
- [68] W.-C. Yang, H.-S. Kang, G.-S. Ri, y J.-S. Kim, "Consistency improvement method of pair-wise matrix based on consistency ratio decreasing rate and attribute weighting method considered decision makers levels in analytic hierarchy process: Application to hip joint prosthesis material selection," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–22, 10 2022.

- [69] L. K. Rallabandi y R. Vandrangi, "A method for improving cosine consistency index of pairwise comparison matrix in analytic hierarchy process," *International Journal of Mathematics in Operational Research*, vol. 23, p. 259, 2022.
- [70] W. Deng, Q. Ren, L. Ren, T. Kang, D. Zhang, J. Zhu, H. Tang, H. Li, y X. Chen, "A consistency index analysis and calculation method for multi-measurement data," in *2022 China International Conference on Electricity Distribution (CICED)*, 2022, pp. 1732–1736.
- [71] A. Bose, "Consistency re-evaluation in analytic hierarchy process based on simulated consistent matrices," *Journal of Multi-Criteria Decision Analysis*, vol. 29, pp. 393–401, 9 2022.
- [72] S. Pant, A. Kumar, M. Ram, Y. Klochkov, y H. K. Sharma, "Consistency indices in analytic hierarchy process: A review," *Mathematics*, vol. 10, p. 1206, 4 2022.
- [73] G. R. L. David y V. R. P. Xavier, "Memoria técnica del proyecto de titulación "Implementación de técnicas de comunicaciones digitales sobre un dispositivo FPGA"," <http://dspace.ucuenca.edu.ec/handle/123456789/25942>, 2016, (Accessed on 07/22/2024).
- [74] H. Zhou, "Design and FPGA implementation of OFDM system with channel estimation and synchronization," Master's thesis, Concordia University, 6 2013, unpublished. [En línea]. Disponible: <https://spectrum.library.concordia.ca/id/eprint/977626/>
- [75] S. G. Reddy, M. Y. Penchalaiah, y M. A. Rajani, "Design and development of OFDM baseband transceiver using virtex-6 FPGA family," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, p. 8, 8 2014.
- [76] G. Heinrich Mas, "Implementation of a GFDM scheme for a high mobility scenario on a 5G simulator including parameter optimization," PhD thesis, UPC, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, 9 2017. [En línea]. Disponible: <http://hdl.handle.net/2117/110849>
- [77] D.-I. M. Matthé, "Multiple-input multiple-output detection algorithms for generalized frequency division multiplexing," Ph.D. dissertation, Technische Universität Dresden, 3 2018. [En línea]. Disponible: <https://tud.qucosa.de/api/qucosa%3A30965/attachment/ATT-0/?L=1>
- [78] I. Gaspar, A. Navarro, N. Michailow, y G. Fettweis, "FPGA implementation of generalized frequency division multiplexing transmitter using NI LabVIEW and NI PXI platform," in

National Instruments, Virtuelle Instrumente in der Praxis (VIP 2013). Research Gate, 10 2013, p. 6.

- [79] J. Demel, C. Bockelmann, A. Dekorsy, A. Rode, S. Koslowski, y F. Jondral, “An optimized GFDM software implementation for future cloud-ran and field tests,” *Proceedings of the GNU Radio Conference*, vol. 2, num. 1, p. 9, 2017. [En línea]. Disponible: <https://pubs.gnuradio.org/index.php/grcon/article/view/23>
- [80] Advanced Micro Devices, *Lookup Table*, 2024. [En línea]. Disponible: <https://docs.amd.com/r/2021.2-English/ug1483-model-composer-sys-gen-user-guide/Lookup-Table>
- [81] ———, *Flip-Flops, Registers, and Latches*, 2024. [En línea]. Disponible: <https://docs.amd.com/r/en-US/ug901-vivado-synthesis/Advantages-of-SystemVerilog>
- [82] A. Nimr, Z. Li, M. Chafii, y G. Fettweis, “Generalized frequency division multiplexing: Unified multicarrier framework,” *Radio Access Network Slicing and Virtualization for 5G Vertical Industries*, pp. 63–82, 2021.
- [83] Analog Devices, “Ad9767,” <https://www.analog.com/en/products/AD9767.html>, Accessed: 24/06/2024.
- [84] L. Harnefors, M. Schweizer, J. Kukkola, M. Routimo, M. Hinkkanen, y X. Wang, “Generic PLL-Based Grid-Forming Control,” *IEEE Transactions on Power Electronics*, vol. 37, num. 2, pp. 1201–1204, 2022.
- [85] K. R. Sreelekshmi, K. R. Subhalekshmi, y J. Thomas, “Design of higher order phase locked loop,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, p. 4, 5 2015.
- [86] “Ant500 - great scott gadgets,” <https://greatscottgadgets.com/ant500/>, (Accessed on 07/22/2024).

