



**Instituto Politécnico Nacional**

**Centro de Investigación en Computación**

Secretaría de Investigación y Posgrado



**Módem para la comunicación por la red eléctrica**

**Tesis**

que para obtener el grado de

**Maestro en Ciencias en Ingeniería de Cómputo  
con opción en Sistemas Digitales**

presenta

**Ing. Raúl Antonio Méndez**

Directores de tesis:

**Dr. Ricardo Barrón Fernández  
Dr. José Luis Oropeza Rodríguez**

México, D.F.

Diciembre del 2011

# Agradecimientos

## **Gracias**

*A mis padres, Fortino Antonio Pérez y Angelica Méndez Méndez por depositar en mi su confianza, cariño y apoyo. A mis hermanos, por compartir conmigo esta aventura llamada vida.*

## **Gracias**

*A mis directores de tesis, Dr. Ricardo Barrón Fernández y Dr. José Luis Oropeza Rodríguez cuya orientación y ayuda han hecho posible este trabajo.*

## **Gracias**

*Al Instituto Politécnico Nacional, por permitirme realizar mi formación durante mis estudios de maestría.*

## **Gracias**

*A mis profesores, por compartir su conocimiento, y formar parte de mi preparación tanto a nivel profesional como personal, durante mi estancia en el Centro de Investigación en Computación.*

## **Gracias**

*A mis compañeros, por compartir sus consejos, experiencias, y por su apoyo durante mi formación.*

## **Gracias**

*Al Consejo Nacional de Ciencia y Tecnología, CONACyT, por el apoyo económico proporcionado durante mi preparación académica.*

# Índice general

Índice de tablas	IV
Índice de figuras	V
<b>1. Introducción</b>	<b>1</b>
1.1. La red eléctrica como medio de comunicación	1
1.2. Antecedentes	2
1.3. Planteamiento del problema	3
1.4. Justificación	4
1.5. Hipótesis	4
1.6. Objetivo general	4
1.7. Objetivos específicos	4
1.8. Alcance	5
1.9. Aportaciones	5
1.10. Organización de la tesis	5
1.11. Resumen del capítulo	6
<b>2. Estado del arte</b>	<b>7</b>
2.1. Comunicaciones eléctricas	7
2.2. Tecnología PLC	9
2.3. Clasificación y áreas de aplicación	10
2.4. Estándares existentes	11
2.5. El canal: Problemas y soluciones	14
2.5.1. Impedancia	14
2.5.2. Atenuación	14
2.5.3. Ruido	14
2.6. Trabajos relacionados	15
2.7. Resumen del capítulo	16
<b>3. Marco Teórico</b>	<b>17</b>
3.1. Sistema digital de comunicaciones	17
3.2. Técnicas de modulación digital	19
3.3. QAM	20
3.4. OFDM	22
3.4.1. Modulación y demodulación	23
3.4.2. Aplicación de la DFT	25

3.4.3.	Intervalo de guarda . . . . .	26
3.4.4.	Sincronización . . . . .	27
3.5.	Transformada Rápida de Fourier . . . . .	28
3.5.1.	Diezmado en frecuencia . . . . .	31
3.6.	Resumen del capítulo . . . . .	32
<b>4.</b>	<b>Implementación del módem</b> . . . . .	<b>34</b>
4.1.	Arquitectura general . . . . .	34
4.2.	Transmisor . . . . .	35
4.3.	Receptor . . . . .	35
4.4.	Acoplamiento . . . . .	37
4.5.	Hardware . . . . .	38
4.5.1.	TMS320C6416T DSK . . . . .	38
4.5.2.	Spartan 3E Breakout Board . . . . .	39
4.5.3.	ADS5485EVM . . . . .	39
4.5.4.	DAC5672EVM . . . . .	39
4.6.	Software . . . . .	40
4.6.1.	Code Composer Studio 3.3 . . . . .	40
4.6.2.	Xilinx ISE Desing Suite 12.2 . . . . .	40
4.6.3.	MATLAB . . . . .	40
4.7.	Algoritmos . . . . .	40
4.7.1.	Modulador y demodulador QAM . . . . .	41
4.7.2.	FFT, IFFT . . . . .	42
4.7.3.	Módulo de correlación IG . . . . .	43
4.7.4.	Detector de símbolo piloto . . . . .	44
4.7.5.	Estimador de diferencia de ventana . . . . .	45
4.7.6.	Estimador de frecuencia de muestreo . . . . .	45
4.7.7.	Estimador de canal . . . . .	46
4.7.8.	Ecuador . . . . .	47
4.8.	Módulo de adquisición . . . . .	47
4.8.1.	Controlador EMIF . . . . .	47
4.8.2.	Controlador ADC . . . . .	48
4.8.3.	Controlador DAC . . . . .	49
4.8.4.	FIFO . . . . .	50
4.9.	Resumen del capítulo . . . . .	51
<b>5.</b>	<b>Pruebas y resultados</b> . . . . .	<b>52</b>
5.0.1.	Procedimiento experimental . . . . .	52
5.0.2.	Resultados del experimento 1 . . . . .	57
5.0.3.	Resultados del experimento 2 . . . . .	62
5.0.4.	Resultados del experimento 3 . . . . .	63
5.1.	Resumen del capítulo . . . . .	65
<b>6.</b>	<b>Conclusiones y trabajo futuro</b> . . . . .	<b>66</b>
6.1.	Conclusiones . . . . .	66
6.2.	Trabajo futuro . . . . .	67

<b>Anexo</b>	<b>68</b>
<b>A. Código VHDL de la etapa de adquisición</b>	<b>69</b>
A.1. Módulo principal . . . . .	69
A.2. Módulo del decodificador . . . . .	74
A.3. Módulo de generación de las señales de reloj . . . . .	75
A.4. Módulo de la FIFO asíncrona . . . . .	79
A.4.1. Módulo de sincronización del apuntador de lectura . . . . .	81
A.4.2. Módulo de sincronización del apuntador de escritura . . . . .	82
A.4.3. Módulo de conversión a código Gray . . . . .	84
A.5. Módulo controlador del ADC . . . . .	85
A.6. Módulo controlador del DAC . . . . .	86
<b>Referencias bibliográficas</b>	<b>88</b>

# Índice de tablas

2.1. Cronología de las comunicaciones eléctricas. . . . .	8
2.2. Resumen de las tasas de transmisión de información por la red eléctrica. . . . .	9
2.3. Clasificación de los sistemas PLC. . . . .	10
2.4. Aplicaciones PLC y productos comerciales. . . . .	11
2.5. Especificaciones del estándar IEEE1901. . . . .	13
2.6. Trabajos relacionados. . . . .	16
5.1. Parámetros del transreceptor OFDM . . . . .	53
5.2. Parámetros del experimento . . . . .	56

# Índice de figuras

1.1. Frecuencia utilizada en la distribución de la energía eléctrica. . . . .	2
1.2. Módem PLC. . . . .	4
2.1. Regulación de frecuencias y límites de emisión para PLC en banda angosta. . .	12
3.1. Diagrama de bloques básico de un SDC. . . . .	18
3.2. Proyección de la constelación I-Q. . . . .	19
3.3. Constelación 16-QAM. . . . .	20
3.4. Ejemplo de un modulador 16-QAM. . . . .	21
3.5. Señal OFDM en el dominio del tiempo y la frecuencia. . . . .	22
3.6. Esquema general de un modulador OFDM. . . . .	24
3.7. Esquema general de un demodulador OFDM. . . . .	25
3.8. Interferencia ISI e ICI debidos a la respuesta del canal. . . . .	27
3.9. Intervalo de guarda/Prefijo cíclico. . . . .	27
3.10. Señal continua. . . . .	29
3.11. Señal discreta. . . . .	29
4.1. Arquitectura del módem propuesto. . . . .	34
4.2. Transmisor. . . . .	35
4.3. Receptor. . . . .	36
4.4. Circuito de acoplamiento. . . . .	38
4.5. Diagrama de bloques del TMS320C6416T DSK. . . . .	38
4.6. Módulo de evaluación Xilinx Spartan 3E. . . . .	39
4.7. Fases del ciclo de desarrollo en CCS. . . . .	40
4.8. Diagrama de bloques del módulo de adquisición. . . . .	47
4.9. Controlador EMIF. . . . .	48
4.10. Controlador ADC. . . . .	49
4.11. Controlador DAC. . . . .	50
4.12. Estructura de la FIFO asíncrona. . . . .	51
5.1. Diagrama general de conexiones del sistema. . . . .	52
5.2. Diagrama de conexiones para el experimento 1. . . . .	53
5.3. Señal OFDM de referencia. . . . .	54
5.4. Diagrama de conexiones para el experimento 2. . . . .	54
5.5. Diagrama de conexiones para el experimento 3. . . . .	55
5.6. Señal OFDM sin ruido. . . . .	56
5.7. Señal OFDM para una SNR de 0 dB. . . . .	57

5.8. Señal de correlación. . . . .	57
5.9. Gráfica de error y ganancia para el detector de símbolo piloto. . . . .	58
5.10. Señal del símbolo piloto. . . . .	58
5.11. Magnitud recibida del símbolo piloto. . . . .	59
5.12. Fase recibida del símbolo piloto. . . . .	59
5.13. Señal del símbolo piloto después de la sincronización. . . . .	60
5.14. Magnitud del símbolo piloto después de la sincronización. . . . .	60
5.15. Fase del símbolo piloto después de la sincronización. . . . .	60
5.16. Constelación QAM con sincronización burda y fina. . . . .	61
5.17. Tasa de error por bit para una SNR dada. . . . .	62
5.18. Error en muestras de la función de correlación para una SNR dada. . . . .	63
5.19. Error en muestras de la función de correlación para una SNR dada. . . . .	64
5.20. Comportamiento del <i>error</i> y la <i>ganancia</i> para una SNR dada. . . . .	65

# Glosario de abreviaturas

<b>A/D, ADC</b>	<i>Analog to Digital Converter.</i> Convertidor analógico-digital.
<b>ADSL</b>	<i>Asymmetric Digital Subscriber Line.</i> Línea de abonado digital asimétrica.
<b>ANSI</b>	<i>American National Standards Institute.</i> Instituto nacional americano de estándares.
<b>ASK</b>	<i>Amplitude Shift Keying.</i> Modulación por desplazamiento de amplitud.
<b>BER</b>	<i>Bit Error Rate.</i> Tasa de error por bit.
<b>CCS</b>	<i>Code Composer Studio.</i> Herramienta de desarrollo para los DSPs de Texas Instruments.
<b>CENELEC</b>	<i>Del francés, Comité Européen de Normalisation Electrotechnique.</i> Comité Europeo para la estandarización electrotécnica.
<b>CISPR</b>	<i>Del francés, Comité International Spécial des Perturbations Radioélectriques.</i> Comité internacional especial de perturbaciones radioeléctricas.
<b>COFDM</b>	<i>Coded OFDM.</i> OFDM codificado.
<b>CP</b>	<i>Cyclic Prefix.</i> Prefijo cíclico.
<b>D/A, DAC</b>	<i>Digital to Analog Converter.</i> Convertidor digital-analógico.
<b>DCM</b>	<i>Digital Clock Manager.</i> Manejador de reloj digital.
<b>DCSK</b>	<i>Differential Code Shift Keying.</i> Modulación por variación de código diferencial.
<b>DFT</b>	<i>Discrete Fourier Transform .</i> Transformada discreta de Fourier.
<b>DIF</b>	<i>Decimation In Frequency.</i> Decimación o diezmado en frecuencia.
<b>DIT</b>	<i>Decimation In Time.</i> Decimación o diezmado en tiempo.
<b>DMT</b>	<i>Discrete Multitone.</i> Multitono discreto.
<b>DSK</b>	<i>Development Software Kit.</i> Kit para el desarrollo de software.
<b>DSL</b>	<i>Digital Subscriber Line.</i> Línea de suscripción digital.

