



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Evaluación del estándar IEEE 802.11a a nivel de la capa física utilizando  
la tarjeta FPGA-Xilinx**

AUTOR:

Alcocer Alcocer, Guido Adrián

Trabajo de Titulación previo a la obtención del Grado de  
**INGENIERO EN TELECOMUNICACIONES**

TUTOR:

M. Sc. Suarez Murillo, Efraín Oswaldo

Guayaquil, Ecuador

13 de Septiembre del 2019



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr. **Alcocer Alcocer, Guido Adrián** como requerimiento para la obtención del grado de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR

---

M. Sc. Suarez Murillo, Efraín Oswaldo

DIRECTOR DE CARRERA

---

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, a los 13 días del mes de septiembre del año 2019



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Alcocer Alcocer, Guido Adrián**

**DECLARÓ QUE:**

El trabajo de titulación “**Evaluación del estándar IEEE 802.11a a nivel de la capa física utilizando la tarjeta FPGA-Xilinx**”, previo a la obtención del título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 13 días del mes de septiembre del año 2019

EL AUTOR

---

Alcocer Alcocer, Guido Adrián



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**AUTORIZACIÓN**

Yo, **Alcocer Alcocer, Guido Adrián**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: “**Evaluación del estándar IEEE 802.11a a nivel de la capa física utilizando la tarjeta FPGA-Xilinx**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 13 días del mes de septiembre del año 2019

EL AUTOR

---

Alcocer Alcocer, Guido Adrián

# REPORTE DE URKUND

The screenshot displays the URKUND interface. On the left, a sidebar lists document details: **Documento**: Guido Alcocer\_Final.docx (D55027032); **Presentado**: 2019-08-26 05:01 (-05:00); **Presentado por**: femandopm23@hotmail.com; **Recibido**: edwin.palacios.ucsg@analysis.orkund.com; **Mensaje**: Revisión TT de Guido Alcocer [Mostrar el mensaje completo](#). The main message content shows a yellow highlight: "1% de estas 25 páginas, se componen de texto presente en 2 fuentes." On the right, a panel titled "Lista de fuentes" shows a table with columns "Categoría" and "Enlace/nombre de archivo". It lists two sources: "bibdigital.epn.edu.ec/handle/15000/7285" and "dspace.utpl.edu.ec/handle/20.500.11962/21683". Below the table are sections for "Fuentes alternativas" and "Fuentes no usadas". The bottom toolbar includes icons for navigation and actions like "1 Advertencias", "Reiniciar", "Exportar", and "Compartir".

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA: Evaluación del estándar IEEE 802.11\* a nivel de la capa física utilizando la tarjeta FPGA-Xilinx

AUTOR: Alcocer Alcocer, Guido Adrián

Trabajo de Titulación previo a la obtención del Grado de INGENIERO EN Telecomunicaciones

TUTOR: Suarez Murillo, Efraín Oswaldo

Guayaquil, Ecuador

12 de Septiembre del 2019

## Agradecimiento

Primeramente, a Dios infinitivamente porque me dio la oportunidad de vivir, por bendecirme cada día, por darme la fuerza y la fortaleza para alcanzar mis metas. A mis padres porque ellos estuvieron en los días más difíciles de mi vida, y por el apoyo que me brindaron. Al tutor de mi trabajo de titulación quien supo impartir sus conocimientos y dar las correcciones debidas para culminar este paso con éxito.

EL AUTOR

---

ALCOCER ALCOCER, GUIDO ADRIÁN

## **Dedicatoria**

Dedico este presente trabajo de titulación a mi Dios Todo Poderoso, por darme la vida, su inmenso amor y por darme las fuerzas para salir adelante a pesar de los problemas que se presentaron a lo largo de mi vida estudiantil.

A mis padres, tíos y hermana, por ser el motor fundamental en mi vida, por su comprensión, amor, consejos y apoyo incondicional en todos los momentos de mi vida, y a todos quienes compartieron conmigo el tiempo y el esfuerzo para culminar con éxito este trabajo y alcanzar el sueño anhelado.

EL AUTOR

---

ALCOCER ALCOCER, GUIDO ADRIÁN



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_

**M. Sc. ROMERO PAZ, MANUEL DE JESÚS**  
DECANO

f. \_\_\_\_\_

**M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO**  
COORDINADOR DE TITULACIÓN

f. \_\_\_\_\_

**M. Sc. ECHEVERRIA PARRA, RICARDO XAVIER**  
OPONENTE

## ÍNDICE GENERAL

Dedicatoria.....	VII
ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS.....	XIII
Resumen.....	XIV
CAPÍTULO 1: GENERALIDADES DEL PROYECTO DE GRADO.....	2
1.1.    Introducción.....	2
1.2.    Antecedentes.....	3
1.3.    Definición del problema.....	3
1.4.    Justificación del problema a investigar.....	3
1.5.    Objetivos.....	3
1.6.    Hipótesis.....	4
1.7.    Metodología de investigación.....	4
CAPÍTULO 2: MARCO TEÓRICO.....	5
2.1.    El Estándar IEEE 802.11a.....	5
2.2.    Estructura general.....	5
2.3.    El formato de trama de IEEE 802.11a.....	6
2.4.    Bloques de transmisor IEEE802.11a.....	8
2.4.1.    Codificador de datos.....	8
2.4.2.    Codificador Convolutivo.....	8
2.4.3.    Intercalación de datos.....	9
2.4.4.    Modulación de subportadora.....	9
2.5.    Diseño del receptor IEEE 802.11a.....	10
2.5.1.    Arquitectura del receptor.....	10
2.5.2.    Detección de tramas.....	11
2.5.3.    Sincronización de frecuencia gruesa.....	13
2.5.4.    Sincronización de tiempo.....	14
2.5.5.    Sincronización de frecuencia fina.....	17
2.5.6.    Estimación de canal.....	17
2.5.7.    Ecuación de canal.....	19

2.6. Simulador IEEE 802.11a en MatLab.....	20
CAPÍTULO 3: DESARROLLO DE LA PROPUESTA. ....	23
3.1. Vision general del proyecto implementado en una FPGA. ....	23
3.2. Implementación del transmisor.....	24
3.2.1. Subsistema 1: Control de procesos.....	24
3.2.2. Subsistema 2: FecAndInterleaver.....	25
3.2.3. Subsistema 3: modulación de portadora. ....	26
3.2.4. Subsistema 4: IFFT cíclico .....	28
3.2.5. Subsistema 5: Interpolación. ....	32
3.3. Implementación del receptor. ....	34
3.3.1. Subsistema 1: Convertidor descendente a banda base. ....	36
3.3.2. Subsistema 2: Decodificación de preámbulos.....	37
3.3.3. Subsistema 3: Estimación de canal.....	39
3.3.4. Subsistema 4: Ecuador de canal.....	40
3.3.5. Subsistema 5: Demodulación de subportadora .....	43
3.3.6. Subsistema 5: Decodificador Viterbi.....	44
3.4. Mediciones de desempeño.....	46
Conclusiones .....	52
Recomendaciones .....	53
Bibliografía .....	54

## ÍNDICE DE FIGURAS

### Capítulo 2:

Figura 2. 1: Formato de trama del estándar IEEE 802.11a.....	6
Figura 2. 2: Diagrama de bloques del transmisor IEEE 802.11a .....	8
Figura 2. 3: La asignación de frecuencia de las subportadoras IEEE 802.11a .....	9
Figura 2. 4: Diagrama de bloques de la arquitectura del receptor IEEE 802.11a .....	10
Figura 2. 5: Variables de decisión, (a) método de potencia, (b) método de Schmidl y Cox.....	12
Figura 2. 6: Schmidl y Cox retrasan y correlacionan algoritmo.....	13
Figura 2. 7: (a) Métrica de autocorrelación, (b) Métrica de correlación cruzada .....	15
Figura 2. 8: Rendimiento del algoritmo de sincronización de tiempo.....	16
Figura 2. 9: Instantánea de tiempo de ejecución del simulador IEEE802.11a de Matlab.....	21

### Capítulo 3:

Figura 3. 1: Modelo del generador del sistema del transmisor .....	24
Figura 3. 2: Diagrama de tiempos de la estructura de la tubería del transmisor. .....	25
Figura 3. 3: Implementación del subsistema FecAndInterleaver .....	26
Figura 3. 4: Subsistema de modulación de portadora.....	28
Figura 3. 5: Estructura Radix 22 SDF tipo mariposa para IFFT de 16 puntos. .....	30
Figura 3. 6: Implementación de FPGA de la Butterfly 1 en la etapa 1. ....	31
Figura 3. 7: Implementación compleja del multiplicador con cuatro DSP48 conectadas en cascada .....	31
Figura 3. 8: Ilustración de la ventana del símbolo OFDM .....	32

Figura 3. 9: Diagrama de bloques para la interpolación por 3/2 .....	33
Figura 3. 10: Implementación de filtro de interpolación.....	33
Figura 3. 11: Modelo del generador del sistema del receptor .....	35
Figura 3. 12: Esquema de conversión descendente de la señal recibida ....	36
Figura 3. 13: Implementación del algoritmo de retraso y correlación.....	37
Figura 3. 14: Implementación del filtro de correlación, $h_1[n]$ .....	39
Figura 3. 15: Diagrama de bloques del subsistema de canal de estimación.	40
Figura 3. 16: Subsistema de ecualización de canales. ....	41
Figura 3. 17: (a) Diagrama enrejado del decodificador Viterbi, (b) estructura de mariposa. ....	44
Figura 3. 18: Diagrama de bloques del generador del sistema del decodificador Viterbi .....	45
Figura 3. 19: Diagrama de bloques de la unidad de mariposa ACS. ....	45
Figura 3. 20: Diagrama de tiempos del acceso de memoria de rastreo.....	46
Figura 3. 21: Instantánea del programa de interfaz de usuario.....	47
Figura 3. 22: Instantánea de tiempo de ejecución de la constelación BPSK. ....	47
Figura 3. 23: Instantánea de tiempo de ejecución de la constelación QPSK. ....	48
Figura 3. 24: Instantánea de tiempo de ejecución de la constelación 16-QAM. ....	48
Figura 3. 25: Instantánea de tiempo de ejecución de la constelación 64-QAM. ....	49
Figura 3. 26: Instantánea de tiempo de ejecución de medición de la BER. .	49
Figura 3. 27: Rendimiento BER de la modulación QPSK .....	50
Figura 3. 28: Rendimiento BER de modulación 16-QAM.....	51
Figura 3. 29: Rendimiento BER de la modulación 64-QAM.....	51

## ÍNDICE DE TABLAS

### Capítulo 2:

Tabla 2. 1: Parámetros dependientes de la velocidad en el estándar IEEE 802.11a.....	7
--	---

### Capítulo 3:

Tabla 3. 1: Mapeo IQ de las modulaciones BPSK y QPSK. ....	27
Tabla 3. 2: Mapeo IQ de la modulación 16-QAM. ....	27
Tabla 3. 3: Mapeo IQ de la modulación 64-QAM. ....	27
Tabla 3. 4: Comparación de recursos de los núcleos IFFT.....	32
Tabla 3. 5: Períodos de las secuencias cortas del preámbulo .....	38

## Resumen

La multiplexación por división de frecuencia ortogonal (OFDM) es una de las técnicas de transmisión más prometedoras entre las tecnologías existentes debido a sus características únicas. Hoy en día, existe en el mercado una gran cantidad de posibilidades para implementar una red inalámbrica. El IEEE respondiendo a las necesidades del mercado y los fabricantes, comprendió la necesidad de un estándar que limitase y definiese cada uno, para que su uso fuese lo más eficiente posible. La estandarización ha permitido desvincularse de tecnologías propietarias, consiguiendo una plataforma abierta con productos de mayores prestaciones y a un precio mucho más ajustado. Este trabajo de titulación comienza con una breve descripción general de las ventajas y desventajas de los sistemas OFDM y tiene como objetivo implementar la capa física del estándar LAN inalámbrico IEEE802.11a, que es uno de los estándares IEEE basados en OFDM, en los arreglos de puerta programables de campo (FPGA). Antes de comenzar la implementación, el sistema general IEEE802.11a se ha modelado y simulado en el entorno MATLAB.

**Palabras claves:** MULTIPLEXACIÓN, FRECUENCIA, OFDM, IEEE, ESTÁNDARES, FPGA.

## **CAPÍTULO 1: GENERALIDADES DEL PROYECTO DE GRADO.**

### **1.1. Introducción.**

IEEE estandarizó los protocolos 802.11 para redes inalámbricas de área local (*Wireless Local Area Network, WLAN*) en 1997, incluidos los estándares 802.11a, 802.11b, 802.11g, 802.11n y 802.11ac1. En estos cinco estándares, 802.11b y 802.11g usan la banda de 2.4 GHz, que también es utilizada por muchos otros protocolos inalámbricos, tales como Bluetooth y ZigBee.

El protocolo 802.11n usa las bandas de frecuencias de 2.4 GHz y 5 GHz, mientras que 802.11a y 802.11ac usan solo la banda de frecuencia de 5 GHz. Con más y más aplicaciones que utilizan tecnologías inalámbricas en los últimos años, se llena en la banda de 2.4 GHz. En aplicaciones del mundo real, la velocidad de transmisión de datos no siempre alcanza la velocidad nominal en parte debido a la colisión del canal, lo que conduce a una experiencia degenerada del usuario.

Por lo tanto, los protocolos 802.11a, 802.11n y 802.11ac que utilizan la banda de 5 GHz han atraído cada vez más atención porque la interferencia potencial en la banda de 5 GHz es menor que la de la banda de 2.4 GHz. Se considera que 802.11a está desactualizado ya que la velocidad máxima de datos es de solo 54 Mbps. Como sucesor de 802.11n, 802.11ac tiene el potencial de lograr un rendimiento agregado más allá de 1 Gbps en la banda de 5 GHz.

En aplicaciones prácticas, se puede notar que el ancho de banda real del protocolo 802.11ac puede no alcanzar el ancho de banda nominal, en parte debido al impacto de la configuración del hardware y el entorno de trabajo. Este documento es para probar la velocidad de transmisión de datos real con el protocolo 802.11a en plataformas FPGA-Xilinx (en el lado de transmisión y recepción) y usarlo como referencia para aplicaciones de comunicación inalámbricas. Mientras tanto, las influencias de la distancia de transmisión y la duración de la transmisión en la velocidad de datos también se prueban y discuten.

## **1.2. Antecedentes.**

Los dispositivos FPGA son muy utilizados para el desarrollo de diferentes aplicaciones en especial se pueden comportar como transmisores, receptores, codificadores, moduladores, entre otras, para lo cual FPGA es muy robusta para evaluar redes de comunicaciones inalámbricas y alámbricas. En la carrera de telecomunicaciones se pudieron encontrar algunos trabajos donde utilizando las plataformas FPGA, tales como Altera y Xilinx. Las dos son muy utilizadas y robustas para el procesamiento de información, e inclusive se puede utilizar para el procesamiento de señales digitales e imágenes.

## **1.3. Definición del problema**

Los altos costos de equipos de comunicaciones inalámbricas no permiten que se puedan realizar trabajos investigativos mediante el uso de FPGAs. En consecuencia, surge la necesidad de la implementación y evaluación del estándar IEEE 802.11a a nivel de la capa física utilizando la plataforma FPGA-Xilinx.

## **1.4. Justificación del problema a investigar.**

Mediante la implementación de FPGA se obtendrán las métricas de rendimiento del estándar IEEE 802.11a y de la tasa de errores de bits (BER).

## **1.5. Objetivos**

### **1.5.1. Objetivo General:**

Realizar la evaluación del estándar IEEE 802.11a a nivel de la capa física utilizando la tarjeta FPGA-Xilinx

### **1.5.2. Objetivos específicos:**

- ✓ Describir la fundamentación teórica del estándar IEEE 802.11a.
- ✓ Diseñar los subsistemas de transmisión y recepción del sistema propuesto para el estándar IEEE 802.11 a mediante una FPGA-Xilinx.
- ✓ Evaluar los resultados obtenidos mediante la variación de parámetros de variables tanto del transmisor como del receptor.

## **1.6. Hipótesis**

Mediante la evaluación del estándar IEEE 802.11a al nivel de la capa MAC y que permitirá contar con una nueva herramienta didáctica para nuevas propuestas de trabajos de titulación utilizando FPGAs tanto Altera como Xilinx.

## **1.7. Metodología de investigación.**

El trabajo de titulación utiliza el método de investigación cuasiexperimental, cuyo diseño es descriptivo y explicativo y su enfoque es puramente cuantitativo. Es descriptivo, porque caracteriza los fundamentos teóricos de las comunicaciones inalámbricas a través de los estándares de IEEE y es explicativo, porque a través de las simulaciones realizadas en una FPGA se explica la propuesta del modelado.

## **CAPÍTULO 2: MARCO TEÓRICO**

### **2.1. El Estándar IEEE 802.11a.**

El estándar IEEE 802.11a, publicado por el Comité de Estándares IEEE LAN/MAN (IEEE 802.11) en 1999, es un estándar de comunicación de red de área local inalámbrica en la banda de 5 GHz. Define los requisitos para la capa física (PHY) y la capa de control de acceso medio (MAC). La capa física define cómo se transmiten los bits sin formato en un paquete a través de un enlace de comunicación y especifica las funciones de codificación y señalización que transforman los bits sin formato en ondas de radio. La capa MAC define la interfaz entre la capa física y el bus de interfaz de la máquina. En este capítulo, se explicará brevemente la capa física del estándar IEEE 802.11a.

### **2.2. Estructura general**

El IEEE802.11a es el primer estándar del comité IEEE 802.11 que utiliza la multiplexación por división de frecuencia ortogonal (OFDM) como técnica de modulación. Transmite una forma de onda analógica, convertida a partir de una señal digital, a través de las bandas de Infraestructura de Información Nacional sin Licencia (U-NII), 5.15- 5.25 GHz, 5.25-5.35 GHz y 5.725- .825 GHz. Cada banda contiene 4 canales con un ancho de banda de 20 MHz y los límites de potencia de salida de estas bandas son 40 mW, 200 mW y 800 mW, respectivamente. (Ramón A., 2014, pág. 35)

IEEE802.11a divide el canal de 20 MHz en 64 subportadoras con un espaciado de 312.5 KHz y utiliza 48 de ellas como subportadoras de datos, 4 de ellas como piloto y las otras como protección para evitar interferencia del canal adyacente. Mientras que las subportadoras piloto transmiten una secuencia de símbolos predeterminada para el seguimiento del canal, las subportadoras de datos transmiten flujo de información modulada mediante el uso de técnicas de modulación de amplitud de fase (PSK) o modulación de amplitud en cuadratura (QAM). (Herrera B. & Rodríguez L., 2018, pág. 46)

El esquema OFDM permite que el IEEE802.11a transfiera los datos sin procesar a una velocidad máxima de 54 Mbps. El estándar también es

compatible con las velocidades de datos de 6, 9, 12, 18, 24, 36 y 48 Mbps al cambiar el tipo de modulación y la velocidad de codificación de corrección de errores de reenvío (*Forward Error Correction, FEC*) de las subportadoras de datos. La duración del símbolo se especifica como 4  $\mu$ s en el estándar y 800 ns se utilizan para prefijo cíclico para garantizar una recepción libre de ISI de los símbolos transmitidos a través de un canal con un retardo extendido hasta 250 ns.