Anexos

Anexo A. Desarrollo de Pruebas experimentales

La Figura A1 visualiza el escenario de pruebas para los experimentos 1 y 2, donde se puede observar a los módulos transmisor y receptor.

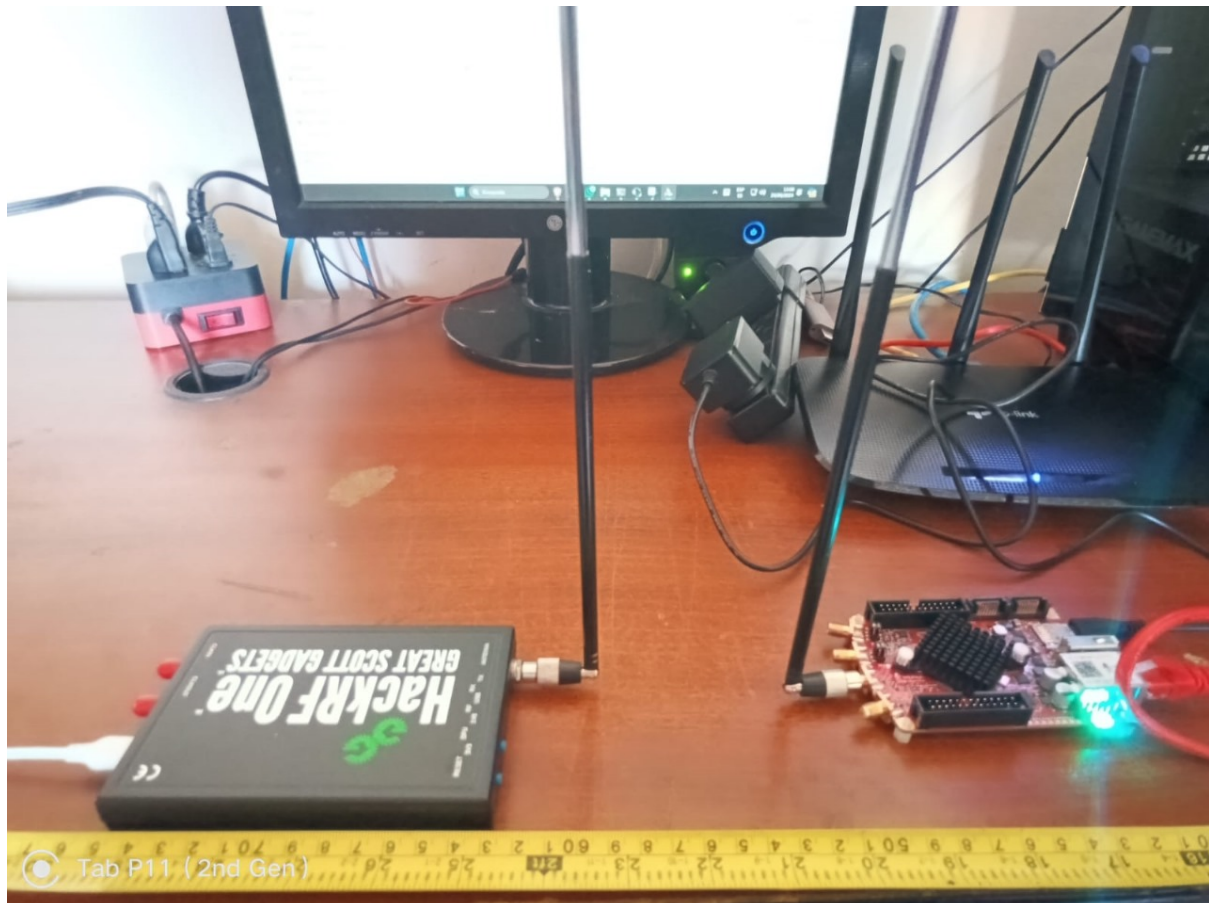


Figura A1: Ejecución de los experimentos 1 y 2

Anexo B. Implementación del sistema de transmisión Multiportadora

En el enlace al repositorio de GitHub contiene los archivos de programación para la tarjeta FPGA, y los resultados del experimento 2, estructurados de la siguiente forma:

- OFDM: Incluye los programas de descripción de hardware en VHDL para la tarjeta STEMLab 125-14, abarcando todo el proyecto para de modulación OFDM junto con los archivos IP CORE.
- GFDM: Contiene los archivos de descripción de hardware escritos en VHDL para la tarjeta STEMLab 125-14, abarcando todo el proyecto para de modulación GFDM junto

con los archivos IP CORE.

- EV_TESIS: Contiene dos subdirectorios:
 - Resultados: Contiene los resultados del experimento 2, es decir los valores de BER para modulación en sus diferentes distancias de separación.
 - Software: Posee los programas de Python y Matlab utilizados para la evaluación del sistema.

Enlace al repositorio de GitHub: [https://github.com/JhoCam007/ModulacionMultiportadoraF](https://github.com/JhoCam007/ModulacionMultiportadoraFPGA)

PGA