<b>DSP</b>	<i>Digital Signal Processor.</i> Procesador digital de señales.
<b>DSSS</b>	<i>Direct Sequence Spread Spectrum.</i> Espectro ensanchado por secuencia directa.
<b>EIA</b>	<i>Electronic Industries Alliance.</i> Alianza de industrias electrónicas.
<b>EMC</b>	<i>Electromagnetic Compatibility.</i> Compatibilidad electromagnética.
<b>EMIF</b>	<i>External Memory Interface.</i> Interfaz de memoria externa.
<b>ETSI</b>	<i>European Telecommunications Standards Institute.</i> Instituto europeo de estándares sobre telecomunicaciones.
<b>FCC</b>	<i>Federal Communications Comision.</i> Comisión federal de comunicaciones.
<b>FDM</b>	<i>Frequency Division Multiplexing.</i> Multiplexado por división de frecuencia.
<b>FEC</b>	<i>Forward Error Correction.</i> Corrección de errores hacia adelante.
<b>FFH</b>	<i>Fast Frequency Hopping.</i> Modulación por salto rápido de frecuencia.
<b>FFT</b>	<i>Fast Fourier Transform.</i> Transformada rápida de Fourier.
<b>FIFO</b>	<i>First Input First Output.</i> Primero en entrar, primero en salir.
<b>FPGA</b>	<i>Field Programmable Gate Array.</i> Arreglo de compuertas programables en campo.
<b>FSK</b>	<i>Frequency Shift Keying.</i> Modulación por desplazamiento de frecuencia.
<b>HV</b>	<i>High Voltage.</i> Alto voltaje.
<b>ICI</b>	<i>Intercarrier Interference.</i> Interferencia entre portadoras.
<b>IEC</b>	<i>International Electrotechnical Commission.</i> Comisión internacional electrotécnica.
<b>IEEE</b>	<i>Institute of Electrical and Electronics Engineers.</i> Instituto de ingenieros eléctricos y electrónicos.
<b>IFFT</b>	<i>Inverse Fast Fourier Transform.</i> Transformada inversa rápida de Fourier.
<b>IG</b>	<i>Intervalo de Guarda.</i>
<b>ISI</b>	<i>Itersymbol Interference.</i> Interferencia entre símbolos.
<b>ITU</b>	<i>International Telecommunications Union.</i> Union internacional de telecomunicaciones.
<b>JTAG</b>	<i>Joint Test Action Group.</i> Referido al estándar ampliamente adoptado para la depuración de aplicaciones empotradas.
<b>LDPC</b>	<i>Low Density Parity Check.</i> Códigos correctores de baja densidad.

<b>LLS</b>	<i>Linear Least Squares.</i> Recta que mejor se aproxima a una serie en el sentido de mínimos cuadrados.
<b>LS</b>	<i>Least Squares.</i> Criterio de mínimos cuadrados.
<b>LV</b>	<i>Low Voltage.</i> Bajo voltaje.
<b>MCM</b>	<i>Multicarrier Modulation.</i> Modulación multiportadora.
<b>ML</b>	<i>Maximum Likelihood.</i> Máxima probabilidad.
<b>MMSE</b>	<i>Minimum Mean Square Error.</i> Error cuadrático medio mínimo.
<b>MV</b>	<i>Medium Voltage.</i> Medio voltaje.
<b>OFDM</b>	<i>Orthogonal Frequency Division Multiplexing.</i> Multiplexado por división de frecuencias ortogonales.
<b>OPERA</b>	<i>Open PLC European Research Alliance.</i> Alianza europea de investigación sobre PLC abierta.
<b>PAM</b>	<i>Pulse Amplitude Modulation.</i> Modulación por amplitud de pulso.
<b>PLC</b>	<i>Power Line Communication.</i> Comunicaciones por la red eléctrica.
<b>PLT</b>	<i>Power Line Telecommunications.</i> Telecomunicaciones por la red eléctrica.
<b>PROM</b>	<i>Programable Read Only Memory.</i> Memoria programable de sólo lectura.
<b>PSK</b>	<i>Phase Shift Keying.</i> Modulación por desplazamiento de fase.
<b>QAM</b>	<i>Quadrature Amplitude Modulation.</i> Modulación digital de amplitud en cuadratura.
<b>QPSK</b>	<i>Quadrature Phase Shift Keying.</i> Modulación por desplazamiento de fase en cuadratura.
<b>RS</b>	<i>Reed Solomon.</i> Nombre de un método corrector de errores.
<b>RTDX</b>	<i>Real Time Data Exchange.</i> Intercambio de datos en tiempo real.
<b>SDC</b>	<i>Sistema digital de comunicaciones</i>
<b>SDRAM</b>	<i>Synchronous Dynamic Random Access Memory.</i> Memoria dinámica de acceso aleatorio con interfaz síncrona.
<b>SFSK</b>	<i>Spread Frequency Shift Keying.</i> Similar a FSK pero con las portadoras más separadas.
<b>SPI</b>	<i>Serial Peripheral Interface.</i> Periférico de interfaz serie.
<b>TCM</b>	<i>Trellis Coded Modulation.</i> Modulación con codificación de Trellis.

<b>TIA</b>	<i>Telecommunications Industry Association.</i> Asociación de la industria de telecomunicaciones
<b>USB</b>	<i>Universal Serial Bus.</i> Bus serial universal.
<b>VCO</b>	<i>Voltage Control Oscillator.</i> Oscilador controlado por voltaje.
<b>xDSL</b>	<i>Referido a las diferentes tecnologías DSL.</i>

# Resumen

La red eléctrica ha ganado gran interés en los últimos años como medio para la transmisión de información. Pero debido a que la red eléctrica no fue pensada para transportar señales de comunicación, sino para la distribución de la energía eléctrica, presenta características que lo convierten en un medio muy *hostil* para fines de transmisión de datos. Por lo tanto, este trabajo se centra en la problemática de diseñar una plataforma de hardware y software que implemente técnicas avanzadas de procesamiento digital de señales, y de comunicaciones digitales, adecuadas para su implantación en un *módem* para la comunicación por la red eléctrica. Específicamente, se detalla la implementación de OFDM como técnica de modulación; ya que junto a DMT, han sido consideradas como las técnicas más adecuadas para resolver el problema de la transmisión de información por la red eléctrica. Esta implementación de OFDM utiliza la FFT como núcleo principal para los procesos de modulación y demodulación. La FFT programada es para un máximo de 1024 puntos, pero con posibilidades de crecer, siempre y cuando la cantidad de memoria del dispositivo utilizado lo permita. Para eficientar el uso de la memoria, la FFT realiza los cálculos *in-place*, lo cual significa que el mismo buffer de entrada es utilizado como buffer de salida. Además, los factores de fase son mapeados a memoria, lo cual acelera el tiempo de cómputo. Por otro lado, la implementación de OFDM se basa en el uso de *símbolos piloto* para los procesos de detección del inicio de trama, sincronización fina y estimación de canal. La detección de inicio de trama utiliza la correlación del prefijo cíclico, lo cual se conoce comúnmente como proceso de sincronización burda. El proceso de sincronización fina llevado a cabo corresponde a un proceso de sincronización en tiempo, y se basa en la observación de los desplazamientos de fase de cada una de las subportadoras del símbolo piloto. Para el mapeo de bits en cada subportadora, se eligió utilizar QAM, con posibilidad de elegir entre 4-QAM, 16-QAM, 64-QAM, 128-QAM y 256-QAM.

Los algoritmos que aquí se presentan son programados en lenguaje C, empleando al procesador digital de señales (DSP) TMS320C6416T como unidad de procesamiento. Por otra parte, se presenta el diseño en VHDL basado en el FPGA Spartan 3E de Xilinx de una etapa de adquisición, que virtualmente permitiría pruebas a frecuencias del rango de los 5 a 30MHz (pruebas en banda ancha).

Las pruebas y resultados que se presentan, fueron obtenidos utilizando MATLAB como herramienta de simulación y al TMS320C6416T DSK como herramienta para la experimentación. Las pruebas son enfocadas principalmente a evaluar los procesos involucrados en la recepción de información, pues, por ejemplo, el proceso de sincronización es una de las partes más susceptibles a errores en un sistema basado en OFDM.

El diseño que se expone en este trabajo, tiene el potencial de ser flexible, ya que facilita su uso como plataforma para la investigación de nuevas técnicas y algoritmos que ayuden a mejorar los procesos implicados en la transmisión de información por la red eléctrica. Además, los códigos en lenguaje C y VHDL facilitan su migración a otros dispositivos con recursos de hardware superiores cuando el diseño así lo requiera.

# Abstract

Lately the power line has gained great interest for information transmission. Because the power grid was not designed for transport communication signals but for power distribution, it presents characteristics that transform it into *hostile* environment for data transmission. Therefore, this work focuses on the problematic of designing a hardware and software platform that implements advanced techniques of digital signal processing and digital communications suitable for implantation on power line communication modem. Specifically, it details the implementation of OFDM as a modulation technique, since, along with DMT they have been considered as the most appropriate techniques to solve the problem of information transmission over the power line. This implementation has been using the FFT as a core for the modulation and demodulation processes. The FFT is programmed for a maximum of 1024 points, with the possibility to increase if and when the quantity of the device memory allows it. For efficient use of memory, the FFT performs computations *in-place*, this means the input buffer is the same for the output. In addition, the phase factors are mapped to memory, speeding the computation time. On the other hand, the implementation of OFDM is based on the use of *pilot symbols* for start frame detection, fine synchronization and channel estimation processes. The start frame detection uses the correlation of the cyclic prefix, which is commonly known as coarse synchronization process. The fine synchronization process performed corresponds to a time synchronization process and is based on the observation of pilot symbol sub-carriers phase shifts. QAM is selected for mapping of bits in each sub-carrier, with option to choose between 4-QAM, 16-QAM, 64-QAM, 128-QAM and 256-QAM.

The algorithms that are presented in this work are programmed in C language using a digital signal processor (DSP) TMS3206416T as a processing unit. Moreover, the design of an acquisition module based on the Spartan 3E FPGA from Xilinx is presented. This module virtually allows trials using frequencies in the range of 5 to 30MHz (broadband trials).

Test and results presented were obtained using MATLAB as a simulation tool and the TMS320C6416T DSK as an experimentation tool. The tests were focused primarily to evaluate the processes involved in receiving information. Then, for example, the synchronization process is one of the most susceptible to errors in an OFDM based system.

The design featured in this work has the potential to be flexible, which enables using it as an investigation platform on new techniques and algorithms to help improve the processes involved on power line communication. In addition, the codes in C and VHDL language facilitate their migration to other devices with higher hardware resources when required by the design.

# Capítulo 1

## Introducción

El siguiente capítulo introduce la problemática abordada en el presente trabajo. Se inicia exponiendo los motivos y antecedentes que han dado origen a la idea de utilizar la red eléctrica para propósitos de comunicación; continuando con la descripción del problema planteado, su justificación, los objetivos perseguidos, los alcances y las aportaciones esperadas. La parte final del capítulo versa brevemente sobre la organización del actual documento.

### 1.1. La red eléctrica como medio de comunicación

La creciente demanda de servicios de comunicación de alta velocidad ha forzado el uso, para propósitos de transmisión de datos, de medios de comunicación que originalmente no fueron diseñados para tales propósitos, tal es el caso, por ejemplo, de la red eléctrica y telefonía.

Prácticamente una red eléctrica se puede encontrar en cualquier hogar, oficina u organización. Esto hace posible pensar en un medio de comunicación que permita compartir grandes cantidades de información entre diferentes usuarios y lugares. Tal información podría ser, por ejemplo, para el intercambio de datos multimedia (audio, video, juegos), para el uso de dispositivos comunes (impresoras, scanners), y por supuesto para la conexión a Internet. Pero debido a que la red eléctrica no fue pensada para propósitos de comunicación, presenta características que hacen de éste, un medio muy hostil para la transmisión de información [22].

La tecnología PLC (Power Line Communications por sus siglas en inglés) permite el uso de la red eléctrica para propósitos de comunicación y, hoy, también de servicios de comunicación en banda ancha (Broadband Power Line Communications, BPL). Aunque la tecnología PLC existe desde 1930, no había sido considerada como un medio factible para la comunicación, debido a su baja velocidad, poca funcionalidad, y alto costo. Pero gracias a los recientes avances hechos en la tecnología, y sobre todo en las técnicas de modulación, es permisible que la red eléctrica pueda ser considerada como un medio de comunicación muy atractivo y rentable [45].