### 2.3. El formato de trama de IEEE 802.11a.

Cada trama en la capa física de IEEE 802.11a incluye preámbulos del procedimiento de convergencia de capa física (PLCP), encabezado PLCP y unidad de datos de servicio de capa física (PSDU), bits de cola y almohadilla, como se muestra en la figura 2.1. El preámbulo PLCP consta de 10 preámbulos cortos y dos preámbulos largos. Los preámbulos cortos se utilizan para la detección de cuadros, el control automático de ganancia y la sincronización de temporización. La frecuencia establecida y la respuesta del canal también se estima a través de los preámbulos largos que se envían inmediatamente después del preámbulo corto. (Estrella F., 2017, pág. 60)

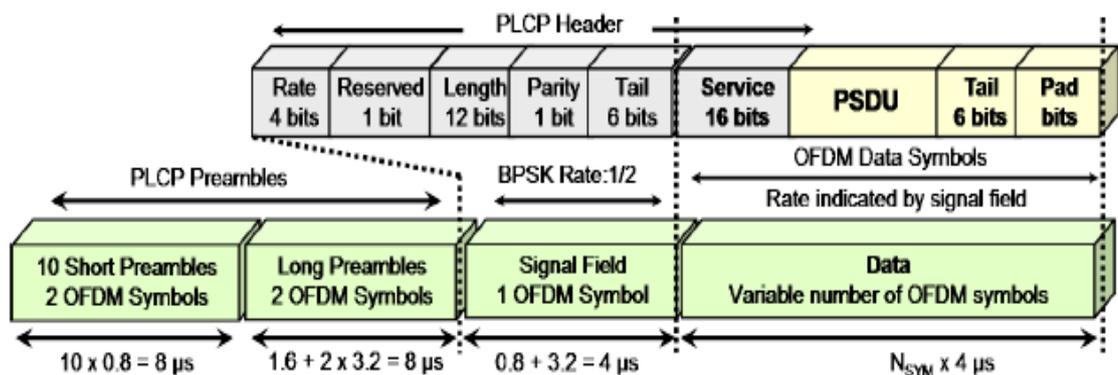


Figura 2. 1: Formato de trama del estándar IEEE 802.11a

Fuente: (Estrella F., 2017, pág. 61)

La longitud de los preámbulos cortos es de 0.8  $\mu$ s, lo que equivale a una quinta parte de un período de símbolo regular. Se generan tomando la IFFT de la siguiente secuencia,  $S_{26:26}$ , que produce una señal periódica con un período de 16 muestras. Al repetir la señal producida dos veces después de agregar el prefijo cíclico, se generan diez símbolos de preámbulo cortos idénticos.

$$\begin{aligned}
S_{-26:26} = \sqrt{13/6} & (0, 0, 1 + j, 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, -1 - j, 0, 0, 0, \\
& -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, 0, 0, 0, -1 - j, 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, \\
& 0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, 0) \quad (2.1)
\end{aligned}$$

La secuencia de preámbulo larga se compone de un prefijo cíclico y dos símbolos de preámbulo largos idénticos. A diferencia de los otros símbolos OFDM, la longitud del prefijo cíclico para esta secuencia es igual a 32 muestras. La razón de esto es el uso de los preámbulos largos para la estimación de desplazamiento de frecuencia fina evitando la discontinuidad entre los símbolos consecutivos. Las 52 subportadoras se utilizan durante la generación de los preámbulos largos y están modulados por los elementos de la siguiente secuencia. (Baños, Afaqui, Lopez, & Garcia, 2017, pág. 39)

$$\begin{aligned}
L_{-26:26} = (1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, \\
1, 1, 1, 0, 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, \\
-1, 1, -1, 1, 1, 1, 1) \quad (2.2)
\end{aligned}$$

El primer símbolo después de los símbolos de entrenamiento largos se denomina campo de señal, que se transmite utilizando la modulación BPSK y la velocidad de codificación en 1/2, y contiene los parámetros de velocidad y longitud. La información sobre el tipo de modulación y la velocidad FEC que se utiliza en el resto de la trama se transmite a través del parámetro de velocidad de 4 bits y el parámetro de longitud indica el número de bytes de información en la PSDU, tal como se muestra en la tabla 2.1.

Tabla 2. 1: Parámetros dependientes de la velocidad en el estándar IEEE 802.11a

velocidad	velocidad de datos	Modulación	Codificación de velocidad (R)	Bits codificados por símbolo	Bits de datos por símbolo
1101	6 Mbps	BPSK	½	48	24
1111	9 Mbps	BPSK	¾	48	36
0101	12 Mbps	QPSK	½	96	48
0111	18 Mbps	QPSK	¾	96	72
1001	24 Mbps	16QAM	½	192	96
1011	36 Mbps	16QAM	¾	192	144
0001	48 Mbps	64QAM	2/3	288	192
0011	54 Mbps	64QAM	¾	288	216

Fuente: (Baños et al., 2017, pág. 42)

El campo de servicio, transmitido después del campo de señal, contiene 16 bits y se utiliza para sincronizar el descifrador de datos en el receptor. Luego, la trama OFDM transporta la carga útil de PSDU que envía la capa MAC. Los seis bits de cola cero siguen a la PSDU para forzar el decodificador Viterbi en el receptor a estado cero. Finalmente, el final de la trama se llena con los bits de la almohadilla para que el número de bits en los símbolos de datos se convierta en un múltiplo de los bits codificados en un símbolo OFDM.

## 2.4. Bloques de transmisor IEEE802.11a

El estándar IEEE802.11a especifica solo la parte de transmisión de la capa física y proporciona los requisitos de rendimiento para el receptor. Esto permite a diferentes fabricantes desarrollar sus propias soluciones de receptor que son compatibles entre sí. La cadena de transmisión especificada por el estándar se ilustra en la figura 2.2. Los bloques transmisores en esta cadena se explican brevemente en las siguientes subsecciones.

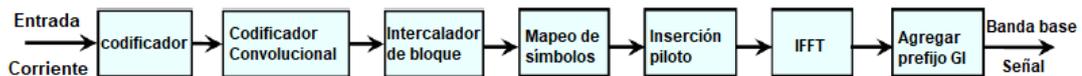


Figura 2. 2: Diagrama de bloques del transmisor IEEE 802.11a

Fuente: (Estrella F., 2017, pág. 68)

### 2.4.1. Codificador de datos.

El transmisor IEEE 802.11a utiliza un aleatorizado de datos que utiliza una secuencia binaria pseudoaleatoria (PRBS) para aleatorizar todos los bits de información, excepto el campo de señal, para no transmitir una secuencia larga de unos o ceros. El aleatorizado utiliza el polinomio  $S(x) = x^7 + x^4 + 1$  que genera una secuencia cíclica de longitud 127 y el estado inicial del aleatorizado se establece aleatoriamente al comienzo de la transmisión. El receptor estima el estado inicial del codificador observando los primeros siete bits del campo de servicio.

### 2.4.2. Codificador Convolutional

Para lograr una transferencia de datos confiable, todos los bits de información en la trama, incluido el campo de señal, están codificados con un codificador convolutional. El estándar IEEE 802.11a utiliza los polinomios del

generador estándar de la industria,  $g_0 = 1338$  y  $g_1 = 1718$  para producir dos bits de salida para cada bit de entrada, y admite las tasas de codificación 1/2, 2/3 y 3/4 al pinchar los datos antes de la transmisión.

### 2.4.3. Intercalación de datos

La técnica de intercalado de bloques se utiliza en el estándar IEEE 802.11a para mejorar el rendimiento de los códigos de corrección de errores de reenvío. Todos los bits en la salida del codificador convolucional están intercalados por un intercalador de bloques y el tamaño del bloque de intercaladores está determinado por el número de bits codificados por símbolo OFDM, NCBPS. El entrelazado consta de dos pasos de permutación y en el primer paso de permutación, los bits codificados adyacentes se asignan a subportadoras no adyacentes. En el segundo paso de permutación, el índice de bits de los bits codificados consecutivos en la constelación se cambia continuamente para evitar los largos recorridos de los bits LSB.

### 2.4.4. Modulación de subportadora.

En el transmisor IEEE 802.11a, un IFFT de 64 puntos multiplexa las subportadoras ortogonales y las subportadoras se reenumeran como en la figura 2.3 antes de realizar la transformación de Fourier. Solo 48 de ellos se utilizan para la transmisión de datos y se modulan utilizando BPSK, QPSK, 16-QAM o 64-QAM de acuerdo con el parámetro de velocidad (Rate). Las subportadoras  $P_{-21}, P_{-7}, P_7$  y  $P_{21}$  están dedicadas a señales piloto de tipo peine que se utilizan para rastrear las variaciones de fase debido al canal variable en el tiempo o un error de desplazamiento de frecuencia. Las subportadoras piloto se modulan utilizando BPSK y, para evitar la generación de líneas espectrales, transmiten una secuencia binaria pseudoaleatoria generada por el mismo polinomio utilizado en el aleatorizado.

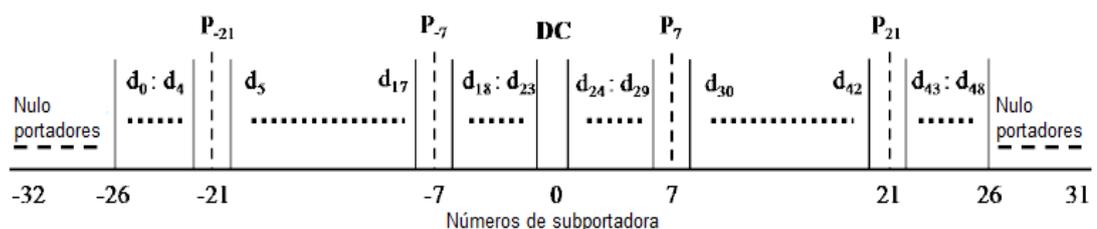


Figura 2. 3: La asignación de frecuencia de las subportadoras IEEE 802.11a  
Fuente: (Estrella F., 2017, pág. 85)

## 2.5. Diseño del receptor IEEE 802.11a.

Como se mencionó en la sección 2.1, el estándar IEEE 802.11a no especifica la estructura del receptor. Por esta razón, antes de implementar el transceptor IEEE 802.11a en la FPGA, primero se debe diseñar el receptor en un entorno MATLAB. Además, también se debe desarrollar un simulador para observar el rendimiento del receptor y garantizar que nuestra solución de receptor sea compatible con el transmisor especificado en el estándar. En esta sección, los algoritmos que se utilizan en el receptor y el simulador MATLAB se explicarán en detalle.

### 2.5.1. Arquitectura del receptor

La figura 2.4 muestra la arquitectura del receptor de banda base para el estándar IEEE 802.11a. El receptor obtiene la señal entrante de banda base de un convertidor analógico/digital (*Analog to Digital Converter, ADC*) con una frecuencia de muestreo de 20 MHz. Luego, básicamente realiza las operaciones del transmisor (véase la figura 2.2) en orden inverso para reconstruir la secuencia transmitida. A diferencia del transmisor, el receptor también incluye algunos bloques de sincronización para demodular la señal recibida correctamente. (Salazar Soler, 2016, pág. 50)

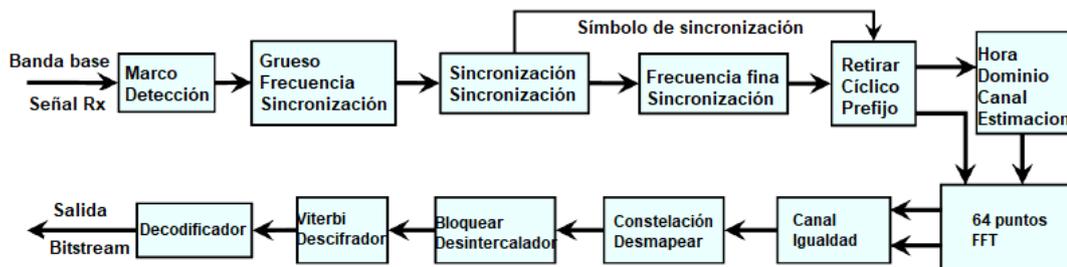


Figura 2. 4: Diagrama de bloques de la arquitectura del receptor IEEE 802.11a  
Fuente: (Salazar Soler, 2016, pág. 52)

Como primer paso en la cadena del receptor, se detecta una trama entrante buscando los preámbulos cortos en la señal recibida. Luego, el receptor estima aproximadamente el desplazamiento de frecuencia de la portadora y encuentra el límite del símbolo OFDM mediante el uso de los preámbulos cortos. A continuación, el error de desplazamiento de frecuencia se corrige de manera más precisa con la ayuda de los preámbulos largos y las muestras recibidas se abren en una ventana para eliminar el prefijo cíclico.

Además, los coeficientes de respuesta al impulso del canal se estiman utilizando los símbolos largos del preámbulo.

Después de tomar la FFT de las muestras recibidas en ventana y la respuesta al impulso del canal, se realiza una ecualización del canal del dominio de frecuencia para implementar una demodulación coherente. Luego, los símbolos complejos recibidos en la constelación se transforman en símbolos de bits y la secuencia de estos símbolos se reorganiza mediante el desintercalado de bloques, que es la operación inversa de la intercalación en el transmisor. Finalmente, los códigos de corrección de errores se decodifican usando el algoritmo de decodificación Viterbi de decisión dura o suave y el bloque descodificador recupera el flujo de bits original.

En las siguientes subsecciones, se explicarán en detalle los algoritmos de sincronización, estimación de canales y ecualización utilizados en el receptor. Los otros bloques mostrados en la figura 2.4, como la decodificación de Viterbi, el mapeo de constelaciones y los bloques de desintercalado, se analizarán en la sección 3.2, que proporciona los detalles de implementación del receptor.

### **2.5.2. Detección de tramas.**

La detección de tramas es la tarea de decidir si hay o no una trama entrante y dar una estimación aproximada del tiempo de inicio de la trama. Esta tarea puede describirse como un problema de prueba de hipótesis al comparar una variable de decisión  $\mu$  con un umbral predefinido,  $Thr$ . Si la variable de decisión supera el umbral, indica la presencia de la trama.

El algoritmo más conocido para encontrar el límite de inicio de la trama entrante es medir la potencia de la señal recibida. Este algoritmo forma una variable de decisión  $\mu$  al tomar la relación de la potencia de la señal recibida dentro de las dos ventanas consecutivas, tal como se expresa en la ecuación. (2.3) Cuando no hay una trama transmitida, las muestras de señal recibidas,  $r[n]$ , consisten solo en ruido y la potencia recibida es igual a la potencia de ruido. Sin embargo, durante la transmisión de una trama, la potencia de

entrada en el receptor es igual a la suma del ruido y la potencia de la señal. Entonces, como se muestra en la figura 2.5a, la variable de decisión  $\mu$  crea un pico en el límite inicial de la trama.

$$\mu(n) = \frac{\sum_{i=0}^{D-1} |r[n-i]|^2}{\sum_{i=0}^{D-1} |r[n-i-D]|^2} \quad (2.3)$$

Donde D es igual a la longitud de un símbolo de preámbulo corto.

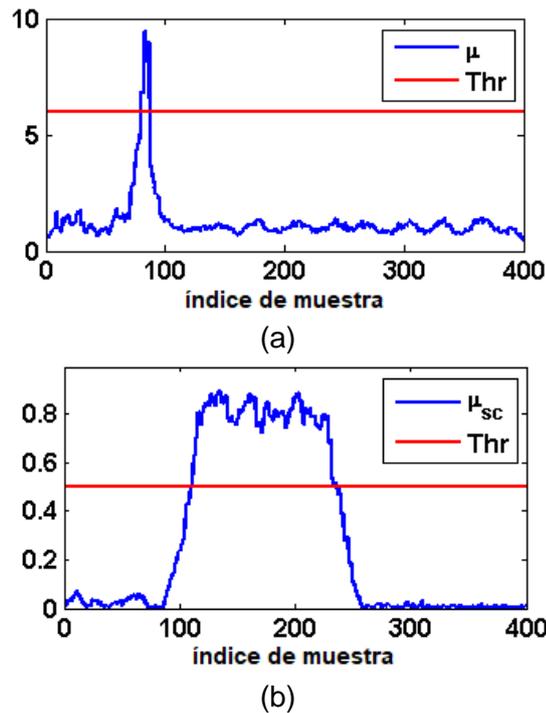


Figura 2. 5: Variables de decisión, (a) método de potencia, (b) método de Schmidl y Cox

Fuente: (Chen & Yang, 2012. pág. 26)

El método de potencia es un algoritmo eficiente si el receptor no tiene conocimiento a priori de los preámbulos en la trama recibida. Sin embargo, el estándar IEEE 802.11a utiliza una secuencia predeterminada como un preámbulo corto y repite esta secuencia 10 veces al comienzo de cada trama. Para aprovechar la periodicidad de los preámbulos, se utiliza el algoritmo "delay and correlate" propuesto por Schmidl y Cox. (Chen & Yang, 2012, pág.27)

Del mismo modo que el método de potencia, el algoritmo de Schmidl y Cox también utiliza dos ventanas deslizantes para generar la variable de

decisión. La primera ventana,  $P$  en la figura 2.6, mide la correlación entre la señal recibida y su versión retardada. Para obtener una variable de decisión independiente del nivel de señal, la segunda ventana calcula la potencia de la señal recibida dentro de la ventana de correlación y normaliza la variable de decisión. La figura 2.5b muestra este método crea una meseta a lo largo de la duración de los preámbulos cortos, por lo que el inicio de tramas puede detectarse de manera sólida al comparar la variable de decisión,  $\mu_{sc}$ , con un umbral predefinido.

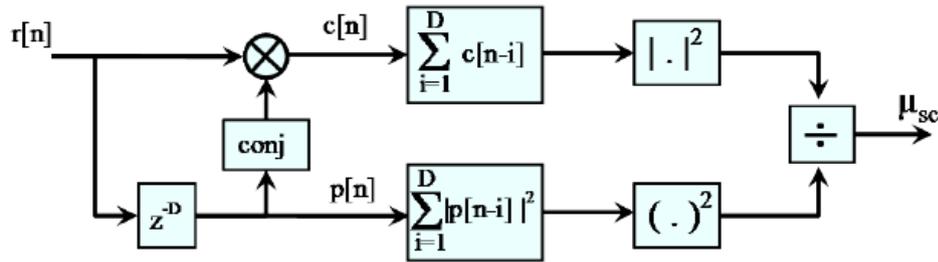


Figura 2. 6: Diagrama de bloques de correlación de señal recibida y retardada.

Fuente: (Chen & Yang, 2012, pág. 35)

### 2.5.3. Sincronización de frecuencia gruesa

Antes de realizar la sincronización de temporización que utiliza la correlación cruzada entre la señal recibida y la señal transmitida original, el desplazamiento de frecuencia portadora entre el transmisor y el receptor debe corregirse aproximadamente. Para lo cual se debe utilizar el algoritmo propuesto para realizar la sincronización de frecuencia. De acuerdo con este algoritmo, el desplazamiento de frecuencia, denotado por  $\delta f$ , se puede estimar encontrando la fase de la correlación entre dos símbolos de entrenamiento consecutivos.