Para que la red eléctrica pueda servir como medio de comunicación se requiere de un dispositivo que permita el acceso al medio de comunicación, dicho dispositivo es denominado módem PLC; encargado de que las señales de datos puedan ser transmitidas por las redes de energía eléctrica. Un módem PLC convierte una señal de datos desde un dispositivo convencional de comunicación, por ejemplo, una computadora, un teléfono, y demás, en una forma que es apta para su transmisión por la red eléctrica. Así, los módems PLC, representan

un equipo de comunicación PLC específico, que proporciona la interfaz necesaria para la interconexión de varios dispositivos de comunicación sobre la red de energía eléctrica.

En México, la infraestructura eléctrica se encuentra instalada en más lugares que cualquier tecnología de comunicación digital, como por ejemplo telefonía o televisión por cable. Esto hace atractiva a la tecnología PLC, ya que en teoría, permitiría proporcionar el servicio de Internet y servicios digitales de alta velocidad a mayores segmentos de la población que básicamente cuentan con el servicio eléctrico.

## 1.2. Antecedentes

El uso de la red eléctrica para propósitos de comunicación no es una idea nueva, ya que anteriormente este medio había sido utilizado para aplicaciones de control de baja velocidad, como por ejemplo, la lectura automática de medidores y el control de líneas eléctricas. Posteriormente, las propias compañías eléctricas empezaron a utilizar su infraestructura interna para el intercambio de datos de modo interno [30].

A finales de los 90, gracias a los avances tecnológicos, es posible alcanzar velocidades de transmisión del orden de los Megabits. Y hoy, tras varios años de pruebas piloto e investigaciones en este campo, se puede pensar en la tecnología PLC como una alternativa fiable a los sistemas de comunicación existentes como ADSL, Cable. De hecho, esta tecnología está plenamente en operación en países como Estados Unidos, Alemania, Gran Bretaña y Francia [20].

La habilidad de transmitir datos sobre la red eléctrica, en muchos aspectos se basa en el concepto por el cual las compañías telefónicas observaron que el cable de par trenzado estándar instalado en los hogares y oficinas podría ser utilizado para transmitir datos. Reconocieron que las líneas telefónicas de par trenzado instaladas, eran capaces de soportar rangos de frecuencia de hasta 1MHz y, debido a que una conversación telefónica sólo utiliza 3KHz de ancho de banda, hace posible que la transmisión de datos pueda ocurrir a frecuencias más allá de las utilizadas para la transmisión de voz, por medio de la técnica de modulación referida como multiplexado por división de frecuencia (FDM). Como se puede observar en

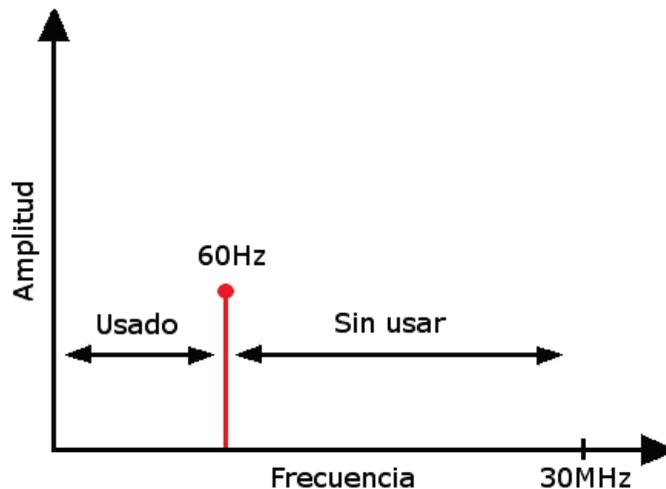


Figura 1.1: Frecuencia utilizada en la distribución de la energía eléctrica.

la figura 1.1, para el caso de la redes de energía eléctrica, la corriente alterna estándar es transmitida en una frecuencia de 60Hz en Norte América y en 50Hz en Europa y en muchos otros lugares alrededor del mundo. Esto significa que, al igual que para la red de telefonía, se tiene casi la totalidad del rango de frecuencias disponibles para ser utilizadas en otros propósitos, incluyendo la transmisión de datos.

Aunque existe la posibilidad de utilizar la red eléctrica para propósitos de transmisión de datos, hay muchas razones por las que un canal de comunicación empleando la línea eléctrica como medio puede presentar problemas. Entre las razones principales se pueden mencionar las siguientes [37]:

- Atenuación variante con la frecuencia y el tiempo.
- Dependiente de la localización, topología de la red y terminación de cargas.
- Interferencia elevada debido a cargas ruidosas.
- Alta cantidad de ruido de fondo no blanco.
- Diferentes tipos de ruido impulsivo.
- Cuestiones de compatibilidad electromagnética (EMC) que limitan la cantidad de potencia disponible para la transmisión.

De lo anterior, se pueden resumir en cuatro los factores principales que juegan un papel importante en la selección de la técnica adecuada de modulación para un sistema PLC:

- La naturaleza del canal selectivo en frecuencia y variante en el tiempo.
- La presencia de ruido y perturbaciones impulsivas.
- El aspecto multiusuario del sistema, con muchos clientes que solicitan el acceso al mismo medio físico simultáneamente.
- Las limitaciones regulatorias con respecto a la compatibilidad electromagnética (EMC).

Las técnicas de modulación multiportadora han demostrado ser las técnicas de transmisión más adecuadas para la comunicación por la red eléctrica y los principales fabricantes de módems basan su capa física en OFDM, que junto a DMT (Discrete MultiTone), son las dos estrategias multiportadora más extendidas.

### 1.3. Planteamiento del problema

La red eléctrica es un medio que originalmente fue pensado para la distribución de la energía eléctrica y no para propósitos de transmisión de información. Debido a esto, exhibe características que hacen de éste, un medio muy hostil cuando se utiliza como medio de comunicación. Por lo tanto, como se puede apreciar en la figura 1.2, el dispositivo PLC es el encargado de adecuar las señales de información para que éstas puedan viajar por el medio eléctrico; aplicando técnicas avanzadas de comunicaciones digitales y de procesamiento digital de señales. Sin embargo, aunque en el mercado existen dispositivos que resuelven el problema de transmitir datos por la red eléctrica, su flexibilidad se limita a la aplicación

específica para la cual fueron diseñados, lo cual, implica un obstáculo si se pretende utilizar con fines de investigación. Por lo anterior, en el presente trabajo, se aborda la problemática de construir una plataforma de hardware y software basado en el procesador digital de señales TMS320C6416T de Texas Instruments, que permitirá realizar pruebas de las técnicas más adecuadas para un módem PLC.

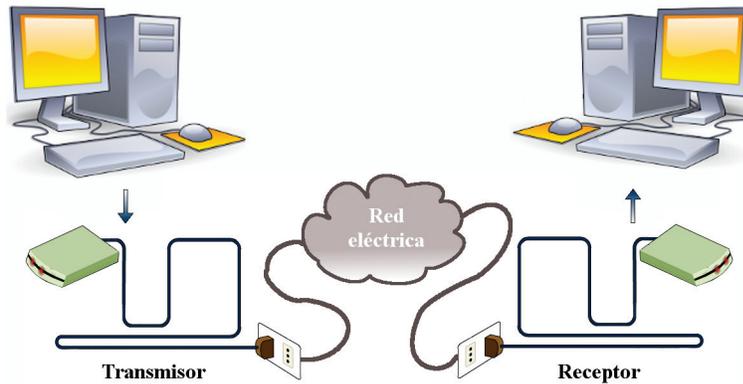


Figura 1.2: Módem PLC.

## 1.4. Justificación

Es de gran interés realizar investigación en el campo de la tecnología PLC para descubrir o desarrollar nuevas técnicas de procesamiento digital de señales, así como de comunicaciones digitales, que permitan hacer frente a las dificultades impuestas por el medio eléctrico; posibilitando de esta manera que la red eléctrica pueda ser considerada como una opción viable frente a las tecnologías de transmisión de información como los sistemas inalámbricos y las tecnologías xDSL, por mencionar algunos. No obstante, para poder realizar investigaciones en el campo PLC, se requiere de una plataforma, tanto de hardware como de software, que ayude en la prueba de algoritmos en condiciones lo más cercanas posibles a un ambiente real.

## 1.5. Hipótesis

La implementación de la plataforma de hardware, permitirá comprobar que la técnica de modulación multiportadora OFDM, es la elección más adecuada para un ambiente PLC.

## 1.6. Objetivo general

Diseñar e implementar una plataforma a nivel de hardware y software para el desarrollo de un módem para la comunicación por la red eléctrica de baja tensión, en un entorno de simulación.

## 1.7. Objetivos específicos

- Realizar la construcción de la plataforma a nivel de hardware para pruebas.

- Programar la plataforma de hardware para la comunicación por la red eléctrica de baja tensión basado en OFDM, en un entorno de simulación.
- Realizar pruebas de desempeño.

## 1.8. Alcance

En esta tesis, se presenta una plataforma en hardware y software que permita realizar pruebas de los bloques más característicos de un sistema basado en OFDM, aplicado al desarrollo de un módem PLC. Para el desarrollo de la plataforma, se hace uso de herramientas y técnicas de procesamiento digital de señales avanzadas, que han demostrado ser las más adecuadas al problema de la transmisión de información por el medio eléctrico.

La propuesta contempla dos etapas para el desarrollo de la plataforma, la etapa de hardware y la etapa de software.

La etapa de hardware contempla los siguientes módulos:

- Módulo de procesamiento.
- Módulo de recepción y transmisión de información.
- Módulo de adquisición.

La etapa de software contempla los siguientes módulos:

- Módulo de recepción y transmisión de información.
- Módulo de modulación y demodulación OFDM.
- Módulo de sincronización.

## 1.9. Aportaciones

- Una plataforma de hardware y software que permita el desarrollo de pruebas de un sistema basado en OFDM aplicado al campo de la tecnología PLC.
- Un módem PLC basado en OFDM.

## 1.10. Organización de la tesis

El presente trabajo se encuentra estructurado de 6 capítulos. El primer capítulo contiene la presente introducción.

En el capítulo 2, se hace una revisión del estado actual de las técnicas empleadas en el desarrollo de dispositivos para la comunicación por la red eléctrica.

En el capítulo 3, se presenta el fundamento teórico utilizado para el desarrollo del módem PLC.

En el capítulo 4, se describe el desarrollo y los bloques funcionales del módem propuesto.

En el capítulo 5, se presentan las pruebas realizadas y los resultados obtenidos en el desarrollo del módem.

Finalmente, en el capítulo 6 se presentan las conclusiones y las posibles mejoras al diseño desarrollado.

## 1.11. Resumen del capítulo

En este capítulo, se ha comentado que la idea de utilizar la red eléctrica como medio de comunicación, surge con la necesidad de encontrar un medio de comunicación alternativo a las tecnologías existentes, con el fin de hacer frente a la demanda de servicios de comunicación de alta velocidad. También se ha expuesto que la red eléctrica es un candidato factible, a causa de que se encuentra prácticamente omnipresente en cualquier instalación, con posibilidad de ser utilizada para compartir información entre diferentes usuarios y lugares.

De igual manera, se ha mencionado que se denomina PLC a la tecnología que utiliza la red eléctrica como medio de comunicación, y, que la idea fundamental se basa en el hecho de que la señal con la que se distribuye la energía eléctrica tiene una determinada frecuencia. Lo cual favorece que puedan ser utilizadas frecuencias diferentes para transmitir información.

Asimismo, se ha manifestado que la red eléctrica exhibe características que dificultan su uso como medio de comunicación; dado que no fué pensada para fines de transmisión de información. Y si bien, existen en el mercado dispositivos que resuelven el problema de transmisión de información por la red eléctrica, no ofrecen la flexibilidad necesaria para evaluar e implementar nuevas técnicas que sirvan para mejorar los procesos involucrados durante la transmisión de información por la red eléctrica. Por tal motivo, en este trabajo, se encara la necesidad de construir una plataforma de hardware y software, que permita investigar y descubrir, técnicas adecuadas de procesamiento digital de señales y de comunicaciones digitales, que puedan ser aplicadas en un dispositivo para la comunicación por la red eléctrica.

## Capítulo 2

# Estado del arte

En este capítulo, se hace la revisión de los aspectos más relevantes de la tecnología PLC. Iniciando con el surgimiento de las comunicaciones eléctricas y siguiendo con una breve descripción de la evolución en el uso del medio eléctrico para propósitos de transmisión de datos. Los aspectos que se abordan son: la clasificación y áreas de aplicación de la tecnología PLC, estándares existentes, los problemas inherentes de utilizar el medio eléctrico como canal de comunicación y algunas de las soluciones que se han ideado para resolver dichos problemas. También se hace la revisión del trabajo relacionado con el desarrollo de dispositivos para la comunicación por la red eléctrica.

### 2.1. Comunicaciones eléctricas

La era de las comunicaciones eléctricas nace con el desarrollo de la telegrafía, y es notable que esta primera forma de comunicación eléctrica fuera un sistema de comunicación digital. El telégrafo eléctrico, fue desarrollado por Samuel Morse y fue mostrado en 1837; y el primer mensaje telegráfico *Attention, the Universe! By kingdoms, right wheel!* fue enviado por Morse sobre una línea de 16Km en 1838.

Ahora, a un poco menos de dos siglos después, es de sorprenderse que la tecnología de las comunicaciones ha evolucionado en una forma tal, que es posible observar desde la tierra por medio de la televisión a un astronauta trabajando en el espacio, envía o recibir información multimedia por medio de un correo electrónico y navegar por Internet para búsqueda de información o simplemente por entretenimiento. Ciertamente, grandes avances han sido realizados desde los días de Morse, pero es igualmente cierto que en las décadas venideras habrá nuevos avances en la tecnología de las comunicaciones.

La tabla 2.1 presenta brevemente los eventos que marcaron pauta en el desarrollo de las comunicaciones eléctricas y por tanto de las comunicaciones digitales tal como se conocen actualmente[5].

Tabla 2.1: Cronología de las comunicaciones eléctricas.

Año	Evento
1800 - 1837	<i>Desarrollos preliminares.</i> Volta descubre la batería primaria; los tratados matemáticos por Fourier, Cauchy y Laplace; experimentos sobre electricidad y magnetismo por Oersted, Ampere, Faraday y Henry; leyes de Ohm (1826); inicios de los sistemas de telégrafos por Gauss, Weber y Wheatstone.
1838 - 1866	<i>Telegrafía.</i> Morse perfecciona su sistema; Steinheil encuentra que la tierra puede ser usada para una trayectoria actual; inician servicios comerciales (1844); se divisan técnicas de multiplexado; William Thomson (Lord Kelvin) calcula la respuesta al pulso de una línea telegráfica (1855); se instalan cables trasatlánticos por Cyrus Field y asociados.
1845	Se enuncian las leyes de circuitos de Kirchhoff.
1864	Las ecuaciones de Maxwell predicen la radiación electromagnética.
1876 - 1899	<i>Telefonía.</i> Transductor acústico perfeccionado por Alexander Graham Bell, después de intentos anteriores por Reis; primer intercambio telefónico en New Haven con 8 líneas (1878); micrófono de carbono de Edison.
1887 - 1907	<i>Telegrafía inalámbrica.</i> Heinrich Hertz verifica la teoría de Maxwell; demostraciones por Marconi y Popov; Marconi patenta un sistema telegráfico inalámbrico completo (1897); la teoría de circuitos sintonizadores desarrollado por Sir Oliver Lodge; comienzan servicios comerciales, incluyendo sistemas buque-tierra y trasatlánticos.
1892 - 1899	Publicaciones de Oliver Heaviside sobre cálculo operacional, circuitos y electromagnetismo.
1904 - 1920	<i>Comunicaciones electrónicas.</i> Lee De Forest inventa el Audion (triodo), basado en el diodo de Fleming; tipos de filtros básicos divisados por G. A. Campbell y otros; experimentos con radiodifusión AM; línea telefónica transcontinental con repetidores electrónicos realizados por la Bell System (1915); introducción de la compañía telefónica multiplexada; E. H. Armstrong perfecciona el receptor de radio superheterodino (1918); primera estación de radiodifusión comercial, KDKA, Pittsburgh.
1920 - 1928	<i>Teoría de transmisión.</i> Artículos destacados sobre teoría de transmisión de señales y ruido por J. R. Carson, H. Nyquist, J. B. Johnson y R. V. L Hartley.
1934	H. S. Black desarrolla el amplificador de realimentación negativa.
1936	El artículo de Armstrong expone el caso para radio FM.
1937	Alec Reeves concibe la modulación por codificación de pulso (PCM).
1938 - 1945	<i>Segunda Guerra Mundial.</i> Se desarrollan los sistemas de Radar y Microondas; FM utilizado extensivamente para comunicaciones militares; mejoras en electrónica, hardware y teoría en todas las áreas.
1944 - 1947	<i>Teoría de comunicaciones estáticas.</i> Rice desarrolla una representación matemática del ruido; Wiener, Kolmogorov y Kotelnikov aplican métodos estáticos a la detección de señales.
1948 - 1950	<i>Teoría de la información y codificación.</i> C. E. Shannon publica los artículos fundamentales de la teoría de la información; Hamming y Golay divisan códigos correctores de errores.
1948 - 1951	Dispositivos transistor inventados por Bardeen, Brattain y Shockley.
1950	Multiplexado por división de tiempo aplicado a la telefonía.
1955	J. R. Pierce propone sistemas de comunicaciones satelitales.
1962	Las comunicaciones satelitales comienzan con el Telstar I.
1962 - 1966	<i>Comunicaciones digitales de alta velocidad.</i> Servicios de transmisión de datos ofrecidos comercialmente; canales de banda ancha diseñados para señalización digital; modulación por codificación de pulso resulta factible para la transmisión de voz y TV; grandes avances en teoría e implementación de transmisión digital, incluyendo métodos de codificación de control de errores por Viterbi y otros, y el desarrollo de equalización adaptativa por Lucky y colaboradores.
1966 - 1975	<i>Sistemas de comunicación de banda ancha.</i> Sistemas de TV Cable; se hacen disponibles servicios de retransmisión satelital; enlaces ópticos utilizando láser y fibra óptica.
1969	Se crea ARPANET (precursor de Internet).
1971	Intel desarrolla primer microprocesador en un solo circuito integrado.
1990 - 2000	<i>Sistemas de comunicaciones digitales.</i> Sistemas de procesamiento digital de señales y comunicaciones en electrodomésticos; receptores con sintonizadores digitales; Sistemas Direct Sequence Spread Spectrum (DSSS); computadoras portátiles.

## 2.2. Tecnología PLC

La tecnología PLC tiene una larga historia, ya que inicialmente la red eléctrica fue empleada con propósitos de control, mantenimiento y tarificación. Por ejemplo, en 1838 se registra como la primera medición remota de la energía eléctrica y en 1897 fue propuesta en el Reino Unido la primera patente sobre señalización de la red eléctrica. En 1905 fueron patentadas aplicaciones en Estados Unidos, y en 1913 tomó lugar la primera producción comercial de medidores electromecánicos. Pero mucho más tarde por 1980, fueron propuestas técnicas sofisticadas de control de errores dentro del hardware de los modems PLC[30]. La primera aplicación que involucra la transmisión de datos sobre la red eléctrica fue hecha para propósitos de protección del sistema de distribución en caso de falla. Hasta hoy, la protección de las líneas de transmisión, permanece entre las principales funciones de la tecnología PLC[38].

Históricamente, las organizaciones públicas que distribuyen la energía eléctrica han sido una importante fuerza impulsora en el desarrollo de la tecnología PLC. Una primera motivación para la comunicación por la red eléctrica está relacionado con el manejo de carga. Una segunda importante motivación ha sido facilitar la lectura de medidores a distancia. Potencialmente éste puede incluir no solamente la medición de la electricidad, sino que también, la medición del agua, gas y temperatura. Es interesante notar que un estudio inglés mostró que una persona que realiza la medición de la energía eléctrica, solamente puede alcanzar una tasa promedio de información de tan solo 1 bit/s[38]. Esto es, en efecto, muy bajo comparado con lo que se puede alcanzar con los sistemas PLC modernos. La tabla 2.2 muestra la evolución de las tasas de transmisión de información por la red eléctrica.

Tabla 2.2: Resumen de las tasas de transmisión de información por la red eléctrica.

Nombre	Periodo	Servicios	Tasas de transmisión	Especificaciones técnicas
CTP: Transmisión de portadora por la red eléctrica	Década de los 20	Telemetría y control supervisado	Tasa de bits baja	Frecuencia: 15 – 500KHz Medio: HV Modulación: AM, FM Potencia trans.: < 10W
Control remoto	Década de los 30	Servicios de telemetría	Tasa de bits baja	Frecuencia: 100 – 900Hz Medio: MV, LV Forma de trans.: Simplex
RCS: Señalización de portadora	Década de los 70	Administración de la demanda eléctrica y lectura de medidores	Ligero incremento en la tasa de transmisión de bits	Frecuencia: 5 – 500KHz Medio: MV, HV Distancia: < 4Km
SCADA: Control supervisado y adquisición de datos	Finales de los 80 y principio de los 90	Transmisión de voz	Incremento de la tasa de transmisión	Frecuencia: 5 – 1500KHz Modulación: ASK, FSK Forma de trans.: Dúplex Potencia: < 10W Distancia: < 4Km
PLC/PLT	1997	Voz, datos e Internet	HomePlug V1: 14Mbps	Frecuencia: 100 – 2000KHz Medio: MV Modulación: OFDM, DSSS Forma de trans.: Dúplex Potencia: < 10W Distancia: < 4Km

Continúa en la siguiente página...

Tabla 2.2 – Continuación

Nombre	Periodo	Servicios	Tasas de transmisión	Especificaciones técnicas
PLC/PLT de banda ancha	2005	Servicios de triple play: Voz, Internet y video	HomePlug Turbo: 85Mbps AV: 200Mbps	Frecuencia: 1 – 30MHz Medio: LV Modulación: BPSK, QPSK, 8, 16, 64, 256 y 4096-QAM para la capa física de OFDM empleando la FFT y BPSK, 4, 8, 16 y 32 PAM para la capa física de OFDM empleando ondeletas Forma de trans.: Half Dúplex Distancia: < 200m

## 2.3. Clasificación y áreas de aplicación

Los sistemas PLC pueden ser clasificados en dos tipos. Sistemas de banda angosta y sistemas de banda ancha. Usualmente los sistemas PLC de banda angosta se refieren a comunicaciones digitales de baja velocidad que utilizan las bandas de frecuencia por debajo de los 150KHz en Europa (siguiendo el estándar CENELEC: 3-148.5KHz), y debajo de los 450KHz en Estados Unidos y algunos países del extremo oriente. Por otro lado, los sistemas PLC de banda ancha utilizan una banda de frecuencia mucho más amplia, con la frecuencia más baja típicamente de 1MHz, y con la frecuencia más alta típicamente de hasta 30 ó 60MHz. También se han hecho investigaciones e implementaciones con frecuencias mucho más altas, incluso de cientos de MHz[39]. La tabla 2.3 muestra con más detalle la clasificación anterior.

Tabla 2.3: Clasificación de los sistemas PLC.

	Banda angosta tasa de transmisión baja	Banda angosta tasa de transmisión alta	Banda ancha
Rango de Frecuencia	9 – 148,5KHz	9 – 500KHz Bandas: A 9 – 95KHz B 95 – 125KHz BCD 95 – 148,5KHz Otras Bandas	1 – 60MHz
Tasa de transmisión	< 10Kbps	50Kbps < .. < 1Mbps	> 10Mbps
Tecnología	FSK, BPSK, FFH, SFSK, DCSK	OFDM, MCM, cod. diferencial	MCM/COFDM, carga de bits
FEC	Ninguno o bajo	Fuerte. Diseñado para una alta confiabilidad	Medio. Diseñado para un máx. rendimiento
Aplicaciones	Lectura automática de medidores Redes de área local	Manejo de energía Medición automática	Última milla Internet Voz sobre IP Televisión de alta definición
Compañías, organizaciones	Busch Jaeger, Echelon Görlitz, Ytran, Renasas, AMI solutions	ADD Group, iAd, Maxim, Prime (ADD, Current Group, Landis & Gyr, STMicroelectronics...)	Amperion, Current, DS2, Homeplug, Mitsubishi, OPERA, Panasonic, Spidcom

Por otro lado, son tres las principales áreas de aplicación de PLC[49, 29] y se resumen en la tabla 2.4.