Si está presente un desplazamiento de frecuencia, las muestras recibidas pueden escribirse en términos de la señal transmitida,  $t[n]$ , y el ruido,  $n[n]$  como en la ecuación (2.4):

$$r[n] = t[n]e^{j2\pi\delta f n/f_s} + n[n] \quad (2.4)$$

Entonces, la correlación compleja,  $C_S[n]$ , entre dos preámbulos cortos consecutivos se escribe como en la ecuación (2.5), donde  $N_S$  es igual a 16, que es la longitud del preámbulo corto.

$$\begin{aligned}
C_S[n] &= \sum_{k=0}^{N_S-1} r[n-k] \times r^*[n-k-N_S] \\
&= \sum_{k=0}^{N_S-1} t[n-k] e^{j2\pi\delta f(n-k)/f_s} \times t^*[n-k-N_S] e^{j2\pi\delta f(k+N_S-n)/f_s} + \text{noise} \\
&= e^{j2\pi\delta f N_S/f_s} \sum_{k=0}^{N_S-1} t[n-k] \times t^*[n-k-N_S] + \text{noise term} \quad (2.5)
\end{aligned}$$

Debido a la periodicidad en los preámbulos cortos, la señal transmitida,  $t[n-k]$ , es igual a  $t[n-k-N_S]$  a través de la ventana de correlación. Y así, la ecuación (2.5) se puede simplificar como

$$C_S[n] = e^{j2\pi\delta f N_S/f_s} \sum_{k=0}^{N_S-1} |t[n-k]|^2 + \text{noise term} \quad (2.6)$$

Entonces, el desplazamiento de frecuencia de la portadora puede estimarse fácilmente a partir de la fase de  $C_S[n]$ .

$$\widehat{\delta f_c} = \frac{f_s \angle C_S[n]}{2\pi N_S} \quad (2.7)$$

Una estimación aproximada del desplazamiento de frecuencia es suficiente para el proceso de sincronización de temporización. Entonces se realiza la operación de cuantificación, definida en la ecuación. (2.8), en el  $\widehat{\delta f_c}$ . Esta cuantización proporciona simplicidad en la implementación al considerar solo unas pocas frecuencias en lugar del rango completo de frecuencias.

$$\lfloor \widehat{\delta f_c} \rfloor = \text{sign}(\widehat{\delta f_c}) \cdot \left\lfloor \frac{128 \cdot |\widehat{\delta f_c}|}{f_s} + 0.5 \right\rfloor \cdot \frac{f_s}{128} \quad (2.8)$$

Donde el operador  $\lfloor x \rfloor$  denota el entero más grande no mayor que  $x$ . Después de estimar el  $\widehat{\delta f_c}$  en términos generales, el desplazamiento de frecuencia estimado se elimina fácilmente de la señal recibida multiplicándolo con una señal exponencial compleja cuya frecuencia es igual al negativo de la frecuencia estimada.

#### 2.5.4. Sincronización de tiempo

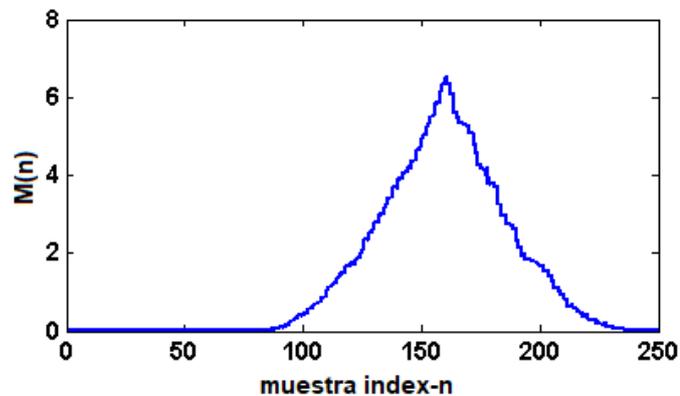
Después de realizar el proceso de sincronización de frecuencia aproximada, la siguiente tarea del receptor es la sincronización de tiempo que

encuentra el punto de inicio de los símbolos OFDM. Esta tarea es uno de los problemas más críticos en el diseño del receptor porque un error de sincronización de tiempo puede causar un ISI significativo que degrada el rendimiento del receptor directamente.

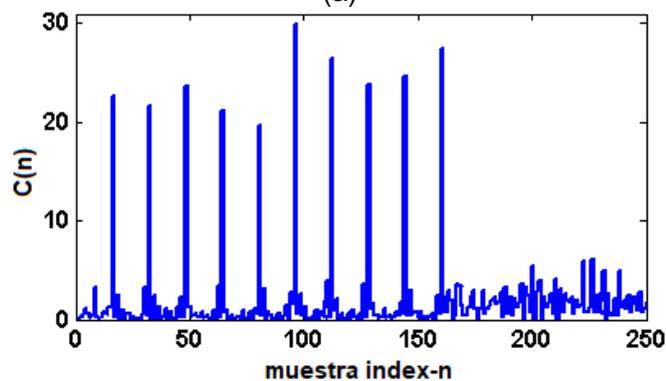
Para encontrar el punto de inicio del símbolo con precisión, se realiza la sincronización de tiempo en dos pasos. En el primer paso, la métrica de autocorrelación,  $M[n]$  en la ecuación (2.9), se calcula utilizando el algoritmo de retraso y correlación. La longitud del bloque de correlación y retraso se selecciona como la mitad de la duración total de los preámbulos cortos, de modo que la métrica  $M[n]$  crea un pico al final del último preámbulo corto (véase la figura 2.7a). Por lo tanto, se obtiene fácilmente una estimación aproximada del desplazamiento de tiempo utilizando la ecuación (2.10).

$$M[n] = \left| \sum_{k=0}^{79} r[n-k]r^*[n-k-80] \right|^2 \quad (2.9)$$

$$\hat{n} = \arg \max_n M[n] \quad (2.10)$$



(a)



(b)

Figura 2. 7: (a) Métrica de autocorrelación, (b) Métrica de correlación cruzada

Fuente: (Chen & Yang, 2012, pág. 38)

La estimación aproximada,  $\hat{n}$ , puede ser ligeramente anterior o posterior al tiempo exacto debido a la existencia de ruido en la señal recibida. Por lo tanto, se realiza el segundo paso para mejorar la precisión de la estimación. En este paso, se calcula una métrica de correlación cruzada,  $C_{[n]}$ , correlacionando la señal recibida con el preámbulo corto original, SP, y luego se calcula el desplazamiento del tiempo de inicio residual de los preámbulos cortos usando la ecuación. (2.11)

$$C[n] = \left| \sum_{k=0}^{15} r[n-k]SP^*[15-k] \right|^2$$

$$\hat{m} = \arg \max_{0 \leq m \leq 15} \sum_{k=0}^7 C[m+16k]$$

Finalmente, el desplazamiento de tiempo exacto se calcula con la ayuda de  $\hat{n}$  y  $\hat{m}$  realizando la regla de decisión definida por la ecuación (2.12)

$$n_{off} = \begin{cases} (\hat{n}/16) \times 16 + \hat{m}, & \text{if } |(\hat{n} \bmod 16) - \hat{m}| \leq 8 \\ (\hat{n}/16 + 1) \times 16 + \hat{m}, & \text{if } ((\hat{n} \bmod 16) - \hat{m}) > 8 \\ (\hat{n}/16 - 1) \times 16 + \hat{m}, & \text{if } ((\hat{n} \bmod 16) - \hat{m}) < -8 \end{cases}$$

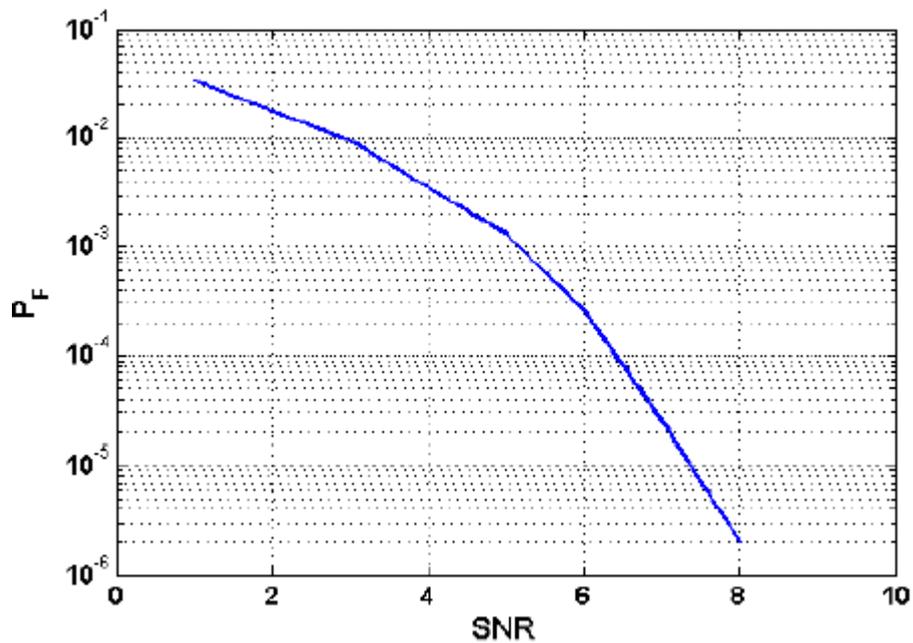


Figura 2. 8: Rendimiento del algoritmo de sincronización de tiempo  
Elaborado por: Autor.

Donde el operador / se utiliza para divisiones enteras. El rendimiento del algoritmo de sincronización de temporización también se simula en MATLAB bajo el ruido blanco gaussiano aditivo y la probabilidad de falla de sincronización versus SNR y que se ilustra en la figura 2.8.

### 2.5.5. Sincronización de frecuencia fina.

En esta subsección el receptor debe corregir el desplazamiento de frecuencia de la portadora debido a la falta de coincidencia del oscilador local o un desplazamiento Doppler para eliminar el ICI. Por lo tanto, se utilizan los preámbulos largos de la misma manera para obtener una estimación más precisa del desplazamiento de frecuencia. Si la función de correlación compleja, definida en la ecuación (2.3) se calcula para los preámbulos largos cuya longitud es 64, da un resultado cuatro veces más preciso que los cortos. Luego, el desplazamiento de frecuencia se estima a partir de la fase del resultado de correlación,  $C_L[n]$ , de la siguiente manera (ecuación 2.13)

$$\widehat{\delta f_L} = \frac{f_s \angle C_L[n]}{2\pi 64}$$

Como resultado, el desplazamiento de frecuencia se puede eliminar de la señal recibida multiplicándola con  $\exp(-j2\pi\widehat{\delta f_L}n/f_s)$  y la estimación final se puede escribir combinando las estimaciones gruesas y finas de la siguiente manera (ecuación 2.14):

$$\widehat{\delta f} = \left[ \widehat{\delta f_c} \right] + \widehat{\delta f_L}$$

La ecuación (2.13) muestra que el desplazamiento de frecuencia máximo estimado está limitado por la región inequívoca de la fase  $C_L[n]$ . Esto significa que este algoritmo puede compensar un desplazamiento de frecuencia máximo de  $|\delta f_{max}|L = f_s/128 = 156:25$  KHz si se usan los únicos preámbulos largos. Desde el uso de los preámbulos cortos para la estimación aproximada, este límite se extiende hasta  $|\delta f_{max}| = f_s/2N_S = 625$  KHz.

### 2.5.6. Estimación de canal

En un canal de comunicación inalámbrico, la señal transmitida llega a la antena del receptor a través de múltiples rutas con diferentes retardos y

ganancias. Por lo tanto, el receptor debe estimar la respuesta del canal para demodular coherentemente los símbolos recibidos. El estándar IEEE 802.11a supone que el canal inalámbrico no cambia durante el período de transmisión de un paquete. Transmite dos símbolos largos de preámbulo al comienzo de cada trama para que el receptor pueda obtener los coeficientes de respuesta al impulso del canal (ecuación 2.15).

$$h[n] = \sum_{i=0}^{L-1} h_i \cdot \delta[n - i] = \sum_{i=0}^{L-1} a_i \exp(j\theta_i) \delta[n - i]$$

Para analizar el efecto del canal de múltiples rutas en las muestras recibidas, se considera un modelo de canal de múltiples rutas simple con L toques como en la ecuación (2.15). La señal recibida resultante se puede escribir de la siguiente manera (ecuación 2.16):

$$y[n] = x[n] \star h[n] + w[n] = \sum_{i=0}^{L-1} h_i \cdot x[n - i] + w[n]$$

donde  $x[n]$  es la señal transmitida en la ecuación (2.2) y  $w[n]$  es un ruido gaussiano blanco de banda base complejo. Usando la periodicidad de la señal transmitida durante los preámbulos largos, la convolución en el dominio del tiempo en la ecuación (2.16) puede expresarse mediante una multiplicación matricial para los símbolos de preámbulo como el siguiente (ecuación 2.17):

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{62} \\ y_{63} \end{bmatrix} = \begin{bmatrix} x_0 & x_{63} & \cdots & x_{63-L+2} \\ x_1 & x_0 & \cdots & x_{63-L+3} \\ \vdots & \vdots & \ddots & \vdots \\ x_{62} & x_{61} & \cdots & x_{63-L} \\ x_{63} & x_{62} & \cdots & x_{63-L+1} \end{bmatrix} \cdot \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{L-2} \\ h_{L-1} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{62} \\ w_{63} \end{bmatrix}$$

$$Y = X \cdot H + W$$

Desde este punto de vista, el vector de respuesta al impulso del canal, H, puede estimarse utilizando la técnica de estimación de mínimos cuadrados (LS) de la siguiente manera (ecuación 2.18):

$$\hat{H} = X^H \cdot (X \cdot X^H)^{-1} \cdot Y = X^+ \cdot Y$$

donde  $X^+$  denota el inverso generalizado de Moore Penrose de  $X$ . La extensión máxima de retardo del canal inalámbrico debe ser menor que la longitud del intervalo CP para una recepción libre de ISI, por lo que la longitud del vector de respuesta al impulso del canal en la ecuación (2.17) se selecciona como igual a la longitud del intervalo CP, 16. Además, las muestras recibidas del primer y segundo símbolos de preámbulo largos se pueden promediar para reducir la variación de ruido en los coeficientes de canal estimados y la ecuación (2.18) puede reescribirse como sigue:

$$\hat{H} = \frac{1}{2} X^+ \cdot (Y_1 + Y_2)$$

### 2.5.7. Ecuación de canal

Después de estimar la respuesta al impulso del canal, el receptor debe eliminar los efectos del canal de la señal recibida. Esta es la tarea del bloque de ecualización que normaliza todas las subportadoras con su función de transferencia de canal estimada. Para analizar el efecto del canal de múltiples rutas en cada subportadora, se escribe la FFT de la señal recibida en la ecuación (2.16) y queda (ecuación 2.20):

$$\begin{aligned} Y[k] &= \sum_{n=0}^{N-1} y[n] \exp\left(-j2\pi \frac{kn}{N}\right) \\ &= \sum_{i=0}^{p-1} a_i e^{j\theta_i} \sum_{n=0}^{N-1} x[n-i] \exp\left(-j2\pi \frac{kn}{N}\right) + W[k] \\ &= \sum_{i=0}^{p-1} a_i e^{j\theta_i} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X[k-i] \exp\left(j2\pi \frac{k(n-i)}{N}\right) \exp\left(-j2\pi \frac{kn}{N}\right) + W[k] \\ &= X[k] \sum_{i=0}^{p-1} a_i \exp\left(j\left(\theta_i - 2\pi \frac{ki}{N}\right)\right) + W[k] \\ &= X[k] \cdot H[k] + W[k] \end{aligned}$$

Como resultado de la ecuación (2.20), los efectos del canal en cada subportadora pueden compensarse si el receptor conoce la función de transferencia de canales para todos los índices de la subportadora. Este conocimiento se puede obtener fácilmente tomando la FFT de los coeficientes

de respuesta al impulso del canal que se encuentran en la ecuación. (3.18) Luego, la ecualización del canal se puede realizar usando un ecualizador de dominio de frecuencia de un toque de la siguiente manera

$$\widehat{X}[k] = \frac{Y[k]}{\widehat{H}[k]}$$

donde  $\widehat{H}[k]$  es la función de transferencia de canal estimada en la  $k$ ' subportadora. Además de la ecualización de cada subportadora, el receptor también debe seguir la fase de la portadora mientras recibe los símbolos de datos en un paquete. Debido al hecho de que el procedimiento de sincronización de frecuencia no es un proceso perfecto, habrá un pequeño error de frecuencia residual en la señal ecualizada. Este error de frecuencia provoca una rotación de fase en la constelación recibida y degrada significativamente el rendimiento del sistema. La rotación de fase debido al error de frecuencia residual es la misma para todas las subportadoras y la cantidad de rotación de fase en el  $n$ 'ésimo símbolo de datos OFDM puede estimarse utilizando las cuatro subportadoras piloto como sigue:

$$\widehat{\Phi}_n = \angle \left[ \sum_{k=-21,-7,7,21} \frac{R_{n,k}}{P_{n,k}} \right]$$

donde  $R_{n,k}$  es la subportadora piloto ecualizada recibida y  $P_{n,k}$  es la subportadora piloto transmitida del  $n$ 'ésimo símbolo de datos OFDM. Después de estimar el valor de fase acumulado, los puntos de constelación ecualizados de cada símbolo de datos OFDM se derivan multiplicándolos con  $\exp(-j\widehat{\Phi}_n)$ .

## 2.6. Simulador IEEE 802.11a en MatLab.

Para observar el rendimiento de los algoritmos del receptor, se modela primero el transmisor IEEE802.11a en MATLAB. El estándar IEEE 802.11a da un ejemplo de codificación de una trama para la capa física. Se aplica el mensaje transmitido definido en este ejemplo a la entrada de nuestro modelo de transmisor y se confirma la validez de nuestro modelo comparando su salida con la forma de onda del dominio del tiempo obtenida en el estándar

para este ejemplo. Después de modelar el transmisor, se diseña el simulador que se muestra en la figura 3.6 para probar nuestra solución de receptor bajo un canal multitrayecto con ruido gaussiano blanco aditivo y estadísticas de desvanecimiento de Rayleigh. A través de este simulador, el rendimiento del receptor se puede observar en varias condiciones de canal y para diferentes parámetros del transmisor y el receptor.

El primer parámetro en la interfaz gráfica de usuario (GUI) está etiquetado como “1. Channel Type” y el modelo de canal entre el transmisor y el receptor se pueden ajustar como AWGN o canal de desvanecimiento dispersivo en tiempo de ejecución a través del menú emergente justo al lado de esta etiqueta. Si se selecciona el modelo de canal de desvanecimiento dispersivo, el número de taps (tomas) de retardo multitrayecto es determinado por el segundo parámetro y la respuesta de impulso del canal instantáneo se puede observar en la figura 2.9d.