Tabla 2.4: Aplicaciones PLC y productos comerciales.

PLC	Aplicaciones	Características	Disponibilidad comercial
Servicios públicos	Transmisión de voz, señales de protección, señales de control, monitoreo de estados, etc., para protección de líneas de transmisión de alto voltaje	Líneas de alto voltaje; banda angosta; baja tasa de datos hasta 64Kbps	ABB, RFL
Acceso	Entrega de servicios de banda ancha, por ejemplo, Internet a cada hogar u oficina, en lugar de DSL o cable	Líneas de medio o bajo voltaje; banda ancha; tasa de datos hasta de 200Mbps	Ambient, Ileo
Interiores	LANs en edificios (banda ancha); automatización del hogar (banda angosta)	Líneas de bajo voltaje; banda ancha/banda angosta; tasa de datos hasta de 200Mbps	Itellon, Echelon

La mayoría de los esfuerzos de investigación se han concentrado en los diferentes esquemas de modulación y codificación para el ambiente PLC. El factor más importante que ha retardado la instalación y uso de PLC (especialmente el uso de comunicaciones en banda ancha), es el lento desarrollo de estándares aceptables por todos los participantes. Los obstáculos clave han sido los acuerdos sobre los límites de emisión.

Los organismos más importantes que trabajan sobre la estandarización en PLC son: ETSI, CENELEC, IEC, ANSI, CISPR, IEEE, EIA, TIA, ITU y la FCC. Los temas cubiertos en sus estándares son: compatibilidad electromagnética, técnicas de medición y prueba de emisión e inmunidad electromagnética, seguridad, procedimientos de prueba básicos del ambiente, redes de área local y acceso a Internet [12].

En este trabajo, se pretende obtener un dispositivo capaz de operar en los rangos de frecuencia correspondiente a banda ancha (5-30MHz), con una tasa de transmisión superior a los 10Mbps; utilizando las redes de baja tensión al interior de los edificios como canal de comunicación. En el diseño no se contempla algún método de detección y corrección de errores. La principal aplicación pensada para el dispositivo es como herramienta para la investigación en el área de la tecnología PLC. Aunque su potencial aplicación contempla la transmisión de datos, voz y video.

## 2.4. Estándares existentes

Entre los estándares que existen actualmente para PLC se pueden dividir en dos grupos. Los estándares aplicables a sistemas diseñados para trabajar en banda angosta y los concernientes a sistemas diseñados para trabajar en banda ancha. La figura 2.1 muestra algunos de los estándares existentes para PLC en banda angosta.

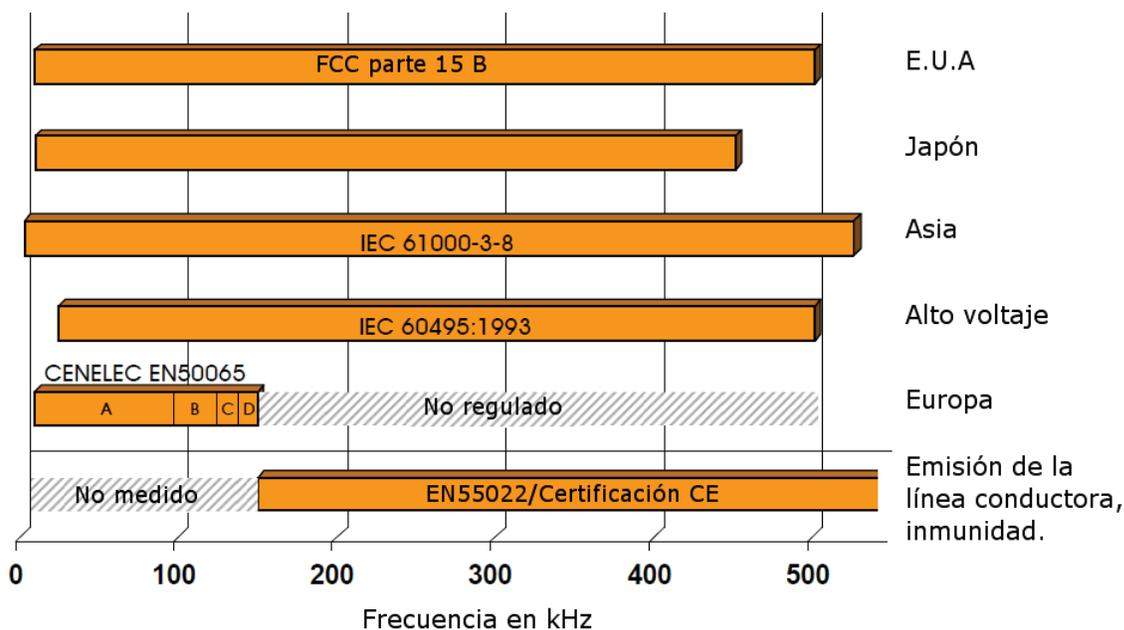


Figura 2.1: Regulación de frecuencias y límites de emisión para PLC en banda angosta.

Para el caso de sistemas PLC de banda ancha (también denominados sistemas BPL), existen grupos y organizaciones líderes en la creación de sus propios estándares. Entre éstos se encuentra OPERA, IEEE y la HomePlug Powerline Alliance[25].

OPERA es un consorcio que engloba a 37 organizaciones. Entre éstas se incluyen compañías públicas de servicio eléctrico, manufactureras de equipo PLC y universidades. Es un proyecto de investigación y desarrollo (R&D) financiado desde la Comisión Europea para crear y promover especificaciones globales abiertas para comunicaciones por la red eléctrica de bajo costo, alta velocidad y alto desempeño [25].

IEEE es otro organismo que ha trabajado activamente en la estandarización de banda ancha sobre la red eléctrica. Entre los estándares actuales para BPL se encuentran IEEE P1901, IEEE P1675 e IEEE P1775. Los párrafos siguientes describen brevemente cada uno de estos estándares.

- IEEE P1901. Fue publicado el 31 de diciembre del 2010 y se enfoca en el balance y uso eficiente del canal de comunicación por la red eléctrica para toda clase de dispositivos BPL, definiendo mecanismos detallados para la coexistencia e interoperabilidad de diferentes dispositivos BPL, asegurando que sea proporcionado el ancho de banda y la calidad de servicio deseado. Contiene las especificaciones para la capa MAC y para la capa PHY de acuerdo al modelo OSI. El estándar incluye dos opciones para la capa PHY. La primera opción se basa en el uso de la FFT para la modulación OFDM, con un esquema de corrección de errores hacia adelante (FEC) basado en un *Turbo Código Convolutional*. La segunda opción se basa en el uso de *ondeletas* para la modulación OFDM. Con un esquema obligatorio de corrección de errores hacia adelante basado en la concatenación de un código *Reed Solomon* (RS) y un *código Convolutional*; con opción de usar un *código de baja densidad* (LDPC, *Low Density Parity Check*) [52].

La tabla 2.5 resume algunas de las especificaciones técnicas de la capa física para este estándar.

Tabla 2.5: Especificaciones del estándar IEEE1901.

<b>Capa PHY OFDM-FFT</b>	
Método de comunicación	OFDM basado en la FFT
Puntos FFT	3072, 6144
Frecuencia de muestreo (MHz), respectivamente	75, 150
Longitud de símbolo ( $\mu s$ )	40.96
Longitud del intervalo de guarda ( $\mu s$ )	Variable de acuerdo a las condiciones de la línea: 5.56, 7.56, 47.12
Modulación primaria (por subportadora)	BPSK, QPSK, 8-, 16-, 64-, 256-, 1024- y 4096-QAM
Banda de frecuencia (MHz)	2-30 (bandas opcionales: 2-48 y 2-60)
Corrección de errores	Turbo código convolucional
Velocidad máxima de transmisión (Mb/s)	545 (8/9 CTC)
<b>Capa PHY OFDM-Ondeleta</b>	
Método de comunicación	OFDM basado en ondeletas
Puntos de la transformada ondeleta discreta	512, 1024
Frecuencia de muestreo (MHz)	62.5, 125
Longitud de símbolo ( $\mu s$ )	8.192
Longitud del intervalo de guarda ( $\mu s$ )	No es necesario
Modulación primaria (por subportadora)	BPSK, 4-, 8-, 16-, 32-PAM
Banda de frecuencia (MHz)	2-28 (banda opcional: 2-60)
Corrección de errores	RS, RS-CC; LDPC (opcional)
Velocidad máxima de transmisión (Mb/s) (banda 2-60 y FEC)	544 (239/255 RS)

- IEEE P1675. Proporciona estándares de prueba y verificación para el hardware comúnmente utilizado en la instalación de banda ancha por la red eléctrica. También provee métodos de instalación estándar para asegurar la compatibilidad con los códigos y estándares aplicables.
- IEEE P1775. Se enfoca en los requerimientos para una compatibilidad electromagnética (EMC), y proporciona métodos de prueba y medición de equipo BPL, así como de su instalación.

La HomePlug Powerline Alliance fue fundada en el año 2000 y engloba a más de 65 compañías. Algunos de los estándares generados por esta alianza son HomePlug 1.0, HomePlug Turbo y HomePlug AV. Estos estándares son para redes domésticas sobre líneas eléctricas. La especificación HomePlug 1.0 permite velocidades de 14Mbps, la HomePlug Turbo de 85Mbps y la HomePlug AV de hasta 200Mbps.

En el diseño de la plataforma que aquí se presenta, no se considera utilizar algún estándar como referencia para su construcción, ya que como se mencionó anteriormente, la principal aplicación pensada es con fines de brindar apoyo en investigaciones básicas en el ámbito de la tecnología PLC. Sin embargo, se procurará que la implementación exhiba la flexibilidad necesaria para que en un futuro pueda ser utilizada en el desarrollo de un módem PLC que se apegue al estándar IEEE1901.

## 2.5. El canal: Problemas y soluciones

La línea eléctrica no presenta características y parámetros constantes, ya que depende del tipo y la cantidad de cargas conectadas a ella. Por lo tanto, las características, ruido, impedancia y atenuación, no varían únicamente con la localización, sino que también lo hacen con respecto al tiempo. Debido a lo anterior, el sistema PLC debe ser capaz de adaptarse a las características inconstantes de la línea eléctrica, y alcanzar la más alta tasa de transmisión posible en tales condiciones [41, 44].

### 2.5.1. Impedancia

La impedancia del sistema de distribución de energía eléctrica es bastante difícil de caracterizar, ya que es altamente variable con la frecuencia y va desde unos pocos ohms a algunos kilo ohms, con picos en algunas frecuencias en donde la red se comporta como un circuito paralelo resonante. En función de una carga determinada y la topología de red, la impedancia puede ser generalmente caracterizada en alguna parte entre los 90 y 100 ohms[30, 44]. Como la impedancia neta es altamente influenciada por la topología de red y la cantidad de cargas conectadas, se puede decir que las líneas eléctricas de bajo voltaje, no presentan esencialmente impedancia característica, ya que las cargas que son conectadas y desconectadas aleatoriamente, introducen un cambio en la impedancia de la línea.

### 2.5.2. Atenuación

Los efectos de atenuación que sufre una señal depende de la distancia que ésta tiene que viajar por la red de energía eléctrica. También depende de las características variadas de las líneas de medio y bajo voltaje, y de los cables que se utilizan para la distribución de la energía eléctrica en los hogares. La atenuación de las señales ocurre más fácilmente en banda ancha (MHz) que en banda angosta (KHz)[26].

### 2.5.3. Ruido

El ruido presente en un canal PLC es una de las cuestiones importantes a considerar para cualquiera que pretenda la transmisión de datos por este medio. Existen diferentes fuentes de ruido, a saber, motores, lámparas fluorescentes y de halógeno, fuentes de alimentación conmutadas, reguladores de intensidad, señales provenientes de transmisoras de radio, que dan como resultado una curva de ruido mucho muy dependiente de la localización y del tiempo. Generalmente el ruido del canal varía fuertemente con la frecuencia, cargas conectadas, hora del día y localización geográfica. Los estudios realizados en lo que corresponde al ruido en el canal PLC indican que pueden ser categorizados en cuatro tipos[30, 38]:

- Ruido coloreado de fondo. Este tipo de ruido se considera como la suma de varias fuentes de ruido de baja potencia (por ejemplo motores universales) y puede ser variante con el tiempo. Su densidad de potencia espectral es dependiente de la frecuencia y decrece con el incremento de la frecuencia.
- Ruido impulsivo periódico. Este tipo de ruido es síncrono y asíncrono a la frecuencia de la red, principalmente derivados de aparatos que generan armónicos de 50 a 100 Hz. Usualmente este tipo de ruido impulsivo se repite en múltiplos de la frecuencia con que se

distribuye la energía eléctrica (60/50Hz). Tiene una duración corta, cerca de unos pocos microsegundos y su densidad espectral decreciente con el incremento de la frecuencia.