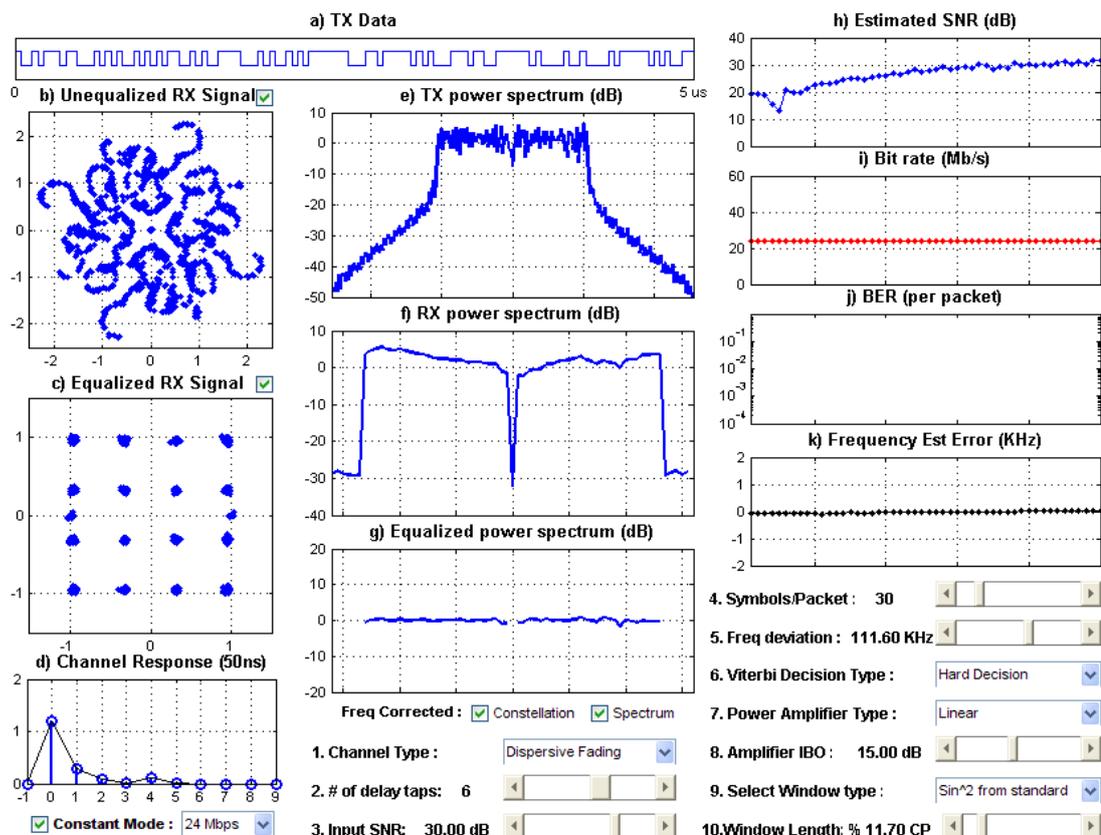


Figura 2. 9: Instantánea de tiempo de ejecución del simulador IEEE802.11a de Matlab

Fuente: (Salazar Soler, 2016, pág. 80)

Además, la relación señal/ruido (*Signal to Noise Ratio, SNR*) en la entrada del receptor se puede configurar usando la barra deslizante etiquetada como "3. Input SNR:" y el simulador informa la SNR estimada de la señal recibida ecualizada, tal como se muestra en la figura 2.9.h. El quinto parámetro en la GUI crea una desviación entre los osciladores locales del transmisor y el receptor y el rendimiento del algoritmo de estimación de frecuencia se muestra en la figura 2.9.k.

El simulador permite al usuario seleccionar el modelo de amplificador de potencia como Modelo Saleh lineal, de recorte o no lineal y ajustar el factor de retroceso de entrada del amplificador. Además, el simulador también muestra el espectro de potencia de la señal transmitida en la figura 2.9.e para demostrar las distorsiones en el espectro causadas por el amplificador de potencia no lineal.

Para suprimir las radiaciones fuera de banda, la técnica de ventanas propuesta en el estándar se puede aplicar a la señal transmitida suavizando las transiciones entre los símbolos OFDM. Los últimos parámetros en la GUI seleccionan la función de ventanas y ajustan la duración del suavizado.

## CAPÍTULO 3: DESARROLLO DE LA PROPUESTA.

### 3.1. Vision general del proyecto implementado en una FPGA.

En el presente trabajo se realiza la implementación del transceptor IEEE 802.11a en la plataforma de desarrollo de radio definida por software (*Software Defined Radio, SDR*) de factor de forma pequeño (SFF) de Lyrtech. La ruta de recepción del módulo RF de Lyrtech convierte la señal recibida a una frecuencia intermedia a 30 MHz. Esto significa que la señal IEEE802.11a recibida en la entrada del convertidor analógico a digital (*Analog to Digital Converter, ADC*) tiene una frecuencia máxima de 40 MHz y, por lo tanto, la frecuencia mínima de muestreo debe ser de 80 MHz de acuerdo con los criterios de Shannon-Nyquist.

Si la señal recibida se muestrea a 80 MHz, se requiere un filtro de paso bajo muy agudo en la etapa de conversión de banda base para suprimir las imágenes especulares de la señal digital. Por esta razón, la frecuencia de muestreo del ADC y la frecuencia de reloj del sistema de la FPGA se seleccionaron como 120 MHz.

Se utiliza la herramienta System Generator de Xilinx para desarrollar la implementación FPGA del transceptor. Esta herramienta facilita el diseño de hardware FPGA para el procesamiento de señales digitales (Digital Signal Processing, DSP) y se extiende a la herramienta de simulación Simulink de MatLab para proporcionar un entorno de desarrollo de alto nivel para FPGA Xilinx. Primero se diseñan los modelos de sistema del transmisor y receptor en el entorno Simulink utilizando la biblioteca Xilinx que incluye los modelos de precisión de bits para los bloques de circuitos en la FPGA. Luego, se obtuvieron los códigos HDL para que los modelos sean creados automáticamente por System Generator de Xilinx.

En este capítulo, se explicará en detalle la implementación del generador del sistema de las arquitecturas de transmisor y receptor según el estándar IEEE802.11 a. Además, el rendimiento de la tasa de error de bit medido (BER)

del receptor bajo un canal de ruido blanco gaussiano aditivo (AWGN) se presentará al final del capítulo.

### 3.2. Implementación del transmisor

La implementación del generador del sistema del transmisor se divide en cinco subsistemas como se muestra en la figura 3.1. El primer subsistema se denomina control de proceso (ProcessControl), está diseñado para controlar el flujo de señal desde la capa MAC hacia el DAC. El segundo subsistema (FecAndInterleaver) se utiliza para implementar las operaciones basadas en bits, es decir, la codificación convolucional y el intercalado.

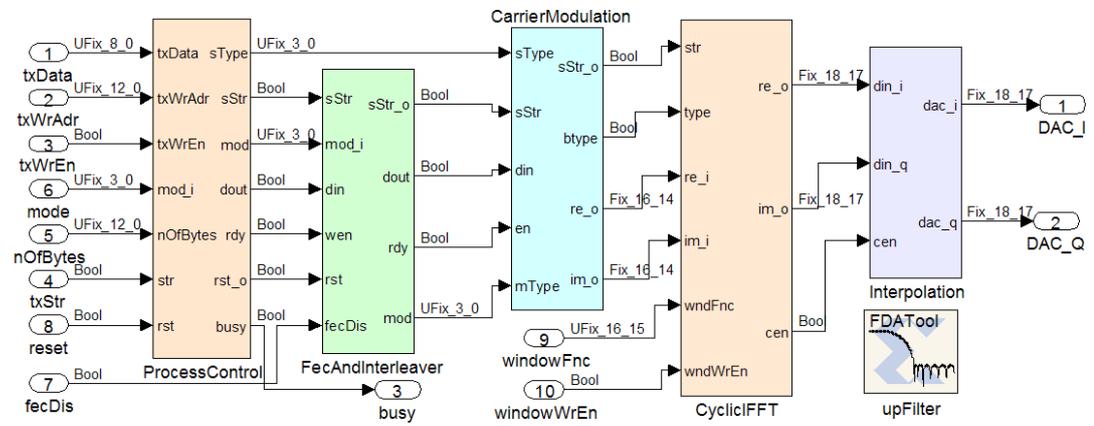


Figura 3. 1: Modelo del generador del sistema del transmisor  
Elaborado por: Autor.

Luego, el subsistema de modulación de portadora (CarrierModulation) convierte los bits intercalados en los símbolos de constelación compleja y el subsistema CyclicIFFT toma las transformadas inversas de Fourier de estos símbolos para generar la señal de banda base en el dominio del tiempo. Finalmente, el bloque de interpolación (Interpolation) sincroniza la salida IFFT con el DAC que funciona a 120 MHz. En las siguientes subsecciones, se explicará en detalle la implementación de cada subsistema.

#### 3.2.1. Subsistema 1: Control de procesos.

La interfaz de la capa física con la capa MAC es proporcionada por el subsistema de control de procesos. La capa MAC ajusta la velocidad de datos y el número de bytes en la trama, que se transmitirán, a través de las entradas "mode" y "nOfBytes", respectivamente. Después de eso, comienza la

transmisión conduciendo la entrada "txStr" a nivel lógico alto y escribe los bytes para la trama actual en el búfer de entrada del transmisor.

Después de disparar desde la entrada "txStr ", el subsistema de control de procesos calcula el número de símbolos OFDM requeridos para el marco actual y genera las señales de inicio de símbolo, "sStr" y de tipo de símbolo, "sType ", para controlar el flujo de señal a través de todos los subsistemas en el transmisor, Además, este bloque también produce el campo de señal y los bits de datos de OFDM codificados al leer los bytes en el búfer de entrada.

En la figura 3.2 se muestra el bloque de control de proceso que produce los bits sin procesar para solo un símbolo OFDM por 480 ciclos de reloj y los otros subsistemas procesan estos bits de forma canalizada para producir las señales en cuadratura y en fase de la banda base.

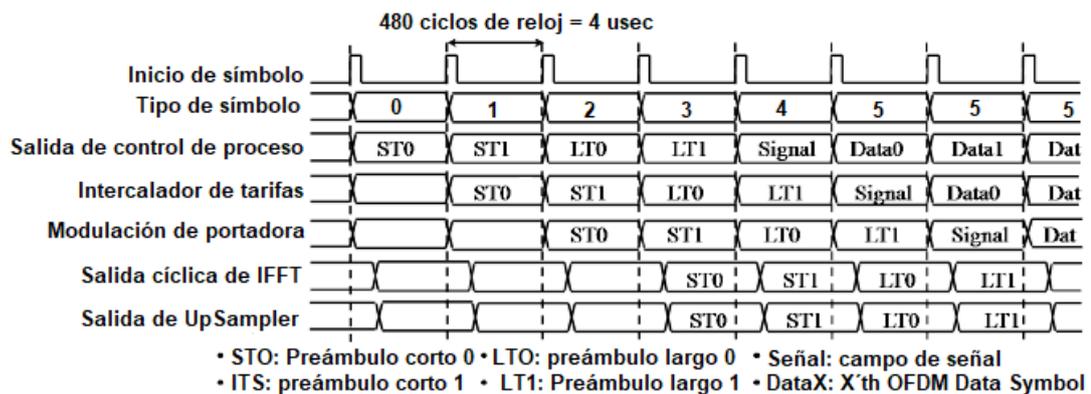


Figura 3. 2: Diagrama de tiempos de la estructura de la tubería del transmisor.  
Elaborado por: Autor.

### 3.2.2. Subsistema 2: FecAndInterleaver

La figura 3.3 muestra la arquitectura de implementación del subsistema FecAndInterleaver (la codificación convolucional y el intercalado). En primer lugar, los bits entrantes del subsistema de control de procesos están codificados por el bloque ConvEncoder que produce la salida de dos bits por cada entrada de un bit. Luego, los bits producidos se almacenan en una memoria de bloque de doble puerto. En el siguiente intervalo de canalización, los bits almacenados del símbolo OFDM anterior se leen en un orden determinado por las permutaciones del intercalador y los códigos de perforación definidos en el estándar IEEE802. 11a.



Tabla 3. 1: Mapeo IQ de las modulaciones BPSK y QPSK.

Modulación BPSK			Modulación QPSK			
Bit: b0	En-fase (In-Phase)	Cuadratura (Quadrature)	Bit: b0	En-fase (In-Phase)	Bit: b1	Cuadratura (Quadrature)
0	-1	0	0	$-1/\sqrt{2}$	0	$-1/\sqrt{2}$
1	+1	0	1	$+1/\sqrt{2}$	1	$+1/\sqrt{2}$

Elaborado por: Autor.

Tabla 3. 2: Mapeo IQ de la modulación 16-QAM.

Bits (b0: b1)	En-fase (In-Phase)	Bits (b2: b3)	Cuadratura (Quadrature)
00	$-3/\sqrt{10}$	00	$-3/\sqrt{10}$
01	$-1/\sqrt{10}$	01	$-1/\sqrt{10}$
11	$+1/\sqrt{10}$	11	$+1/\sqrt{10}$
10	$+3/\sqrt{10}$	10	$+3/\sqrt{10}$

Elaborado por: Autor.

Tabla 3. 3: Mapeo IQ de la modulación 64-QAM.

Bits (b0: b1: b2)	En-fase (In-Phase)	Bits (b3: b4: b5)	Cuadratura (Quadrature)
000	$-7/\sqrt{42}$	000	$-7/\sqrt{42}$
001	$-5/\sqrt{42}$	001	$-5/\sqrt{42}$
011	$-3/\sqrt{42}$	011	$-3/\sqrt{42}$
010	$-1/\sqrt{42}$	010	$-1/\sqrt{42}$
110	$+1/\sqrt{42}$	110	$+1/\sqrt{42}$
111	$+3/\sqrt{42}$	111	$+3/\sqrt{42}$
101	$+5/\sqrt{42}$	101	$+5/\sqrt{42}$
100	$+7/\sqrt{42}$	100	$+7/\sqrt{42}$

Elaborado por: Autor.

En la figura 3.4 se muestra el subsistema de modulación de portadora donde almacena los símbolos de constelación generados en un búfer. En el siguiente intervalo de proceso, los símbolos almacenados se leen en un orden

desplazado, tal como se describió en la sección 2.3.4, y las subportadoras piloto se insertan en las posiciones apropiadas. Para interpolar la señal transmitida, el bloque de modulación de portadora también extiende la secuencia de entrada de la transformada de Fourier a 256 puntos por relleno de ceros. Además, al comienzo de cada trama, este subsistema envía las secuencias de preámbulo almacenadas en una ROM al bloque IFFT multiplexando la salida.

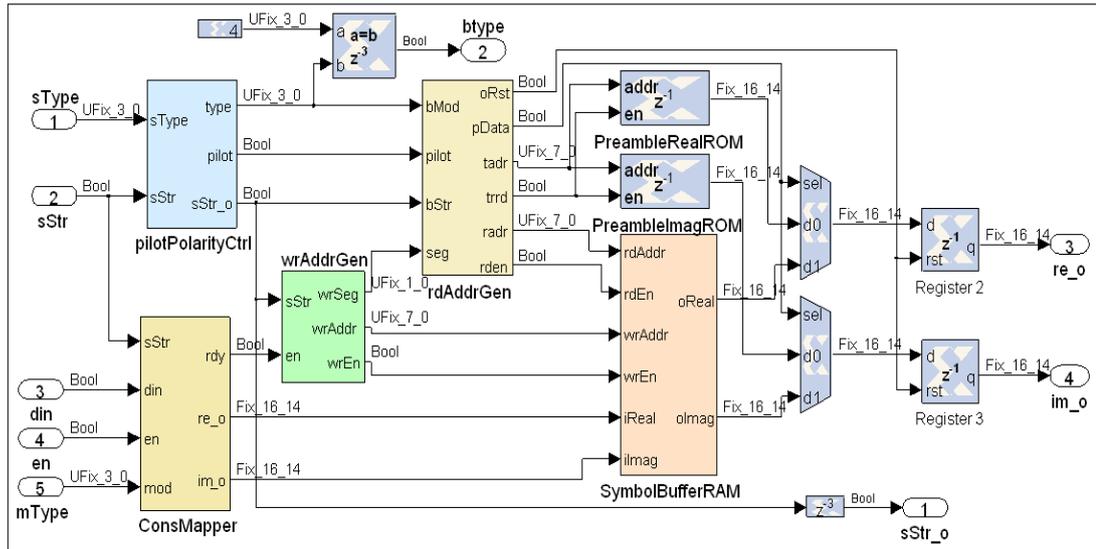


Figura 3. 4: Subsistema de modulación de portadora  
Elaborado por: Autor.

### 3.2.4. Subsistema 4: IFFT cíclico

El subsistema cyclicIFFT toma la IFFT de los símbolos de la subportadora del dominio de la frecuencia para generar la señal de banda base del dominio del tiempo. El tamaño de IFFT se selecciona como 256 para interpolar la señal de salida cuatro veces. Para la implementación FPGA del IFFT de 256 puntos de manera canalizada, se utiliza el algoritmo de retardo único de retroalimentación (*Single Delay Feedback, SDF*) Radix-2<sup>2</sup>.

La complejidad de este algoritmo en términos del número de multiplicaciones complejas es la misma que el algoritmo Radix-4 IFFT, pero conserva la estructura de mariposa y del algoritmo radix-2 para reducir el número de adiciones. Para una operación IFFT de punto  $N$ , el algoritmo SDF Radix 2<sup>2</sup> utiliza el multiplicador complejo  $4 \log_4 N - 1$ , la adición compleja de  $4 \log_4 N$  y la memoria de datos complejos  $N - 1$ .

$$x[n] = \sum_{k=0}^{N-1} X[k] \cdot W_N^{kn} \quad 0 \leq n < N \quad (3.1)$$

El algoritmo *Radix* – 2<sup>2</sup> SDF diezma las muestras de salida de IFFT de punto N, definidas como en la ecuación anterior, por un factor de cuatro. Luego, combina los factores de rotación común y simplifica la ecuación de la siguiente manera.

$$\begin{aligned} x[4n + 2n_2 + n_1] &= \sum_{k=0}^{N/4-1} \sum_{k_1=0}^1 \sum_{k_2=0}^1 X[k + k_1 \frac{N}{2} + k_2 \frac{N}{4}] \cdot W_N^{(4n+2n_2+n_1)(k+k_1 \frac{N}{2} + k_2 \frac{N}{4})} \\ &= \sum_{k=0}^{N/4-1} \sum_{k_2=0}^1 B[k + k_2 N/4, n_1] \cdot W_N^{(k+k_2 N/4)n_1} \cdot W_{N/2}^{(n_2+2n)(k+k_2 N/4)} \\ &= \sum_{k=0}^{N/4-1} (B[k, n_1] + j^{2n_2+n_1} B[k + N/4, n_1]) \cdot W_N^{k(2n_2+n_1)} \cdot W_{N/4}^{kn} \quad (3.2) \end{aligned}$$

Donde,

$$B[k, n_1] = X[k] + (-1)^{n_1} X[k + N/2] \text{ and } n_1 = \{0,1\}, n_2 = \{0,1\}$$

De acuerdo con la ecuación (3.2), una IFFT de N puntos de una secuencia puede calcularse fácilmente utilizando una IFFT de N/4 puntos después de realizar las operaciones de multiplicación compleja y de mariposa, tal como se ilustra en la figura 3.5. Desde este punto de vista, se puede tomar una IFFT de 256 puntos en cuatro etapas, donde cada etapa involucra dos estructuras de mariposa y una multiplicación compleja. La multiplicación en la última etapa es trivial, por lo que se puede eliminar y solo se necesitan tres multiplicaciones complejas para realizar la transformación inversa de Fourier.

La figura 3.6 muestra la implementación del generador del sistema de la primera mariposa en la primera etapa. La arquitectura de mariposa necesita las primeras 128 muestras para calcular las salidas, por lo que se utiliza un bloque de memoria de acceso aleatorio (*Random Access Memory, RAM*) para proporcionar un retardo de 128 muestras. Mientras recibe la última mitad de

la secuencia de entrada, la implementación agrega la muestra entrante con retardo y envía el resultado a la salida.

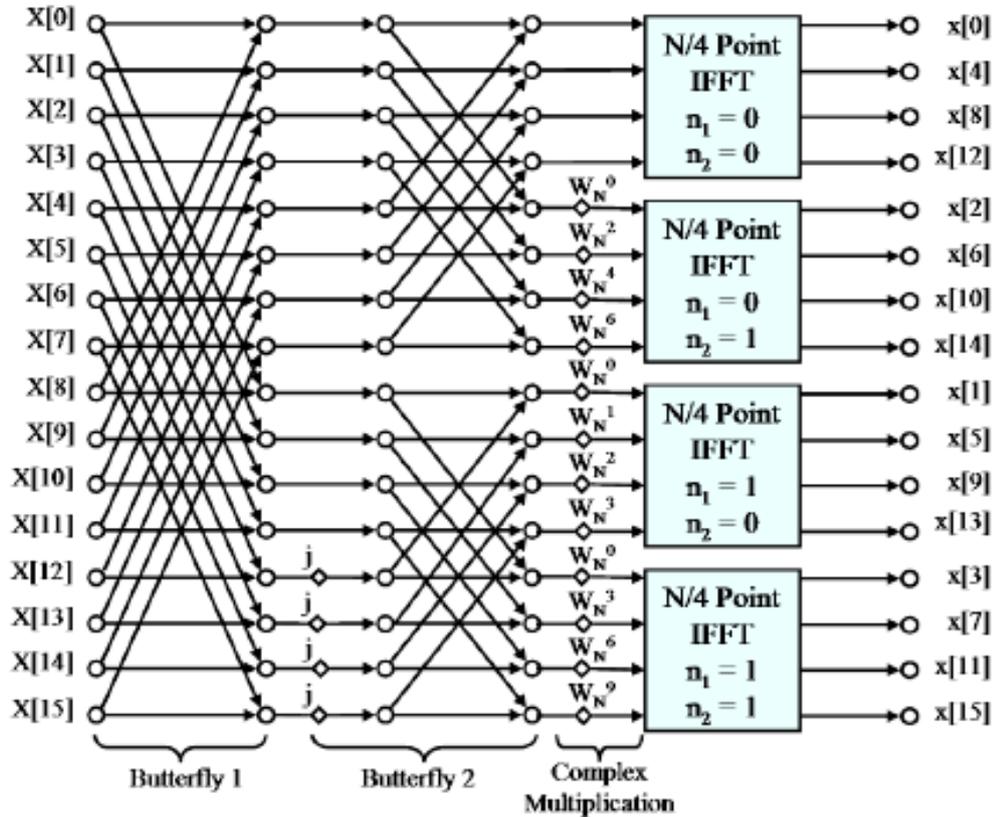


Figura 3. 5: Estructura Radix  $2^2$  SDF tipo mariposa para IFFT de 16 puntos.  
Elaborado por: Autor.

Al mismo tiempo, también resta las muestras de entrada del retardo y escribe el resultado en el búfer de retardo. En el siguiente intervalo de procesamiento, los resultados de la resta se leen desde el búfer y se dirigen a la salida mientras se almacena en búfer la primera mitad de la siguiente secuencia. El proceso pipeline se lleva a cabo de esa manera.

La implementación de la segunda mariposa es similar a la primera, excepto la multiplicación trivial por  $j$ . La salida de la segunda mariposa se multiplica con los factores de rotación en una ROM por un multiplicador complejo. En la figura 3.7 se muestra el multiplicador complejo que se implementa en cascada con cuatro DSP48 para realizar cuatro multiplicaciones reales y dos adiciones sin usar ningún recurso externo.

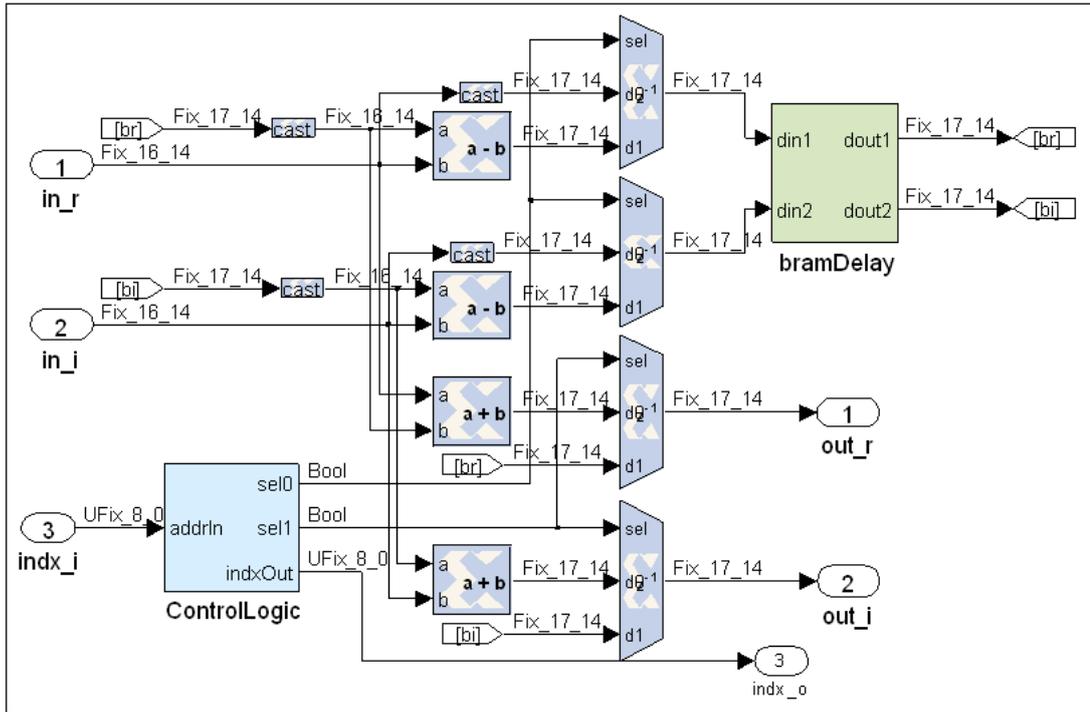


Figura 3. 6: Implementación de FPGA de la Butterfly 1 en la etapa 1.  
Elaborado por: Autor.

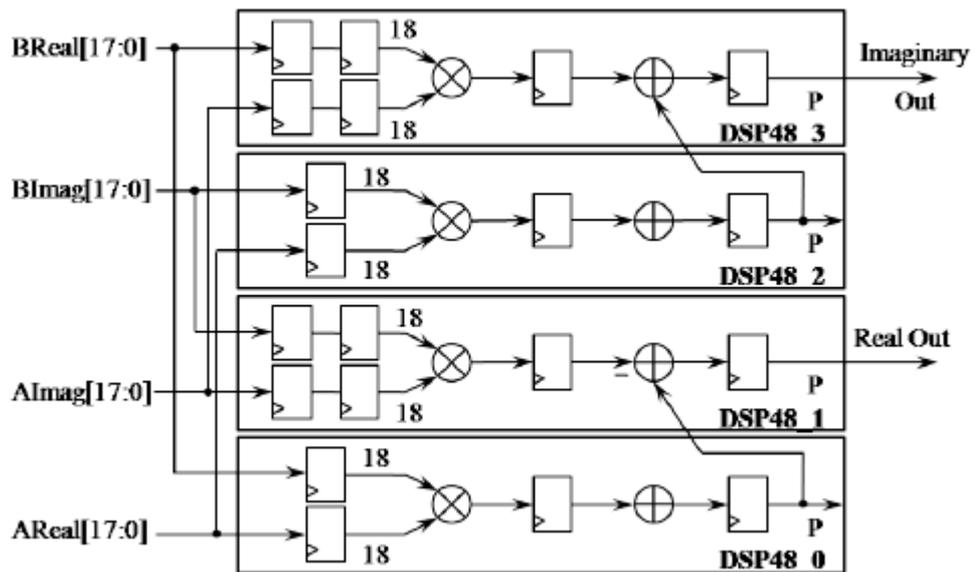


Figura 3. 7: Implementación compleja del multiplicador con cuatro DSP48  
conectadas en cascada  
Elaborado por: Autor.

Las muestras de dominio de tiempo se obtienen en un orden inverso de bits a partir de la salida de la última etapa de IFFT, por lo que las muestras de salida del bloque IFFT se almacenan en una RAM para su reordenamiento. Después de completar la transformada de Fourier, las muestras calculadas se

leen desde la memoria intermedia de una manera cíclica y se multiplican con una función de ventana, tal como se muestra en la figura 3.8, para suprimir las radiaciones fuera de banda. La función de ventana se almacena en una memoria de bloque y la capa MAC puede actualizarla para cambiar la duración de suavizado,  $T_W$ .

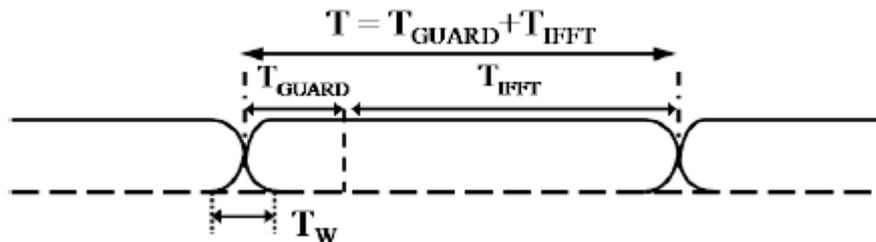


Figura 3. 8: Ilustración de la ventana del símbolo OFDM  
Elaborado por: Autor.

Además de estos, el núcleo de IFFT diseñado también se comparó con el núcleo de Xilinx IFFT disponible en términos de utilización de recursos de FPGA, tal como se muestra en la tabla 3.4. Por lo tanto, nuestra implementación utiliza los segmentos FPGA un 29% más eficientemente que el núcleo Xilinx.

Tabla 3. 4: Comparación de recursos de los núcleos IFFT

Recursos	Núcleo Xilinx IFFT	Núcleo IFFT propuesto
Segmentos (Slice)	1897	1343
Reloj Flip-Flop	2847	1716
Tabla de búsqueda (LUT)	2797	2306
Bloque de RAM	3	3
Multiplicador DSP48	12	12

Elaborado por: Autor.

### 3.2.5. Subsistema 5: Interpolación.

La velocidad de salida del subsistema CyclicIFFT es igual a 80 Msps, mientras que la frecuencia de funcionamiento del reloj del DAC es de 120 MHz. Por esta razón, el subsistema de interpolación está diseñado para interpolar las muestras recibidas del bloque CyclicIFFT por un factor de 3/2. Para realizar la interpolación en 3/2, la señal de entrada se muestrea primero insertando dos ceros entre las muestras de entrada.

Luego, la señal producida es filtrada por un filtro de paso bajo y disminuida por un factor de dos, tal como se muestra en la figura 3.9. Sin embargo, no es necesario implementar todos los pasos de interpolación, solo se puede considerar el cálculo de las muestras utilizadas por el muestreador inferior.

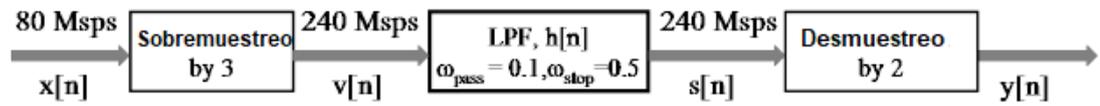


Figura 3. 9: Diagrama de bloques para la interpolación por 3/2  
Elaborado por: Autor.

Si se supone que el bloque de muestreo descendente selecciona las muestras pares de la señal de entrada, la salida de la interpolación,  $y[n]$ , puede expresarse como

$$y[n] = s[2n] = v[2n] \star h[2n] = \sum_{k=0}^{L-1} h[k] \cdot v[2n - k] \quad (3.3)$$

La ecuación (3.3) también se puede simplificar como sigue considerando solo las muestras distintas de cero de  $v[n]$ .

$$y[3n + l] = \begin{cases} \sum_{k=0}^{L/3-1} h[3k] \cdot x[2n - k] & \text{cuando } L=0 \\ \sum_{k=0}^{L/3-1} h[3k + 2] \cdot x[2n - k], & \text{cuando } L=1 \\ \sum_{k=0}^{L/3-1} h[3k + 1] \cdot x[2n + 1 - k], & \text{de otra manera} \end{cases} \quad (3.4)$$

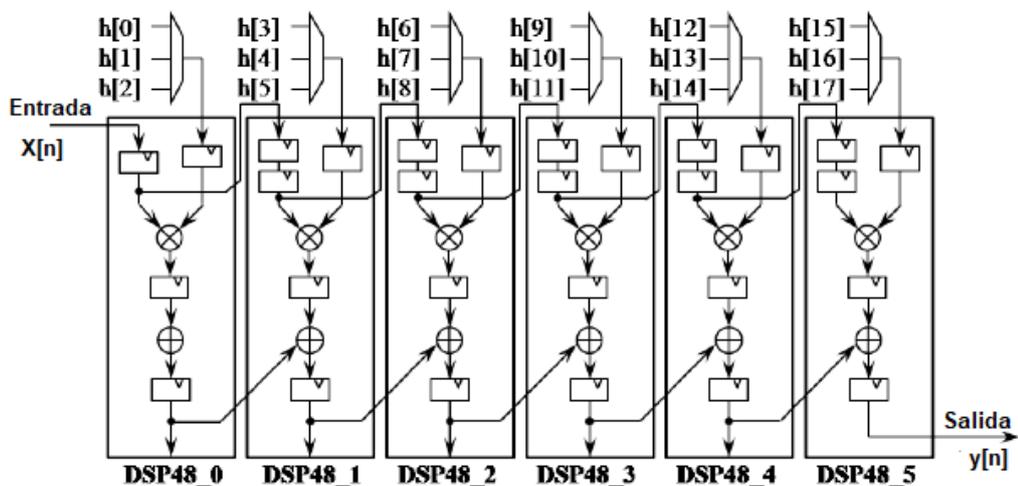


Figura 3. 10: Implementación de filtro de interpolación  
Elaborado por: Autor.

Como resultado de la ecuación (3.4), el filtro FIR de interpolación diseñado con 18 coeficientes se implementa utilizando un filtro FIR de sexto orden. Los coeficientes del filtro se cambian de acuerdo con el índice de tiempo de interpolación y la salida del filtro se calcula en cascada en seis segmentos DSP48 como se ilustra en la figura 3.10.

### 3.3. Implementación del receptor.

La implementación del generador del sistema del receptor se divide en nueve subsistemas, tal como se muestra en la figura 3.11. El primer subsistema, denominado "DownConverter", convierte la señal IF recibida del ADC en componentes I y Q de banda base por un detector de cuadratura y diezma la salida en un factor de seis para producir una frecuencia de muestreo de salida de 20 MHz. Luego, los procesos de detección de tramas, temporización y sincronización de frecuencias se llevan a cabo por el subsistema "preambleDecode" tal como se describió en la sección 2.4.

A continuación, el subsistema "chanEstimation" estima los coeficientes de respuesta al impulso del canal en el dominio del tiempo y elimina el prefijo cíclico de las muestras recibidas. Los 64 puntos FFT de las muestras restantes y los coeficientes de respuesta al impulso del canal son tomados por el subsistema "FFT64", que se implementa de manera similar a la explicada en la subsección 2.5.4. Después de obtener los símbolos de subportadora del dominio de frecuencia, el subsistema "channelEqualizer" realiza los algoritmos de ecualización de canal y del seguimiento de fase portadora descritos en la Subsección 2.2.6.

Luego, el subsistema "subCarrierDemod" calcula las relaciones de probabilidad de registro (*LikeLikelihood Ratios, LLR*) de los bits recibidos demodulando los símbolos de constelación ecualizados. Además, también reorganiza el orden de los LLR mediante una operación inversa del intercalado en el transmisor y los envía al subsistema "viterbiDecoder" para decodificar el flujo de bits que ha sido codificado convolucionalmente.

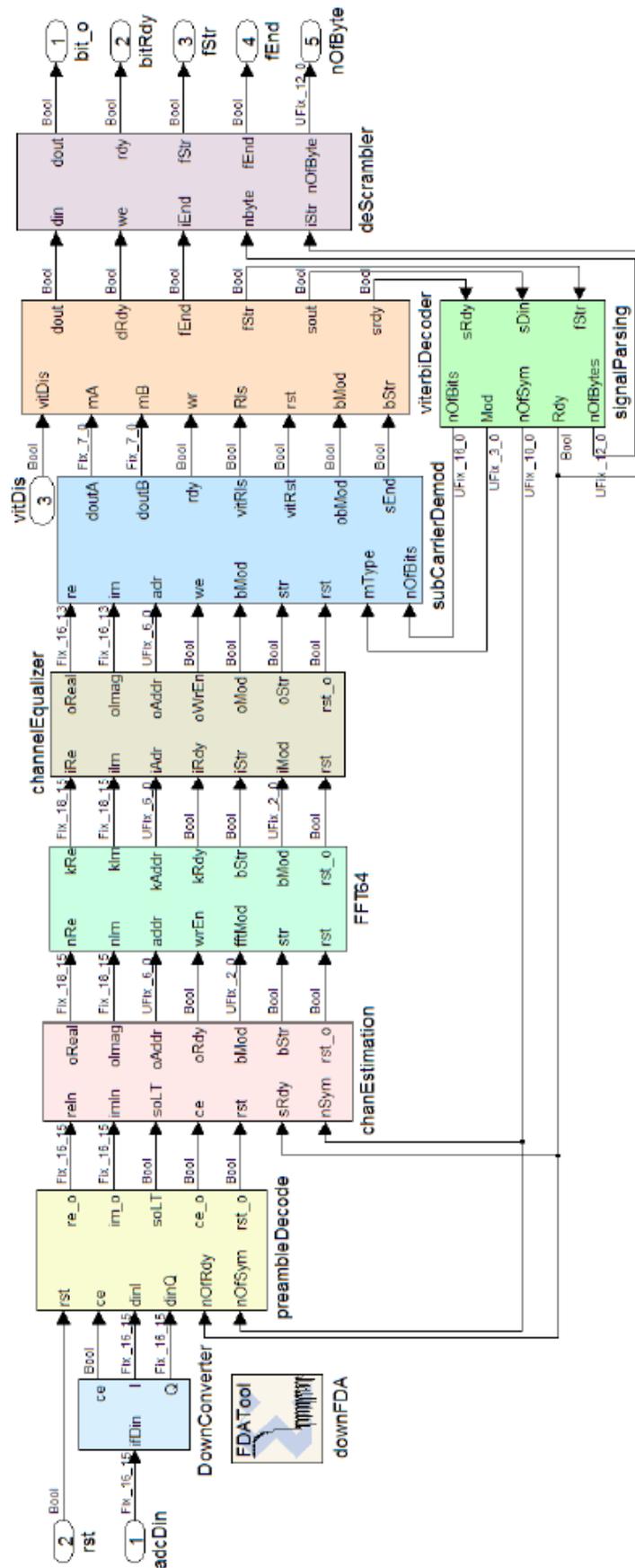


Figura 3. 11: Modelo del generador del sistema del receptor  
Elaborado por: Autor.

Finalmente, los bits decodificados, a excepción del campo de señal, se envían al subsistema “deScrambler” para reconstruir el mensaje transmitido. Los bits decodificados del campo de señal también se envían al bloque “signalParsing” para extraer los parámetros de trama como tipo de modulación, número de bytes transmitidos y número de símbolos OFDM en la trama. En las siguientes subsecciones, se explicará en detalle la implementación de los subsistemas del receptor que son diferentes a las contrapartes del transmisor.

### 3.3.1. Subsistema 1: Convertidor descendente a banda base.

La conversión descendente de la señal IF recibida del ADC en los componentes I&Q de banda base se ilustra en la figura 3.12. La frecuencia portadora de la señal recibida es de 30 MHz y la frecuencia del reloj del sistema es igual a 120 MHz, por lo que los componentes coseno y seno del oscilador local pueden generarse fácilmente repitiendo la secuencia {1,0,-1,0}.