- Ruido de banda angosta. Consiste de señales sinusoidales con amplitud modulada (estaciones de radio, señales provenientes del circuito horizontal de televisión, etc.). Los niveles de interferencia varían a diferentes horas del día.
- Ruido asíncrono impulsivo. Este tipo de ruido es el más perjudicial para la transmisión de datos por la red eléctrica, y se debe principalmente a la conmutación de cargas. Su duración varía de unos pocos milisegundos a milisegundos con tiempos de arrivo aleatorios.

Para hacer frente a los problemas que presenta el canal PLC se han desarrollado técnicas avanzadas de procesamiento digital de señales y de comunicaciones digitales. Específicamente, los experimentos realizados con FSK y OFDM se han mostrado prometedores.

FSK es una técnica de modulación que utiliza dos frecuencias diferentes para enviar una sola señal digital. Básicamente, un '1' lógico es enviado como una onda senoidal a una frecuencia, mientras que un '0' lógico es enviado como una onda senoidal a una frecuencia diferente. Esto permite que la señal modulada sea menos susceptible al ruido. FSK es utilizada en aplicaciones de baja velocidad como medición y control de carga.

OFDM es una técnica de modulación multiportadora ampliamente adoptada para este tipo de canales, debido a su alta eficiencia espectral, inmunidad inherente al ruido impulsivo así como a la interferencia de radio frecuencia (RFI). Además, la interferencia entre símbolos (ISI) y la interferencia entre portadoras (ICI) puede ser fácilmente eliminada insertando un prefijo cíclico (CP) al inicio de cada bloque OFDM. Por otro lado, OFDM puede ser implementado utilizando una técnica de detección coherente o no coherente. Una detección coherente generalmente proporciona una ganancia de SNR frente a la detección no coherente, debido a que la técnica de detección coherente utiliza la información del estado del canal. Esto implica, sin embargo, un receptor más complejo ya que la información del estado del canal se obtiene por medio de técnicas de estimación de canal.

## 2.6. Trabajos relacionados

La tabla 2.6 resume algunas de las características de los trabajos encontrados, relacionados al diseño e implementación de modems PLC basados en OFDM.

En [3] los autores presentan la implementación de un módem PLC basado en DMT. Los algoritmos son programados en un DSP Motorola de la serie 56303, utilizando la FFT como núcleo principal para la modulación DMT. El diseño contempla el uso de codificación de Trellis y el algoritmo de Viterbi para la decodificación. Por otro lado, el proceso de sincronización se realiza explotando las características del prefijo cíclico. Las pruebas realizadas al diseño muestran el desempeño frente a ruido gaussiano e impulsivo. En [15] se presenta el desarrollo de un módem PLC implementado en un DSP TMS320VC5402 de Texas Instruments. El esquema de modulación es OFDM empleando la FFT. En este trabajo se analiza el desempeño de una modulación adaptativa contra una modulación no adaptativa. No se hace mención de las etapas de sincronización ni de estimación de canal. En [4] se propone el diseño de un transceiver OFDM para BPL implementado en un FPGA. En este trabajo, se explora los efectos de una implementación en punto fijo de la modulación OFDM empleando la FFT;

aunque se describe brevemente el uso de ondeletas para la generación de OFDM. Además, se realizan pruebas de algunos tipos de ruido que se pueden encontrar en el medio eléctrico. Por otra parte, el diseño no incluye etapas importantes como la sincronización y estimación de canal. En [35] se describe el desarrollo e implementación de un módem PLC en un FPGA Spartan 3 de Xilinx, siguiendo con el estándar HyperLAN 2. Se utiliza Simulink para el proceso de simulación del diseño, aplicando la FFT para el proceso de modulación OFDM. Los algoritmos programados en el FPGA incluyen las etapas de sincronización y estimación de canal. Las pruebas realizadas se enfocan al proceso de sincronización. En [11], al igual que el trabajo anterior, se describe el diseño de un módem PLC implementado en un FPGA. El esquema de modulación es OFDM empleando la FFT. En este trabajo se realizaron pruebas sobre ruido comúnmente encontrado en un canal PLC, sin embargo, no se mencionan etapas de sincronización, ni de estimación de canal. En [43] se detalla el diseño de un módem PLC implementado en el DSP ADSP BF533 de Analog Devices. El módem utiliza la FFT para el proceso de modulación y demodulación OFDM. El desarrollo contempla un codificador convolucional y decodificador Viterbi para la etapa de detección y corrección de errores, no obstante, no se incluyen los módulos de sincronización ni de estimación de canal. Las pruebas se enfocan a la evaluación de los recursos de memoria y ciclos de CPU utilizados.

Tabla 2.6: Trabajos relacionados.

–	Basak et al. [3]	Jie et al. [15]	Carducci et al. [4]	Rastislav [35]	Garcia et al. [11]	Souha et al. [43]	Propuesta
Año	2001	2002	2008	2009	2009	2010	2011
Mapeo	DPSK	PSK, QAM	DPSK, QAM	QAM	QAM	DSPK	QAM
Modulación	DMT FFT	OFDM FFT	OFDM FFT	OFDM FFT	OFDM FFT	OFDM FFT	OFDM FFT
Sincronización	Si	No	No	Si	No	No	Si
Estimación de canal	No	No	No	Si	No	No	Si
FEC	TCM Viterbi	No	No	No	No	Convol. Viterbi	No

## 2.7. Resumen del capítulo

En este capítulo se ha presentado la evolución de las comunicaciones por la red eléctrica, donde sus inicios se remontan a las aplicaciones en telemetría y control. Más tarde, gracias al avance en las técnicas de procesamiento digital de señales, y sobre todo, en las técnicas de modulación digital, se logra un incremento en las tasas de transmisión de los sistemas PLC, dando lugar a su clasificación y aplicación en diferentes áreas. De esta manera es como surge la tecnología PLC; y con ello, compañías y organizaciones que se involucran en su desarrollo y estandarización. Por otra parte, son muchos los problemas a resolver en un canal que no fue pensado para propósitos de transmisión de información, debido a esto los trabajos encontrados sobre desarrollos de módems PLC, distinguen a OFDM como la técnica de modulación adecuada para resolver el problema de transmisión de información por la red eléctrica.

## Capítulo 3

# Marco Teórico

En este capítulo se presenta una breve introducción de las técnicas de comunicaciones digitales y de procesamiento digital de señales utilizadas en el desarrollo del módem PLC. En primera instancia, se hace la introducción a los sistemas de comunicación digitales y las diferentes técnicas de modulación existentes, para posteriormente hacer la revisión de OFDM y su implementación de manera completamente digital utilizando la DFT. En la última parte del capítulo se expone la FFT y la deducción de la técnica de *diezmado en frecuencia* (DIF).

### 3.1. Sistema digital de comunicaciones

Los sistemas de comunicaciones digitales, son cada vez más atractivos debido a la creciente demanda de comunicación de datos; además, las comunicaciones digitales ofrecen opciones de procesamiento de datos y flexibilidad no disponibles con transmisiones analógicas. La principal característica de un sistema digital de comunicaciones (SDC) es que durante un intervalo finito de tiempo, éste envía una forma de onda desde un conjunto finito de posibles formas de onda, en contraste con un sistema de comunicación analógica, el cual, envía una forma de onda desde una infinita variedad de formas de onda con teóricamente infinita resolución. En un SDC, el objetivo en el receptor no es reproducir la forma de onda transmitida con precisión; en vez de ello, el objetivo es determinar desde una señal perturbada con ruido, cual forma de onda desde un conjunto finito de formas de onda fue enviada por el transmisor.

En la actualidad, los sistemas de comunicaciones digitales, ya sean comerciales o militares, le han ganado terreno a los sistemas de comunicaciones analógicas, y hay muchas razones para este cambio. Entre las razones principales se pueden mencionar que las señales digitales son fáciles de generar, comparadas con las señales analógicas; los circuitos digitales son más *confiables*, pueden ser producidos a un bajo costo comparados a los circuitos analógicos y el hardware digital se presta para una implementación más *flexible* que el hardware analógico.

En la figura 3.1 se pueden observar los bloques básicos de un sistema digital de comunicaciones, en el cual, los bloques de *modulación y demodulación/detección* son llamados un módem. El término módem a menudo abarca varios pasos de procesamiento de señal; cuando este es el caso, el módem puede ser considerado como el *cerebro* del sistema y el conjunto transmisor-receptor pueden ser considerados como los *músculos* del sistema.

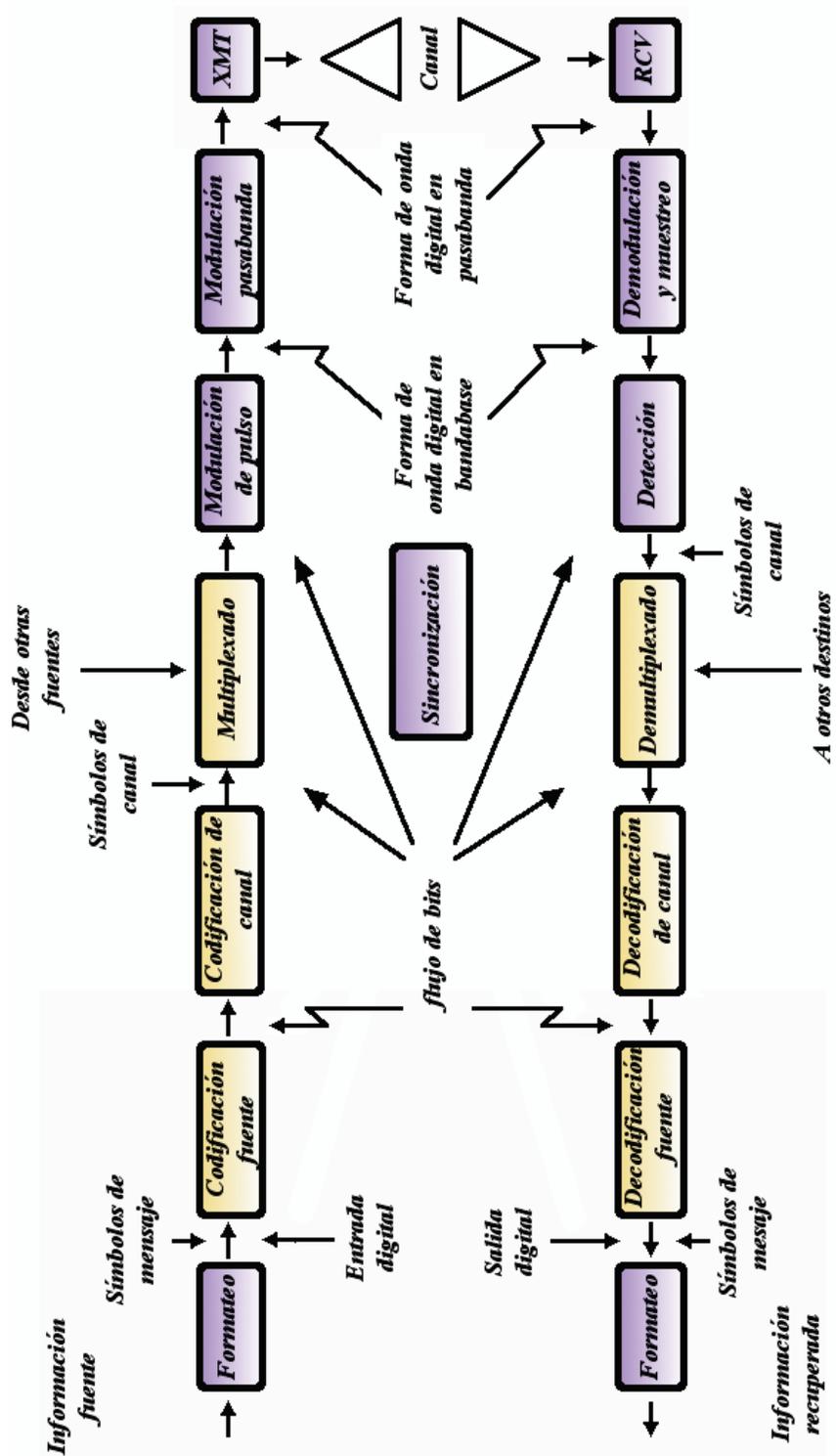


Figura 3.1: Diagrama de bloques básico de un SDC.