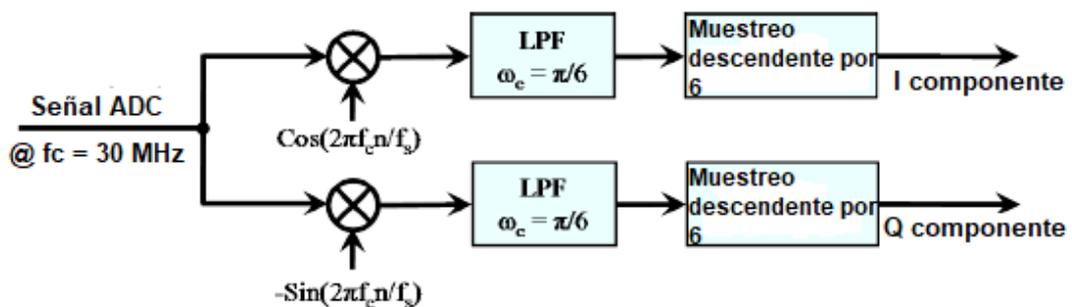


Figura 3. 12: Esquema de conversión descendente de la señal recibida  
Elaborado por: Autor.

El filtro de paso bajo FIR con 36 taps se utiliza para evitar el aliasing causado por el proceso de muestreo descendente y está diseñado con la herramienta de diseño y análisis de filtros (FDATool) de MATLAB. Mientras que la operación de filtrado de una señal compleja arbitraria con filtro FIR de 36 taps requieren de 72 multiplicadores, este costo se reduce a la mitad considerando solo las muestras de señal de entrada distintas de cero.

Además, la arquitectura propuesta para los filtros de diezmo se utiliza para reducir el costo de multiplicación a un sexto. Solo se utilizan seis segmentos DSP48 para implementar los filtros para las ramas I y Q.

### 3.3.2. Subsistema 2: Decodificación de preámbulos.

Las tareas de detección de trama, sincronización de tiempo y frecuencia del receptor son realizadas por el subsistema "preambleDecode". Como se mencionó en la sección 2.4, todas las tareas de sincronización utilizan comúnmente el algoritmo de retardo y correlación para calcular las variables de decisión. La suma del término de correlación para este algoritmo se puede reescribir de manera recursiva de la siguiente manera

$$\begin{aligned}
 M[n] &= \sum_{k=0}^{L-1} x[n-k] \cdot x^*[n-k-L] = \sum_{k=0}^{L-1} R_x[L, n-k] \\
 &= M[n-1] + R_x[L, n] - R_x[L, n-L] \quad (3.5)
 \end{aligned}$$

Como consecuencia de la ecuación (3.5), la arquitectura de retardo y correlación se implementa utilizando solo dos adiciones y una multiplicación compleja. La figura 3.13 muestra la señal compleja entrante que se retrasa a través de un registro de desplazamiento y el término de correlación se calcula mediante un multiplicador complejo basado en DSP48. Luego, la diferencia entre la salida del multiplicador con la versión retrasada se obtiene usando un bloque de resta y el resultado se acumula para formar la variable de decisión.

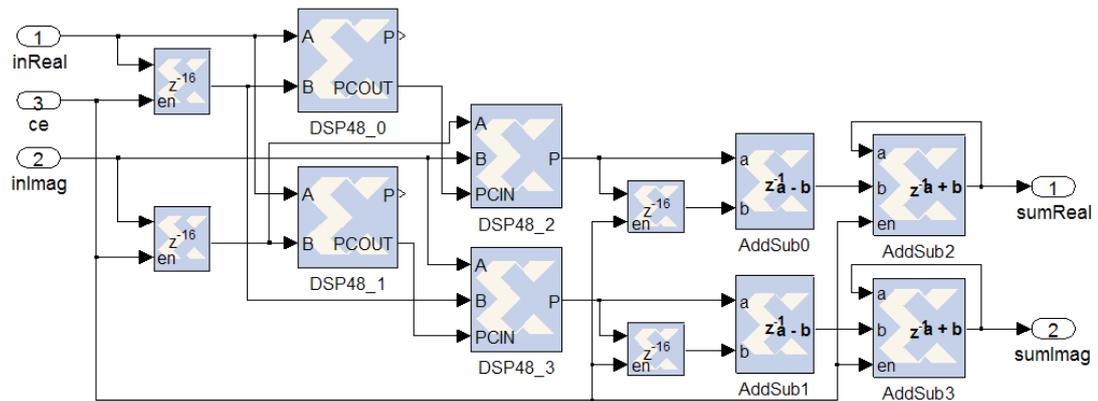


Figura 3. 13: Implementación del algoritmo de retraso y correlación.  
Elaborado por: Autor.

Además del método de retardo y correlación, el algoritmo de sincronización de temporización también utiliza la correlación cruzada entre la señal recibida y el símbolo de preámbulo corto original. El cálculo de la función de correlación cruzada en la ecuación (3.9) requiere 16 multiplicaciones complejas. Sin embargo, en la tabla 3.5 se muestra la parte imaginaria del

símbolo de preámbulo corto, que es una versión desplazada circularmente de su parte real y esta propiedad se puede usar para reducir la complejidad de la implementación.

Tabla 3. 5: Períodos de las secuencias cortas del preámbulo

k	Real (SP[k])	Imag (SP[k])	k	Real (SP[k])	Imag (SP[k])
0	0.1250	0.1250	8	0.1250	0.1250
1	-0.3599	0.0064	9	0.0064	-0.3599
2	-0.0366	-0.2134	10	-0.2134	-0.0366
3	0.3879	-0.0344	11	-0.0344	0.3879
4	0.2500	0.0	12	0.0	0.2500
5	0.3879	-0.0344	13	-0.0344	0.3879
6	-0.0366	-0.2134	14	-0.2134	-0.0366
7	-0.3599	0.0064	15	0.0064	-0.3599

Elaborado por: Autor.

Si la sumatoria en la ecuación (2.9) se divide en dos partes, la operación de correlación cruzada se puede realizar utilizando dos filtros FIR con coeficientes de valor real como los siguientes:

$$\begin{aligned}
 R_{rs}[n] &= \sum_{k=0}^{15} r[n-k]SP^*[15-k] \\
 &= \sum_{k=0}^7 r[n-k]SP^*[15-k] + \sum_{k=0}^7 r[n-k-8]SP^*[7-k] \\
 &= \sum_{k=0}^7 r[n-k] \cdot (h_2[k] - jh_1[k]) + \sum_{k=0}^7 r[n-k-8] \cdot (h_1[k] - jh_2[k]) \\
 &= (r[n] - jr[n-8]) \star h_2[n] + (r[n-8] - jr[n]) \star h_1[n] \quad (3.6)
 \end{aligned}$$

Además, se adapta la arquitectura simétrica del filtro FIR sistólica propuesta para implementar los filtros  $h_1[n]$  y  $h_2[n]$ . La figura 3.14 muestra el número requerido de multiplicadores para  $h_1[n]$  que se reduce a la mitad sumando las muestras de entrada antes de multiplicarse por el mismo coeficiente. Las multiplicaciones simples por 0.25 y 0.125 se logran simplemente moviendo el punto decimal binario a la derecha y las

multiplicaciones fraccionarias restantes se implementan usando multiplicadores incrustados.

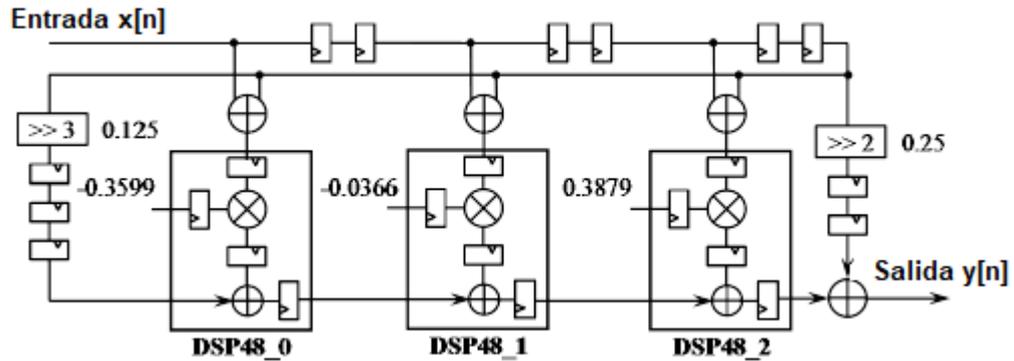


Figura 3. 14: Implementación del filtro de correlación,  $h_1[n]$ .  
Elaborado por: Autor.

En la implementación del algoritmo de sincronización de frecuencia, el núcleo de la computadora digital para rotación de coordenadas (*COordinate Rotation Digital Computer, CORDIC*) de Xilinx se utiliza para encontrar la fase del resultado de correlación en la ecuación 2.12. Después de estimar el desplazamiento de frecuencia a partir del valor de fase, se genera la senoide compleja deseada utilizando un núcleo de sintetizador digital directo (*Direct Digital Synthesizer, DDS*) de Xilinx. Luego, el desplazamiento de frecuencia se elimina multiplicando la señal de entrada con la senoide compleja generada.

### 3.3.3. Subsistema 3: Estimación de canal.

El subsistema de estimación de canal "chanEstimation" mostrado en la figura 3.11, se encarga de estimar los coeficientes de respuesta al impulso del canal como se describe en la subsección 2.5.5. La figura 3.15 muestra el diagrama de bloques de implementación FPGA del subsistema "chanEstimation". Este subsistema calcula en primer lugar el promedio de los símbolos de preámbulo largos con frecuencia corregida y almacena las muestras de resultados en un búfer de RAM basado en bloques. Después de recibir todas las muestras de los símbolos de preámbulo largos, el subsistema lee las muestras de preámbulo de este búfer y el inverso generalizado de  $X$  en la ecuación (2.18) desde una ROM y realiza la compleja operación de multiplicación de matrices.

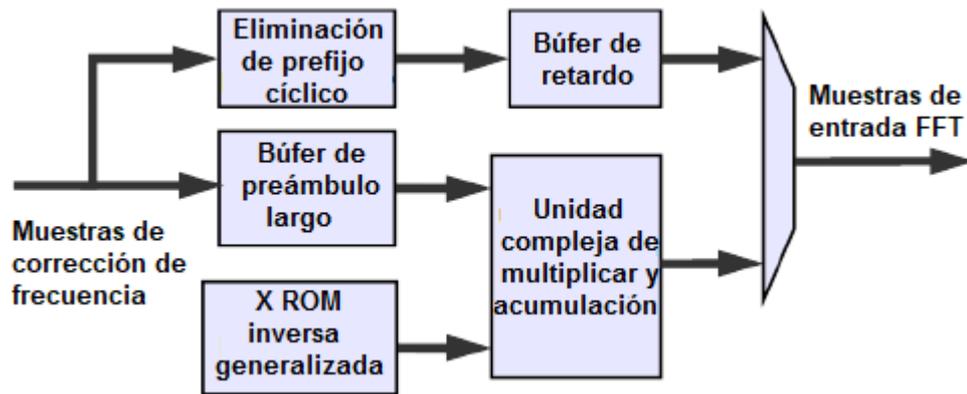


Figura 3. 15: Diagrama de bloques del subsistema de canal de estimación.  
Elaborado por: Autor.

La unidad compleja de multiplicar y acumular mostrada en la figura 3.15 incluye dos multiplicadores complejos paralelos y completa la operación de la matriz en 512 ciclos de reloj. Luego, los coeficientes de canal calculados se envían al bloque FFT para obtener la función de transferencia de canal para todas las subportadoras. Además de la estimación de los coeficientes del canal, este subsistema elimina el intervalo de prefijo cíclico de las muestras recibidas y envía las muestras restantes de los símbolos de datos OFDM al bloque FFT por multiplexación en la salida. Debido a la latencia de ejecución de la operación de matriz, las muestras recibidas deben demorarse antes de enviarse al bloque FFT.

#### 3.3.4. Subsistema 4: Ecualizador de canal

La figura 3.16 muestra la implementación del generador del sistema del subsistema de ecualización de canal (channelEqualizer) que realiza el proceso de ecualización del canal mediante la evaluación de la operación de división compleja en la ecuación (2.20). Esta operación de división se realiza en dos pasos, primero el inverso (bajo multiplicación) de los coeficientes de respuesta de frecuencia del canal se calculan de la siguiente manera,

$$\frac{1}{\hat{H}[k]} = \frac{\hat{H}[k]^*}{\hat{H}[k] \cdot \hat{H}[k]^*} \quad (3.7)$$

entonces los valores calculados se multiplican con los símbolos de subportadora entrantes del subsistema FFT.

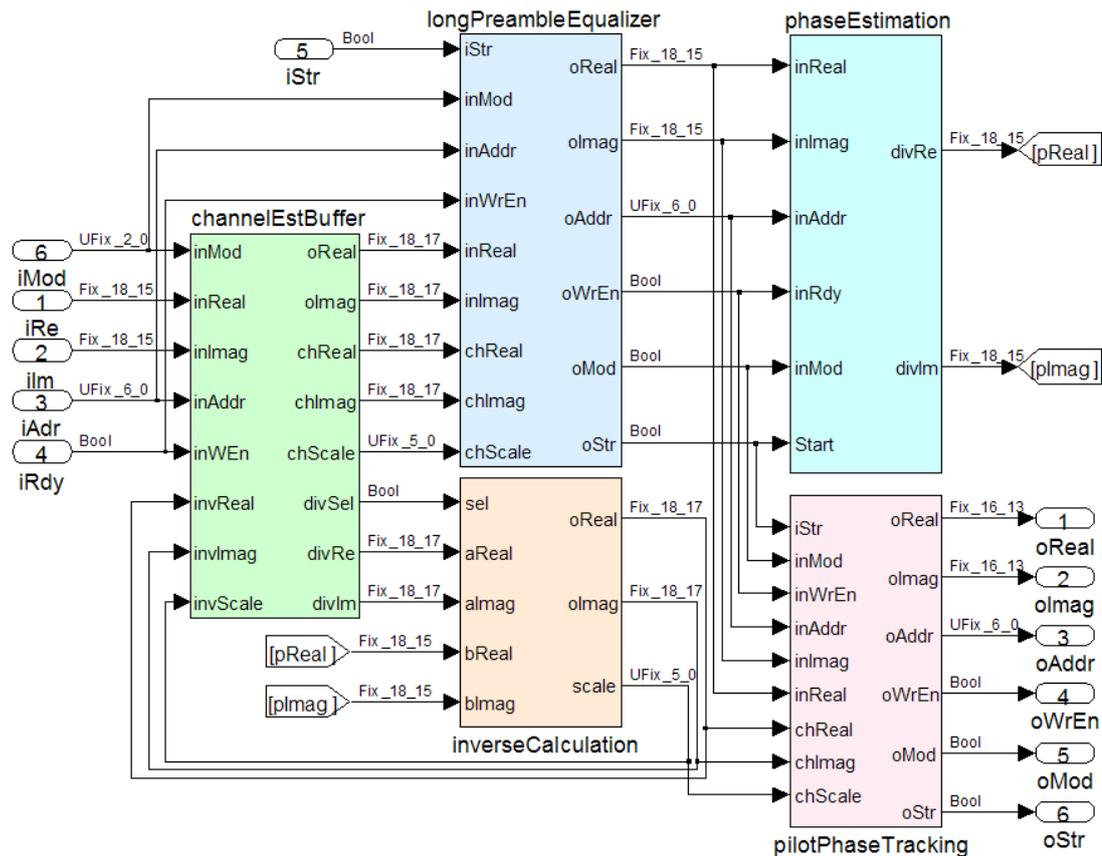


Figura 3. 16: Subsistema de ecualización de canales.

Elaborado por: Autor.

El bloque "inverseCalculation" está diseñado para implementar la operación de división en la ecuación (3.7). Este bloque calcula primero el cuadrado absoluto de los coeficientes de respuesta de frecuencia del canal multiplicándolos con versiones conjugadas complejas y el resultado del cuadrado absoluto se expresa por 32 bits. Para simplificar el cálculo inverso de este resultado, se trunca al tomar los 12 bits más significativos. Para este propósito, el bloque encuentra la posición del primer bit distinto de cero del resultado desde el lado más significativo y toma el consecutivo 12 bits desde esta posición hasta el lado menos significativo.

Luego, estos bits se utilizan para direccionar una tabla de búsqueda basada en RAM de bloque que contiene los valores inversos (bajo multiplicación). Finalmente, los valores obtenidos se multiplican por el conjugado de los coeficientes de respuesta de frecuencia para obtener la inversa bajo la multiplicación de estos coeficientes. Sin embargo, debido a la

escala durante la operación de truncamiento, las salidas de este bloque se transfieren a los siguientes bloques con un factor de escala.

Los coeficientes de respuesta de frecuencia del canal se obtienen del subsistema "FFT64" y son capturados por el bloque "channelEstBuffer". Este bloque envía los coeficientes capturados al bloque "inverseCalculation" para realizar la operación de división y almacena los resultados de división con los factores de escala en una RAM de bloque. Mientras procesa los símbolos OFDM seguidos de los preámbulos, el inverso de las estimaciones de respuesta de frecuencia del canal y los símbolos entrantes se transmiten al bloque "longPreambleEqualizer". En este bloque, el procedimiento de ecualización de canal basado en el preámbulo se completa realizando la compleja multiplicación de las muestras de entrada y compensando el efecto del factor de escala.

Además de la ecualización de cada subportadora, el subsistema "channelEqualizer" también estima el valor de fase de portadora acumulada de cada símbolo de datos OFDM como se indica en la ecuación (2.21) y gira los símbolos de constelación ecualizados de acuerdo con este valor. La estimación de los valores de la fase de portadora se realiza mediante el bloque "phaseEstimation" que agrega los símbolos en las subportadoras piloto después de corregir las polaridades de estas. Luego, se debe encontrar la fase de esta suma y se debe generar una señal exponencial cuya fase sea negativa del valor estimado para realizar el giro.

En lugar de utilizar un CORDIC y un núcleo de generador de seno-coseno para implementar estas operaciones en hardware, la señal de giro exponencial se puede obtener normalizando la amplitud de la suma piloto y conjugando el resultado. Para este propósito, el bloque "phaseEstimation" envía el resultado de la suma al bloque "inverseCalculation". En este caso, este bloque calcula el inverso de la señal de entrada de la siguiente manera:

$$\frac{\hat{P}[n]^*}{\sqrt{\hat{P}[n]^* \cdot \hat{P}[n]}} = \exp(-j \cdot \angle \hat{P}[n]) \quad (3.8)$$

Donde  $\hat{P}[n]$  es la suma de las subportadoras piloto para el enésimo símbolo OFDM. Después de generar la señal de desrotación exponencial, el bloque "pilotPhaseTracking" realiza la desrotación multiplicando los símbolos de la constelación del ecualizador con este exponencial.

### 3.3.5. Subsistema 5: Demodulación de subportadora

El subsistema "subCarrierDemod" demodula los símbolos de constelación ecualizados y obtiene las relaciones de probabilidad de registro de los bits recibidos que serán utilizados por el decodificador Viterbi de decisión flexible. Para los tipos de modulación BPSK y QPSK, el LLR de cada bit puede calcularse fácilmente evaluando solo los componentes en fase o en cuadratura de los símbolos de la constelación. Sin embargo, el cálculo de LLR es más difícil para los tipos de modulación 16-QAM y 64-QAM (véase las tablas 3.2 y 3.3), porque cada componente de la constelación lleva dos y tres bits de información, respectivamente.

Se utiliza el método simplificado de razón de probabilidad logarítmica para calcular los LLR asociados con los bits transportados sobre el componente en fase que se dan en las siguientes ecuaciones. Estas ecuaciones, a excepción del BPSK, también se utilizan para calcular las proporciones de los bits en el eje imaginario sustituyendo el componente de cuadratura del símbolo de constelación en las ecuaciones.

$$\text{BPSK} : LLR(b_0) = [y_I[n] \times 32] \quad (3.9)$$

$$\text{QPSK} : LLR(b_0) = [y_I[n] \times \sqrt{2} \times 32] \quad (3.10)$$

$$\begin{aligned} \text{16QAM} : LLR(b_0) &= [y_I[n] \times \sqrt{10} \div 3 \times 40] \\ LLR(b_1) &= \left[ \left( 2 - \left| y_I[n] \times \sqrt{10} \right| \right) \div 3 \times 40 \right] \end{aligned} \quad (3.11)$$

$$\begin{aligned} \text{64QAM} : LLR(b_0) &= [y_I[n] \times \sqrt{42} \div 7 \times 48] \\ LLR(b_1) &= \left[ \left( 4 - \left| y_I[n] \times \sqrt{42} \right| \right) \div 7 \times 48 \right] \\ LLR(b_2) &= \left[ \left( 2 - \left| 4 - \left| y_I[n] \times \sqrt{42} \right| \right| \right) \div 7 \times 48 \right] \end{aligned} \quad (3.12)$$

donde el operador  $[\cdot]$  denota el redondeo al entero más cercano.

Para realizar las operaciones inversas del intercalado y la punción en el transmisor, el subsistema "subCarrierDemod" almacena los LLR del símbolo OFDM actualmente decodificado en una RAM de bloque. En el siguiente intervalo de decodificación de símbolos, este subsistema lee los LLR almacenados de la memoria en un orden desintercalado y puncionado y los envía al procesador del algoritmo Viterbi.

### 3.3.6. Subsistema 5: Decodificador Viterbi

El receptor IEEE802.11a utiliza el algoritmo de Viterbi para decodificar los códigos de corrección de errores directos codificados convolucionalmente. El algoritmo de Viterbi se representa comúnmente mediante un diagrama enrejado, que muestra las transiciones de estado de manera indexada en el tiempo. Un diagrama enrejado simple con cuatro estados se ilustra en la figura 3.17a. Hay dos ramas salientes para cada estado en un instante de tiempo dado, una corresponde al bit de entrada del codificador de cero y la otra corresponde a un bit de entrada de uno.

El algoritmo primero calcula las probabilidades de cada rama saliente, denominada métrica de rama, y las acumula en la métrica de ruta (estado) del instante de tiempo anterior. Debido a la existencia de dos caminos entrantes para cada estado, el algoritmo selecciona el que tiene una métrica más grande y registra el seleccionado como un camino de sobreviviente. Finalmente, después de procesar todos los datos de entrada en una trama, la secuencia de salida decodificada se obtiene rastreando hacia atrás la ruta más probable.

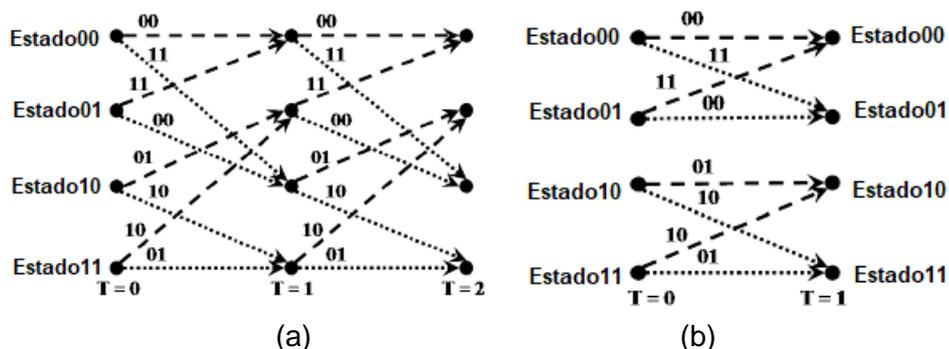


Figura 3. 17: (a) Diagrama enrejado del decodificador Viterbi, (b) estructura de mariposa.

Elaborado por: Autor.

La implementación del generador del sistema del decodificador Viterbi consta de tres unidades, tal como se muestra en la figura 3.18. La unidad de métrica de rama (*Branch Metric Unit, BMU*) se encarga de calcular las métricas para solo ramas de "11" y "10" agregando y restando los LLR entrantes, respectivamente. No es necesario calcular las otras métricas porque son iguales a las negativas de las calculadas.

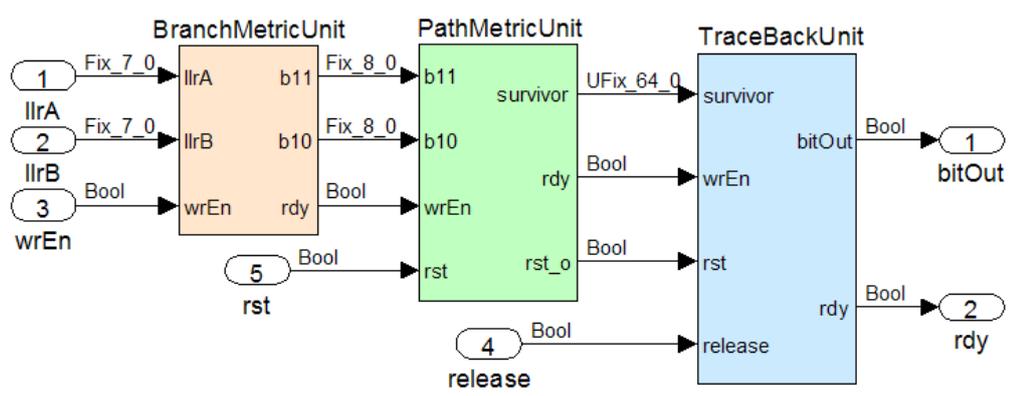


Figura 3. 18: Diagrama de bloques del generador del sistema del decodificador Viterbi

Elaborado por: Autor.

El codificador convolucional utilizado en el estándar IEEE802.11a tiene una línea de retardo con seis registros, por lo que el número de estados en el diagrama enrejado es igual a 26. Para proporcionar una simplicidad en la implementación, estos estados se reorganizan en 32 estructuras de mariposa tal como se muestra en la figura 3.17b para cuatro estados. La unidad de métrica de ruta (*Path Metric Unit, PMU*) del decodificador Viterbi utiliza una unidad de añadir-comparar-seleccionar (ACS), que se muestra en la figura 3.19, para cada mantequilla y para calcular los nuevos valores de las métricas de ruta y determinar las rutas de los sobrevivientes.

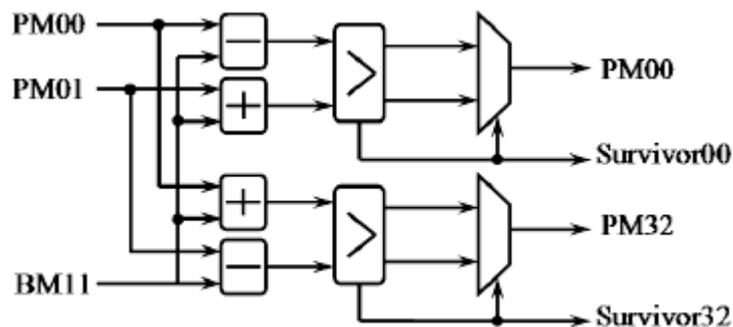


Figura 3. 19: Diagrama de bloques de la unidad de mariposa ACS.

Elaborado por: Autor.

En la implementación de la unidad ACS con aritmética de precisión finita, se utiliza la técnica de normalización de módulo para evitar problemas de sobrecarga de las métricas de ruta calculadas recursivamente. Según esta técnica, el rango del registro de la métrica de la ruta debe ser mayor que dos veces la diferencia máxima posible entre las métricas de la ruta. En este diseño de receptor, las diferencias entre las métricas son menores a 1024 para los LLR expresados con siete bits. Por lo tanto, el ancho de los registros métricos de la ruta se ajustó como 11 bits.

El último bloque en el decodificador Viterbi es la unidad de rastreo (*Trace Back Unit, TBU*) que se implementó. Esta unidad almacena las 64 rutas de sobrevivientes entrantes generadas por la PMU en una de las cuatro RAM de bloque, tal como se ilustra en la figura 3.20. Mientras se almacenan los sobrevivientes del  $n$ -ésimo bloque, el bloque previamente almacenado ( $n-1$ ) se remonta para determinar el estado inicial del bloque  $n-2$ . Al mismo tiempo, los bits de salida para el bloque  $n-3$  se recopilan rastreando hacia atrás el bloque asociado y se almacenan en un registro de desplazamiento. En el siguiente intervalo de procesamiento de bloque, los bits decodificados se leen desde el registro de desplazamiento en orden inverso y se dirigen a la salida. El proceso de decodificación canalizada continúa de esta manera

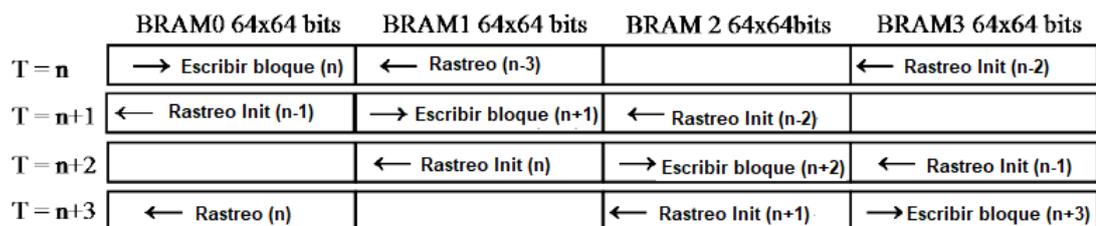


Figura 3. 20: Diagrama de tiempos del acceso de memoria de rastreo.

Elaborado por: Autor.

### 3.4. Mediciones de desempeño

Para probar el prototipo de capa física IEEE802.11a en la plataforma de desarrollo Lyrtech SFF SDR, también se desarrolló una capa MAC simple que se ejecuta en la plataforma y un programa de interfaz de usuario en la PC. Esta estructura permite al usuario observar el rendimiento del transceptor bajo un enlace inalámbrico construido entre dos plataformas Lyrtech o bajo un canal AWGN modelado en banda base.

La figura 3.21 muestra el programa de interfaz de usuario que se conecta a la plataforma a través de Ethernet y que permite ajustar todos los parámetros operativos del módem y las configuraciones RF/IF de la plataforma. Debido a que los módulos de RF en la plataforma pueden operar en un rango de frecuencia de 262 MHz a 876 MHz, para lo cual se realizaran las pruebas del prototipo en este rango de frecuencia en lugar de un escenario real a 5 GHz.

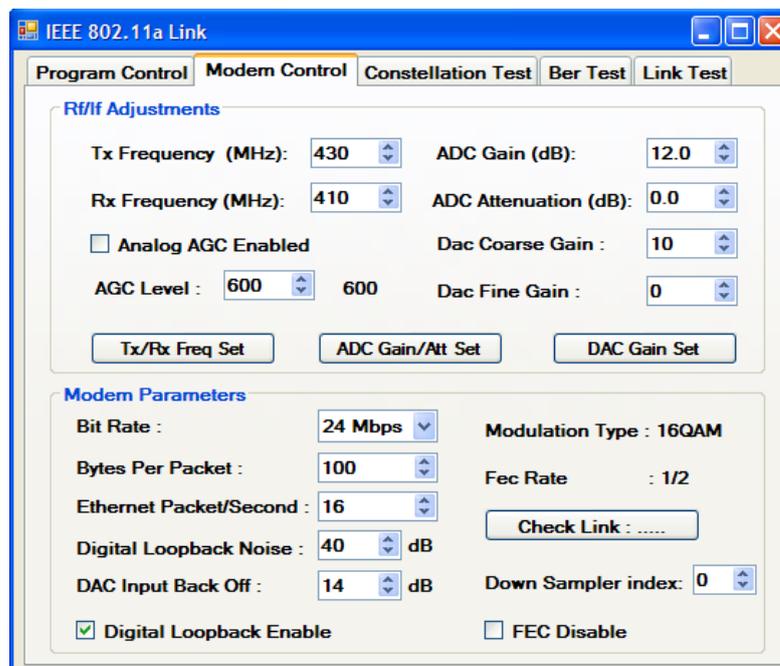


Figura 3. 21: Instantánea del programa de interfaz de usuario  
Elaborado por: Autor.

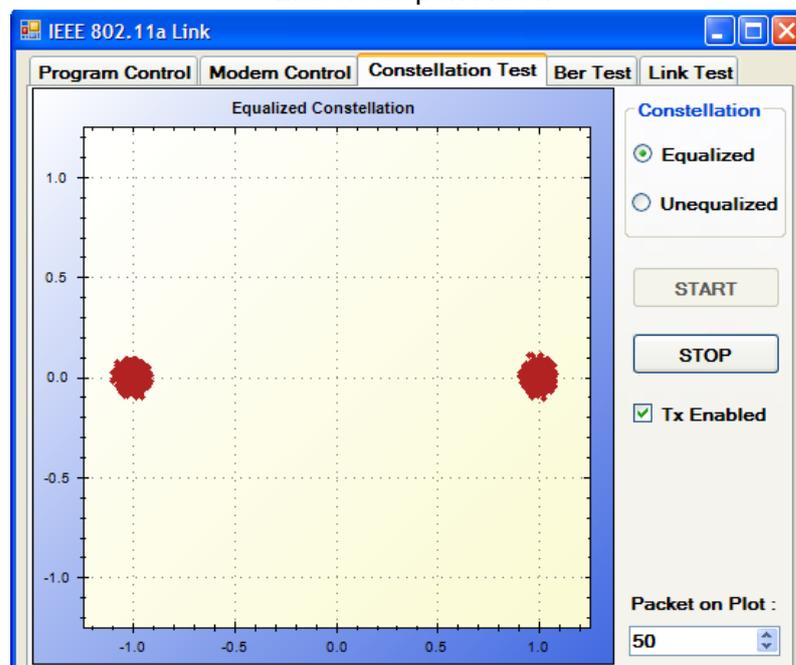


Figura 3. 22: Instantánea de tiempo de ejecución de la constelación BPSK.  
Elaborado por: Autor.

En la pestaña de prueba de constelación (Constellation Test) del programa de usuario, la constelación de la señal recibida se controla en tiempo de ejecución, tal como se ilustra en las figuras 3.22 a 3.25. En este modo, la capa MAC recolecta las muestras de constelación ecualizadas o no ecualizadas del receptor y las envía a la PC a través de un paquete Ethernet.

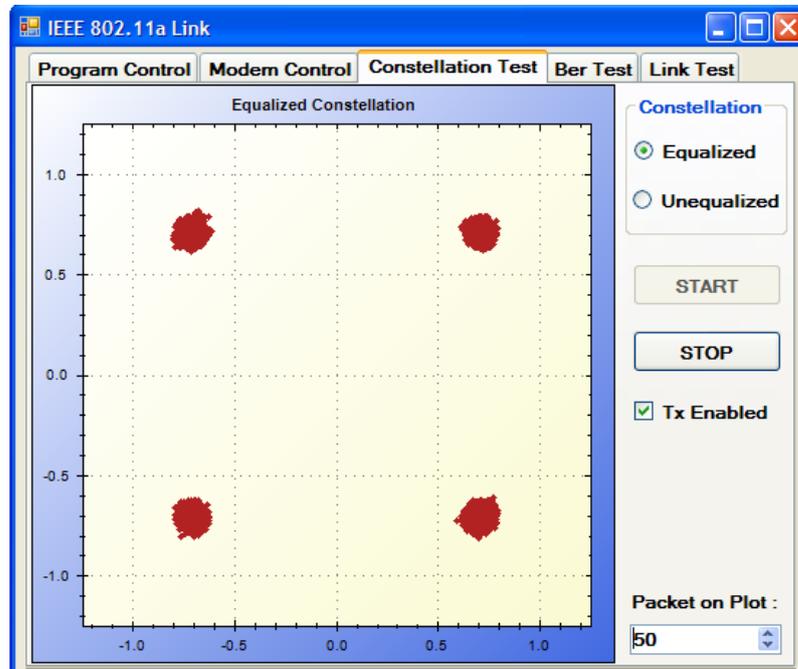


Figura 3. 23: Instantánea de tiempo de ejecución de la constelación QPSK.  
Elaborado por: Autor.

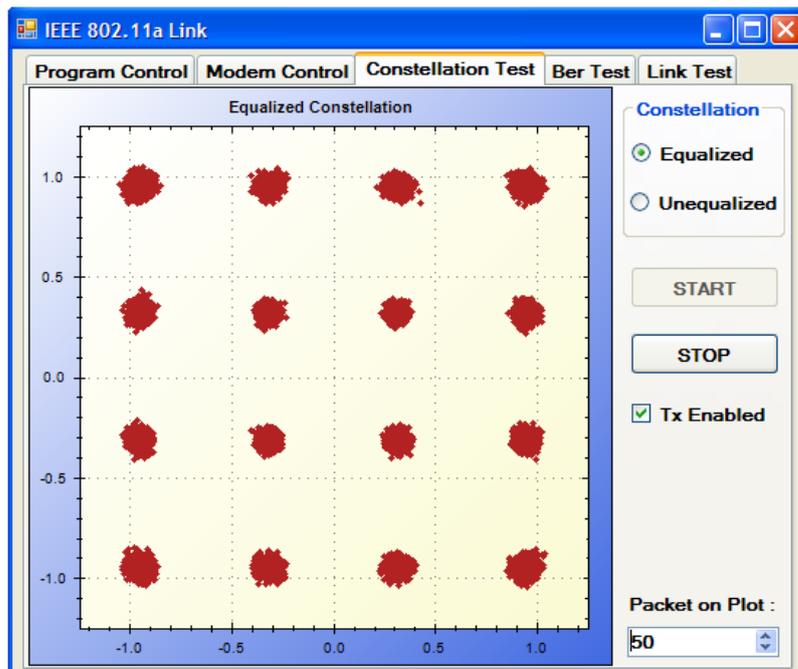


Figura 3. 24: Instantánea de tiempo de ejecución de la constelación 16-QAM.  
Elaborado por: Autor.

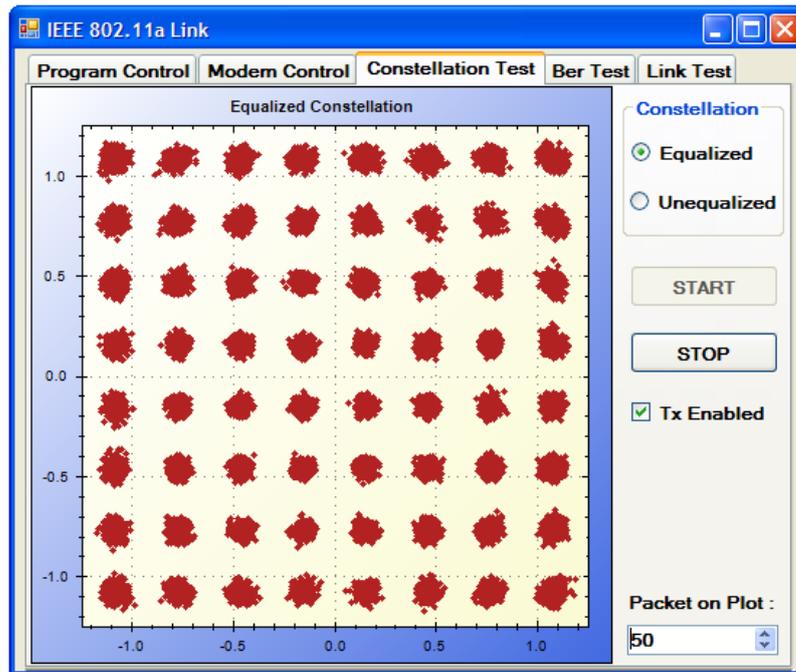


Figura 3. 25: Instantánea de tiempo de ejecución de la constelación 64-QAM.  
Elaborado por: Autor.