## 3.2. Técnicas de modulación digital

Las técnicas de modulación digital pueden agruparse en tres grandes grupos, dependiendo de la característica que se varíe en la señal portadora. Cuando se varía la amplitud, la técnica de modulación digital que se utiliza se conoce como *modulación por desplazamiento de amplitud* (ASK, por sus siglas en inglés). Si se varía la frecuencia o la fase, las técnicas empleadas se denominan *modulación por desplazamiento de frecuencia* (FSK) y *modulación por desplazamiento de fase* (PSK), respectivamente. Cualquiera que sea la técnica de modulación empleada, la amplitud, la frecuencia o la fase de la señal portadora podrá tomar únicamente un número finito de valores discretos.

En comunicaciones digitales, resulta muy práctico representar una fuente discreta de señales a partir de su *espacio de señal* o *constelación*. Una *constelación* es una representación geométrica de señales en un espacio de  $n$  dimensiones, en donde se visualizan todos los símbolos de salida posibles que puede generar un modulador. Gracias a que en una *constelación* cada símbolo tiene asociado un valor de *magnitud* y uno de *fase*, salvo en el caso de la modulación FSK, todos los demás esquemas de modulación digital pueden representarse en un plano de dos dimensiones.

La amplitud y la fase de una señal pueden modularse simultáneamente o por separado, aunque esto resulta difícil de generar y principalmente de detectar. En vez de ello, es muy práctico separar la señal en dos componentes independientes conocidas como componente *en fase* (**I**) y componente *en cuadratura* (**Q**), ambos ortogonales entre sí. Cuando nos referimos a comunicaciones digitales, es común expresar la modulación en términos de estas dos componentes, razón por la cual, la representación de una *constelación bidimensional I-Q* es particularmente útil y puede asociarse a la mayor parte de los métodos de modulación digital.

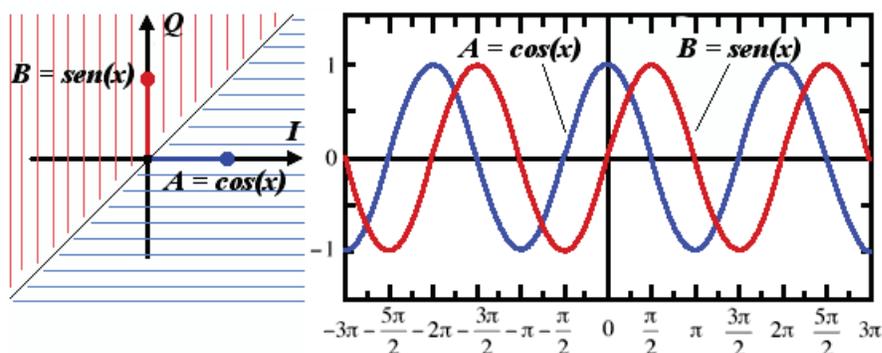


Figura 3.2: Proyección de la constelación I-Q.

Como se puede observar en la figura 3.2, en una *constelación I-Q*, la *componente en fase* se proyecta en el eje  $x$  y la *componente en cuadratura* se proyecta en el eje  $y$  de un plano cartesiano. Se dice que una señal estará en fase cuando su ángulo de fase sea cero grados (situada en el eje I) y que una señal estará en cuadratura cuando se encuentre defasada noventa grados respecto a la señal en fase (situada en el eje Q).

### 3.3. QAM

Entre los formatos de modulación destaca el método conocido como QAM. La *modulación en amplitud en cuadratura* (QAM, por sus siglas en inglés) es un esquema de modulación multinivel en donde se envía una de  $M = 2^L$  señales, con distintas combinaciones de amplitud y fase. Utilizando múltiples niveles, tanto en la modulación en amplitud como en la modulación en fase, es posible la transmisión de grupos de bits, de manera que cada uno de estos grupos serán representativos de un conjunto nivel-fase característico de la portadora de señal, mismo que dará cabida a un símbolo.

Una de las características principales de la modulación QAM es que modula la mitad de los símbolos con una frecuencia y la otra mitad con la misma frecuencia, pero defasada noventa grados. El resultado de las componentes después se suma, dando lugar a la señal QAM. De esta forma, QAM permite llevar dos canales en una misma frecuencia mediante la transmisión ortogonal de uno de ellos con respecto al otro. En la figura 3.3 se puede observar la constelación para una modulación 16-QAM.

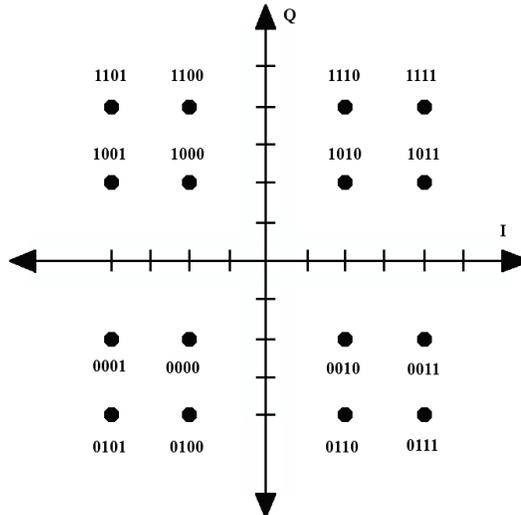


Figura 3.3: Constelación 16-QAM.

16-QAM permite contar con 16 estados diferentes, mismos que estarán determinados por el número de símbolos mapeados en su constelación correspondiente. Debido a que  $16 = 2^4$ , cada uno de estos símbolos puede representarse mediante cuatro bits, dos de ellos correspondientes a la componente en cuadratura, y los dos restantes, correspondientes a la componente en fase de la señal. Puesto que existen estas dos componentes, cada una representada con dos bits en 16-QAM, es posible transmitir cuatro posibles niveles de amplitud para cada componente, lo que supone que, por el efecto de la cuadratura, pueden transmitirse 16 estados.

Para comprender mejor lo anterior, en la figura 3.4 se presenta el diagrama de bloques que define a un modulador 16-QAM. Nótese que, para el caso de la modulación 16-QAM, cada símbolo estará formado por cuatro bits: un bit de polaridad y otro más para el nivel de la componente I. Añadiendo más bits a las componentes I y Q del modulador se obtendrán formatos QAM más eficientes como 64-QAM y 256-QAM. La serie de bits que ingresa al modulador deberá pasar primero por un divisor de bits. El divisor de bits, que en este caso

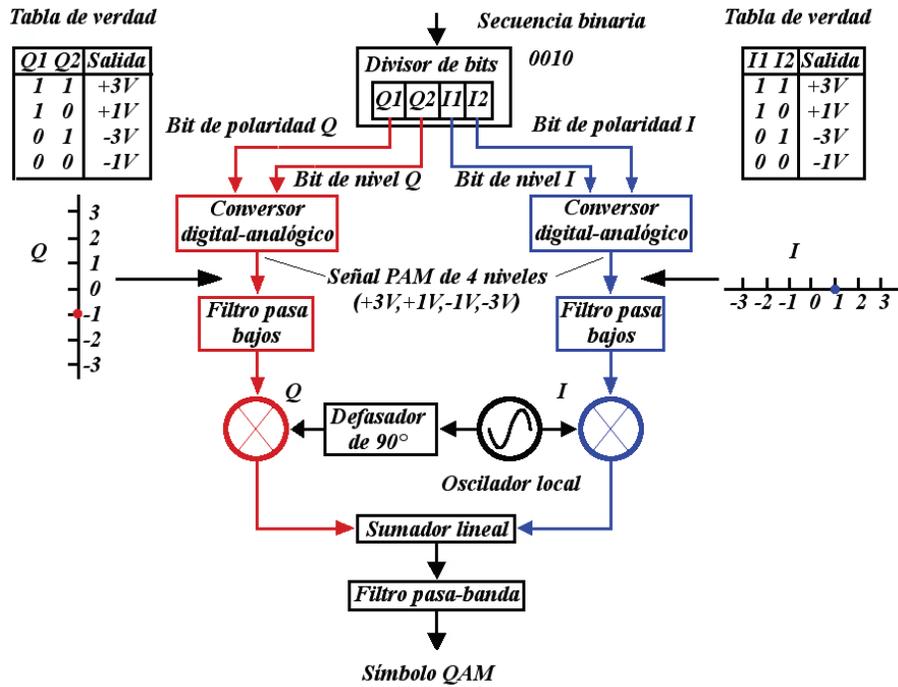


Figura 3.4: Ejemplo de un modulador 16-QAM.

es de cuatro bits por tratarse de una modulación 16-QAM, toma los primeros bits de la serie y los enruta, de manera simultánea y en paralelo, hacia el convertor digital-analógico correspondiente. Supóngase que al divisor ingresan los cuatro primeros bits de la secuencia binaria  $0010110_b$ , por lo que los dos primeros bits  $00_b$  son enrutados como bits en cuadratura y los dos siguientes  $10_b$  como bits en fase. En cada caso, el primero de este par de bits corresponderá al bit de polaridad, y el segundo, al de nivel.

Dirijamos primero nuestra atención hacia los bloques de modulación en cuadratura del diagrama. El primero de los bits del par que ingresa es un cero, de acuerdo a la secuencia dada como ejemplo. Un cero en el primer bit indicará polaridad negativa, mientras que un uno representará polaridad positiva. En el caso del bit de nivel, la amplitud estará determinada por el convertor digital-analógico. Para determinar el valor que representa un par de bits, es necesario observar la tabla de verdad que define las reglas lógicas que corresponden a la operación del convertor. Continuando con el ejemplo, un cero indicará un nivel de 1, mientras que un uno representará un nivel de 3. De esta forma, el primer par de bits que ingresa por el bloque de cuadratura representará un nivel de  $-1V$ , uno de los cuatro posibles niveles que puede tomar la señal a la salida del convertor digital-analógico. La señal que se obtiene es filtrada a través de un dispositivo pasa-bajas y posteriormente enviada hacia un mezclador, cuya función es modular la señal en amplitud sobre una portadora defasada noventa grados respecto a la portadora utilizada en el modulador de la componente en fase.

Ahora enfoquemos la atención en el modulador en fase del diagrama. Una vez que el divisor enruta el segundo par de bits  $10_b$  de la secuencia al convertor digital-analógico, si la tabla de verdad que define la operación de este último es igual a la del convertor del bloque en cuadratura, el par de bits representaría una señal con nivel  $+1V$  a la salida del convertor.

De manera similar, esta señal es filtrada y modulada en amplitud con una portadora que conserva su fase original. Las señales que representan las componentes en fase y en cuadratura son entonces llevadas a un sumador lineal, donde los cuatro bits  $0010_b$  se combinan para dar lugar al símbolo correspondiente. Sin embargo, ¿a qué símbolo de la constelación 16-QAM corresponden los primeros cuatro bits de la secuencia tomada como ejemplo?; los niveles a la salida del conversor digital-analógico responden a esta incógnita. Partiendo del origen de la constelación, ubicamos una marca en la dirección negativa del eje Q y una marca en la dirección positiva del eje I. Estas marcas corresponden a los niveles de salida del conversor en cuadratura ( $-1V$ ) y del conversor en fase ( $+1V$ ), respectivamente.

### 3.4. OFDM

En los últimos años, OFDM se ha empleado satisfactoriamente a una gran variedad de aplicaciones en comunicaciones digitales, por lo que es adoptado como un estándar de la capa física de sistemas importantes, tanto alámbricas como inalámbricas; aunque la teoría, los algoritmos y técnicas de implementación, aún permanecen como temas de interés actuales [48]. En general, OFDM es una técnica de transmisión multiportadora, en la cual el espectro de frecuencia disponible es dividido en subportadoras, cada una modulada por un flujo de datos de baja velocidad. OFDM utiliza el principio de FDM que permite subdividir el ancho de banda disponible en múltiples subcanales, donde cada subcanal transporta información. Sin embargo, OFDM utiliza el espectro de frecuencias más eficientemente, debido a que los subcanales son espaciados más cerca uno del otro. Esto se logra haciendo que las portadoras sean ortogonales unas de otras, previniendo la interferencia entre ellas.