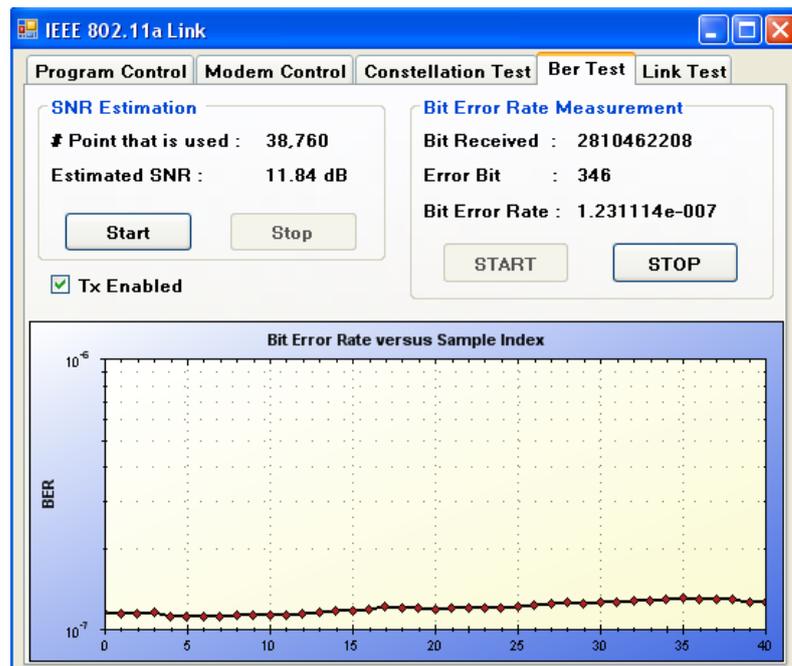


Figura 3. 26: Instantánea de tiempo de ejecución de medición de la BER.  
Elaborado por: Autor.

Además del monitoreo de la constelación, la capa MAC también mide el rendimiento de la tasa de error de bits (BER) del receptor. Genera una secuencia binaria pseudoaleatoria utilizando un registro de desplazamiento de retroalimentación lineal (LFSR) para el transmisor y envía la señal de banda

base generada al transmisor de RF o realiza una prueba de bucle de retorno digital bajo el modelo de canal AWGN. Luego, produce la estadística BER comparando los bits recibidos con una secuencia sincronizada con los transmitidos e informa los resultados estadísticos al programa de interfaz de usuario, tal como se muestra en la figura 3.26.

Al usar el programa de interfaz de usuario, se mide el rendimiento BER del prototipo para los tipos de modulación QPSK, 16-QAM y 64-QAM. Las curvas BER obtenidas se representan en las figuras 3.27 a 3.29 para estos tipos de modulación con las tasas de codificación asociadas.

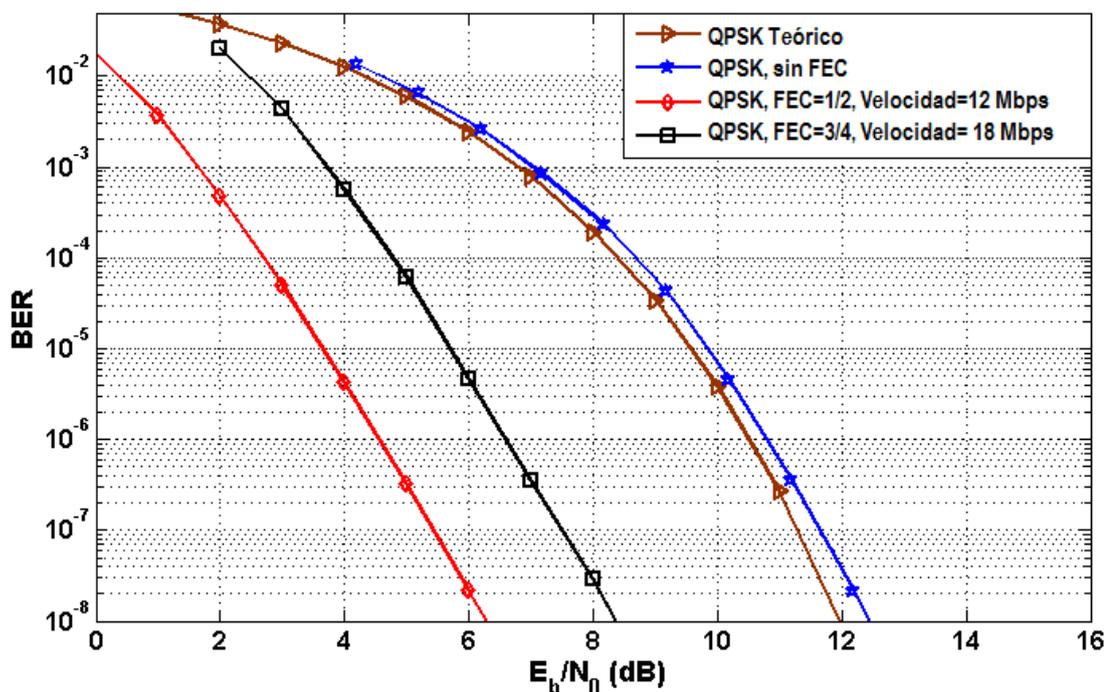


Figura 3. 27: Rendimiento BER de la modulación QPSK  
Elaborado por: Autor.

También, se realiza la medición del rendimiento del transceptor deshabilitando los códigos de corrección de errores y se compara la curva BER obtenida con el límite superior teórico para el tipo de modulación correspondiente.

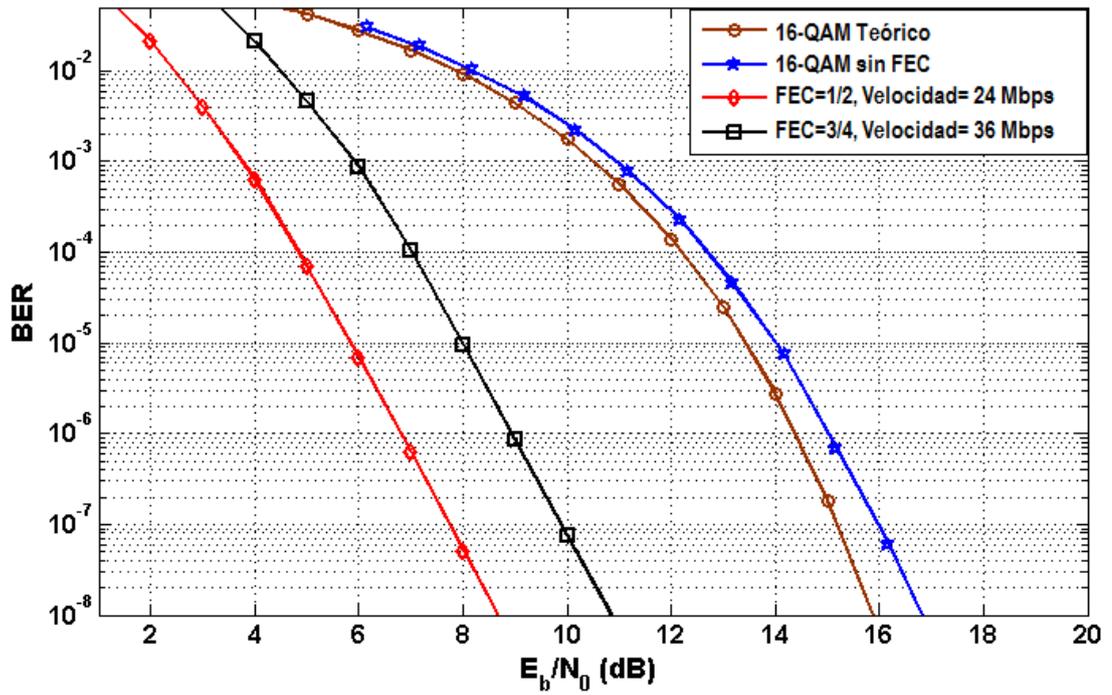


Figura 3. 28: Rendimiento BER de la modulación 16-QAM  
Elaborado por: Autor.

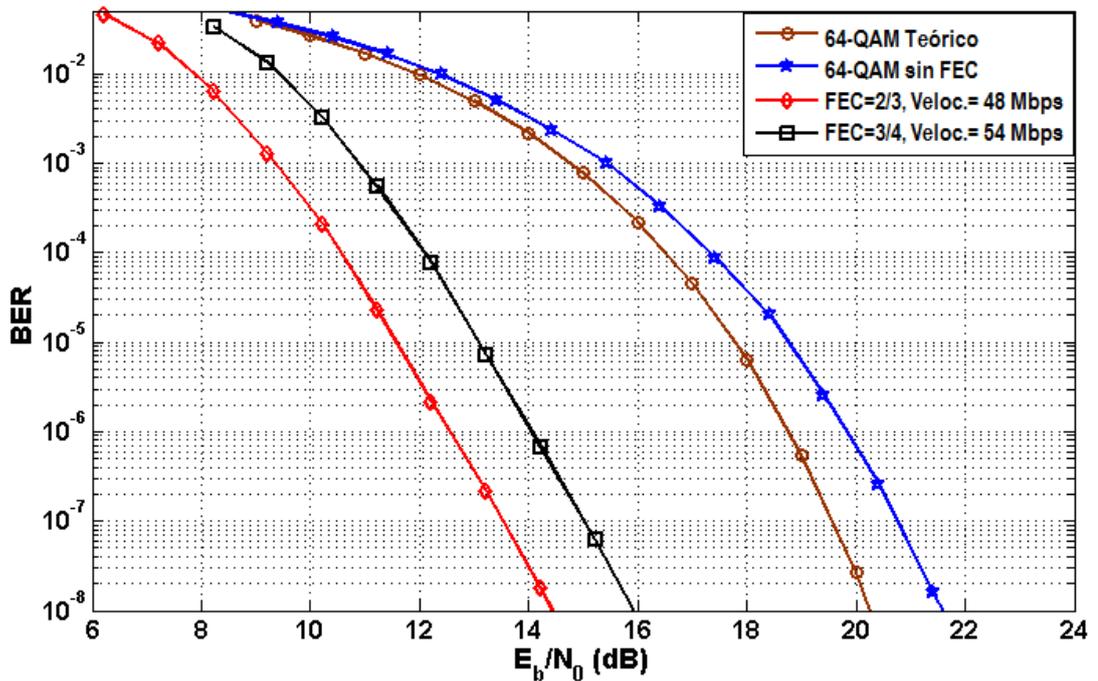


Figura 3. 29: Rendimiento BER de la modulación 64-QAM  
Elaborado por: Autor.

## Conclusiones

1. Según las simulaciones, el receptor se ha desarrollado de tal manera que es más confiable y fácil de implementar. Los problemas de temporización y sincronización de frecuencia de la técnica OFDM se han considerado exclusivamente para obtener un buen rendimiento. Para superar el problema PAPR del sistema, el amplificador en el transmisor funciona en su región lineal y la señal digital enviada al DAC se corta si la potencia de la señal excede el límite máximo permitido.
2. La implementación de alta precisión del transceptor utiliza 5986 segmentos, 45 bloques de RAM y 73 multiplicadores de los recursos lógicos FPGA correspondientes al 39% del chip sx35. Los núcleos FPGA del transmisor y receptor, que admiten todas las velocidades de datos definidas en el estándar, se han diseñado utilizando la caja de herramientas Xilinx "System Generator" para Simulink. Este entorno de implementación permite el desarrollo de sistemas DSP de alto rendimiento y proporciona generación automática de código HDL.
3. El rendimiento de la tasa de error de bit (BER) de la implementación de FPGA diseñada está bastante cerca de los límites superiores teóricos de los tipos de modulación QPSK, 16-QAM y 64-QAM. La diferencia entre las curvas BER medidas y teóricas de la modulación QPSK es de aproximadamente 0.4 dB. La razón de esta degradación es la imperfección de la sincronización de frecuencia, la estimación del canal y las tareas de ecualización del canal del receptor. Además, la degradación de las modulaciones QAM es un poco más que la QPSK porque el tipo de modulación QAM es más sensible al error de fase en la constelación.

## **Recomendaciones**

1. Ampliar el estudio de las funcionalidades de las tarjetas FPGA para desarrollar proyectos de investigación y de trabajos de titulación en el área de las telecomunicaciones y del procesamiento de señales e imágenes digitales.
2. Realizar análisis comparativos del sistema propuesto utilizando las FPGA de Altera y Xilinx.

## Bibliografía

- Baños, V., Afaqui, M. S., Lopez, E., & Garcia, E. (2017). Throughput and Range Characterization of IEEE 802.11ah. *IEEE Latin America Transactions*, 15(9), 1621–1628. <https://doi.org/10.1109/TLA.2017.8015044>
- Batra,P., (2012). Design and Implementation of Sirpinski Carpet Fractal Antenna for WirelessComunication. Trabajo de Titulación de Maestría, Pontificia Universidad Católica del Ecuador). Recuperado de <http://repositorio.puce.edu.ec:80/xmlui/handle/656500/6446>
- Chen, C., & Yang, Z. (2012). Three Timing Synchronization Methods Based on Two Same Preambles for OFDM Systems. *Procedia Engineering*, 29, 1656–1661. <https://doi.org/10.1016/j.proeng.2012.01.190>
- Estrella F., S. F. (2017). *Estudio y análisis para la actualización de red Wlan de la Seps utilizando tecnologías basadas en el estándar IEEE 802.11 AC* (Trabajo de Titulación de Maestría, Pontificia Universidad Católica del Ecuador). Recuperado de <http://repositorio.puce.edu.ec:80/xmlui/handle/22000/14425>
- Fonseca J. (2016). *Redes inalámbricas*. Recuperado de <https://uplikonf.upc.edu/handle/12084/398485/08>
- Herrera B., J. E., & Rodríguez L., V. S. (2018). *Diseño e implementación de un sincronizador OFDM implementado en USRP* (Trabajo de Titulación de Grado, Universidad Técnica Particular de Loja). Recuperado de <http://dspace.utpl.edu.ec/handle/20.500.11962/21683>
- Hidrobo J., (2013). Diseño e Implementación de una antena fractal en base a la a curva de koch para la recepción de señales. Trabajo de Titulación de Grado Escuela Politécnica Nacional). Recuperado de <http://repositorio.epn.edu.ec:47/283848/885>

Paez M., (2006). Comportamiento de la Mampostriobajo Esfuerzos Cortantes debido a efectos Sísmicos. Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/7285>

Ramón A., V. A. (2014). *Estudio y diseño de una red inalámbrica para dotar servicios de telecomunicaciones a 55 centros educativos del cantón Shushufindi aplicando criterios de calidad de servicio y seguridad de red.* (Trabajo de Titulación de Grado, Escuela Politécnica Nacional). Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/7285>

Rodríguez, J., (2018). Implementación de un sincronizador inalámbrica aplicando los estándares de calidad (Trabajo de Titulación de Grado, Universidad Técnica Particular de Loja). Recuperado de <http://dspace.utpl.edu.ec/handle/20./456>

Salazar Soler, J. (2016). *Redes inalámbricas.* Recuperado de <https://upcommons.upc.edu/handle/2117/100918>

Sivadas, V., Afaqui, M. S., Lopez, E., & Garcia, E. (2017). Design and development of Sierpinski Carpet Microstrip Fractal Antenna for Multiband Applications. *688* 15(9), 1621–1628. <https://doi.org/10.1109/TLA.2017.8015044>



**Presidencia  
de la República  
del Ecuador**



**Plan Nacional  
de Ciencia, Tecnología,  
Innovación y Saberes**



**SENESCYT**  
Secretaría Nacional de Educación Superior,  
Ciencia, Tecnología e Innovación

## **DECLARACIÓN Y AUTORIZACIÓN**

Yo, **Alcocer Alcocer, Guido Adrián** con C.C: # 092595822-5 autor del trabajo de titulación: Evaluación del estándar IEEE 802.11a a nivel de la capa física utilizando la tarjeta FPGA-Xilinx, previo a la obtención del título de **Ingeniero en Telecomunicaciones** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 13 de septiembre del 2019

f. \_\_\_\_\_

Nombre: **Alcocer Alcocer, Guido Adrián**

C.C: 092595822-5

<b>REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA</b>			
<b>FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN</b>			
<b>TÍTULO Y SUBTÍTULO:</b>	Evaluación del estándar IEEE 802.11 <sup>a</sup> a nivel de la capa física utilizando la tarjeta FPGA-Xilinx		
<b>AUTOR(ES)</b>	Alcocer Alcocer, Guido Adrián		
<b>REVISOR(ES)/TUTOR(ES)</b>	M. Sc. Suarez Murillo, Efraín Oswaldo		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Técnica		
<b>PROGRAMA:</b>	Ingeniería en Telecomunicaciones		
<b>TITULO OBTENIDO:</b>	Ingeniero en Telecomunicaciones		
<b>FECHA DE PUBLICACIÓN:</b>	13 de septiembre del 2019	<b>No. DE PÁGINAS:</b>	68
<b>ÁREAS TEMÁTICAS:</b>	Comunicaciones Inalámbricas, Sistemas Digitales		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	Multiplexación, Frecuencia, OFDM, IEEE, Estándares, FPGA.		
<b>RESUMEN/ABSTRACT (150-250 palabras):</b>			
<p>La multiplexación por división de frecuencia ortogonal (OFDM) es una de las técnicas de transmisión más prometedoras entre las tecnologías existentes debido a sus características únicas. Hoy en día, existe en el mercado una gran cantidad de posibilidades para implementar una red inalámbrica. El IEEE respondiendo a las necesidades del mercado y los fabricantes, comprendió la necesidad de un estándar que limitase y definiese cada uno, para que su uso fuese lo más eficiente posible. La estandarización ha permitido desvincularse de tecnologías propietarias, consiguiendo una plataforma abierta con productos de mayores prestaciones y a un precio mucho más ajustado. Este trabajo de titulación comienza con una breve descripción general de las ventajas y desventajas de los sistemas OFDM y tiene como objetivo implementar la capa física del estándar LAN inalámbrico IEEE802.11a, que es uno de los estándares IEEE basados en OFDM, en los arreglos de puerta programables de campo (FPGA). Antes de comenzar la implementación, el sistema general IEEE802.11a se ha modelado y simulado en el entorno MATLAB.</p>			
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> 0985671828	E-mail: <a href="mailto:mariamlema98@gmail.com">mariamlema98@gmail.com</a>	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):</b>	<b>Nombre:</b> Edwin Fernando Palacios Meléndez		
	<b>Teléfono:</b> 0967608298		
	<b>E-mail:</b> <a href="mailto:edwin.palacios@cu.ucsg.edu.ec">edwin.palacios@cu.ucsg.edu.ec</a>		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			