En un sistema OFDM, la información es repartida entre las subportadoras, donde cada subportadora corresponde a una senoide y a la suma de estas senoides es lo que se denomina una señal OFDM. Hay dos condiciones a considerar para lograr la ortogonalidad entre subportadoras. Primero, todas las subportadoras deben tener un número entero de ciclos por símbolo OFDM. Segundo, las subportadoras adyacentes deben tener exactamente un ciclo de diferencia por símbolo OFDM. En la figura 3.5 se puede observar la representación en el dominio del tiempo y de la frecuencia de una señal OFDM.

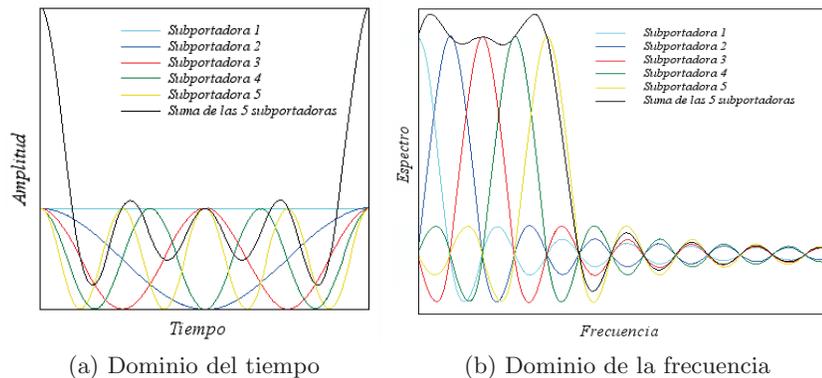


Figura 3.5: Señal OFDM en el dominio del tiempo y la frecuencia.

### 3.4.1. Modulación y demodulación

Como ya se ha mencionado, OFDM es un caso especial de modulación multiportadora, donde la idea básica consiste en separar un flujo de datos de alta velocidad en flujos paralelos de velocidad más baja. En OFDM, se divide el espectro de frecuencias disponibles en  $N$  subportadoras y cada subportadora es modulada por un flujo de datos de baja velocidad. Usualmente, el flujo de datos que modula a cada subportadora consta de valores apropiados en *fase* (I) y *cuadratura* (Q) obtenidos de una constelación como QAM o QPSK. A estos valores que modulan a cada subportadora se les llama *símbolos fuente* y corresponden a valores complejos de la forma  $s = a_I + jb_Q$ . Por lo anterior, OFDM transmite en paralelo  $N$  símbolos fuente  $s_k$ ,  $k = 0, 1, \dots, N - 1$  sobre  $N$  subportadoras, y a la señal resultante se le denomina *símbolo OFDM*.

Si el periodo de un símbolo fuente es  $T_d$ , entonces la *duración del símbolo OFDM* está determinado por

$$T_s = NT_d \quad (3.1)$$

y el espaciamiento mínimo entre dos portadoras adyacentes que permite conservar la ortogonalidad entre éstas, también conocido como *condición de ortogonalidad*, es

$$\Delta f = \frac{1}{T_s} \quad (3.2)$$

De acuerdo a la teoría de las comunicaciones digitales, el  $k$ -ésimo símbolo modulado en QAM o QPSK puede ser representado como

$$u(t) = s_k w(t) e^{j(2\pi f_c t + \theta_c)} \quad (3.3)$$

por lo que si  $s_k$  representa a los símbolos complejos a ser transmitidos por la modulación OFDM y  $w(t)$  corresponde a una ventana rectangular para el formateo de pulso, la señal modulada en OFDM puede ser expresada como

$$s(t) = \sum_{k=0}^{N-1} s_k e^{j(2\pi f_k t)} = \sum_{k=0}^{N-1} s_k \varphi_k(t), \text{ para } 0 \leq t \leq T_s, \quad (3.4)$$

donde  $f_k = f_0 + k\Delta f$ , y

$$\varphi_k(t) = \begin{cases} e^{j(2\pi f_k t)}, & \text{si } 0 \leq t \leq T_s, \\ 0, & \text{en cualquier otro caso} \end{cases} \quad (3.5)$$

para  $k = 0, 1, \dots, N - 1$ .

En la figura 3.6 se puede observar la representación general de un modulador OFDM.

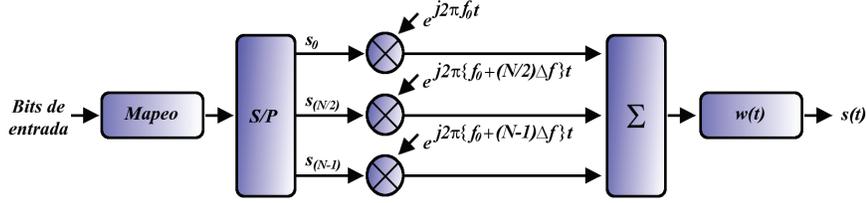


Figura 3.6: Esquema general de un modulador OFDM.

Por otra parte, para que del lado receptor se pueda demodular la señal OFDM, la duración del símbolo debe ser lo suficientemente largo tal que se cumpla la condición de ortogonalidad dada en 3.2. Por consiguiente,  $\varphi_k(t)$  debe ser una base de señales que cumplan con la propiedad de ser ortonormales

$$\int_{-\infty}^{\infty} \varphi_k(t) \varphi_l^*(t) dt = \begin{cases} 0, & \text{si } k \neq l, \\ 1, & \text{si } k = l \end{cases} \quad (3.6)$$

Debido a la condición de ortogonalidad, tenemos

$$\begin{aligned} & \frac{1}{T_s} \int_0^{T_s} \varphi_k(t) \varphi_l^*(t) dt \\ &= \frac{1}{T_s} \int_0^{T_s} e^{j2\pi(f_k - f_l)t} dt \\ &= \frac{1}{T_s} \int_0^{T_s} e^{j2\pi(k-l)\Delta f t} dt \\ &= \delta[k - l] \end{aligned} \quad (3.7)$$

donde  $\delta[k - l]$  es la función delta definida como

$$\delta[n] = \begin{cases} 1, & \text{if } n = 0, \\ 0, & \text{en cualquier otro caso} \end{cases} \quad (3.8)$$

La ecuación 3.7 muestra que  $\{\varphi(t)\}_{k=0}^{N-1}$  es un conjunto ortogonal de funciones. Utilizando esta propiedad, la señal OFDM puede ser demodulada por

$$\begin{aligned} & \frac{1}{T_s} \int_0^{T_s} s(t) e^{-j2\pi f_k t} dt \\ &= \frac{1}{T_s} \int_0^{T_s} \left( \sum_{l=0}^{N-1} s_l \varphi_l(t) \right) \varphi_k^*(t) dt \\ &= \sum_{l=0}^{N-1} s_l \left( \frac{1}{T_s} \int_0^{T_s} \varphi_l(t) \varphi_k^*(t) dt \right) \\ &= \sum_{l=0}^{N-1} s_l \delta[l - k] \\ &= s_k. \end{aligned} \quad (3.9)$$

La figura 3.7 presenta el esquema general de un modulador OFDM.

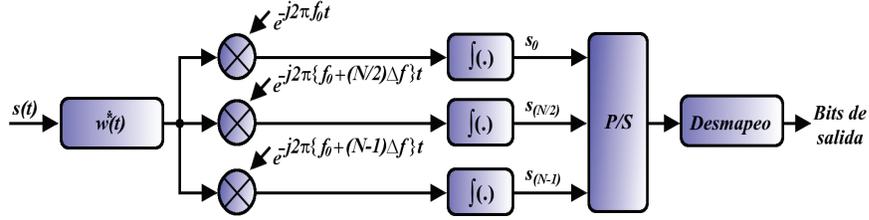


Figura 3.7: Esquema general de un demodulador OFDM.

El principio de ortogonalidad entre señales permite que multiples señales de información puedan ser transmitidas y detectadas sobre un canal de comunicación común sin interferencia.

### 3.4.2. Aplicación de la DFT

Una de las mayores contribuciones para resolver el problema de la complejidad de implementar un sistema OFDM, fue realizada por Weistein y Ebert en 1971. Su propuesta contempla la aplicación de la *transformada discreta de Fourier* (DFT) para disminuir las dificultades de la modulación y demodulación OFDM [24, 36].

De acuerdo a lo visto en la sección anterior, una señal OFDM puede ser expresada como

$$s(t) = \sum_{k=0}^{N-1} s_k e^{j2\pi f_k t}. \quad (3.10)$$

Ahora, si la señal  $s(t)$  es muestreada a la tasa de  $T_{sample} = \frac{T_s}{N}$ , entonces la ecuación anterior puede ser reescrita como

$$S_n = s\left(n \frac{T_s}{N}\right) = \sum_{k=0}^{N-1} s_k e^{j2\pi f_k \frac{nT_s}{N}}. \quad (3.11)$$

Sin perder generalidad, haciendo  $f_0 = 0$ , entonces  $f_k T_s = k$  y la ecuación 3.11 se convierte en

$$\begin{aligned} S_n &= \sum_{k=0}^{N-1} s_k e^{j \frac{2\pi k n}{N}} \\ &= IDFT\{s_k\} \end{aligned} \quad (3.12)$$

donde la IDFT denota la *transformada discreta inversa de Fourier*. Por lo tanto, dado que el transmisor OFDM puede ser implementado utilizando la IDFT, de la misma manera el

receptor puede ser implementado utilizando la DFT, como se muestra a continuación

$$\begin{aligned}
& DFT\{S_n\} & (3.13) \\
& = \sum_{n=0}^{N-1} S_n e^{-j\frac{2\pi kn}{N}} \\
& = \sum_{n=0}^{N-1} \left( \sum_{l=0}^{N-1} s_l e^{j\frac{2\pi ln}{N}} \right) e^{-j\frac{2\pi kn}{N}} \\
& = \sum_{n=0}^{N-1} \sum_{l=0}^{N-1} s_l e^{j\frac{2\pi n(l-k)}{N}} \\
& = \sum_{l=0}^{N-1} s_l \sum_{n=0}^{N-1} e^{j\frac{2\pi n(l-k)}{N}} \\
& = \sum_{l=0}^{N-1} s_l \delta[l - k] \\
& = s_k.
\end{aligned}$$

### 3.4.3. Intervalo de guarda

Uno de los problemas que se presentan al momento de diseñar un sistema para la transmisión de información por la red eléctrica, consiste en el fenómeno de propagación multiruta [51]. El fenómeno surge a causa de que la señal emitida por el transmisor viaja por diferentes rutas antes de llegar al receptor, debido a la propia topología de la red. Esto origina que diferentes versiones de la señal original lleguen con retardo al receptor. A este retardo se le denomina *retardo de dispersión del canal*, y es el responsable de ocasionar problemas de *interferencia entre símbolos* (ISI).

En un sistema OFDM, al transmitir un símbolo inmediatamente seguido de otro, la ortogonalidad entre las portadoras no será mantenida, debido a que el final de un símbolo interferirá en  $k$  muestras con el inicio del otro, a causa de la propia respuesta del canal. Cuando esto sucede, un subcanal comienza a interferir con otro, generando *interferencia entre canales* (ICI). La figura 3.8 muestra los efectos de ISI e ICI debidos a la respuesta del canal.

Una de las razones importantes de utilizar OFDM es por la manera eficiente de lidiar con el retardo de dispersión. Dividiendo suficientemente el número de subportadoras, la duración de los símbolos puede hacerse mayor al retardo de dispersión del canal, lo que permite minimizar los efectos de ISI. De esta manera, las distorsiones introducidas en la señal por el canal son limitadas en el dominio de la frecuencia, ya que cada subcanal se hace suficientemente estrecho para ser afectado únicamente por una respuesta local del canal [21].

Para mitigar los efectos de ISI causados por el retardo de dispersión del canal, se introduce un intervalo en el dominio del tiempo denominado *intervalo de guarda*. Donde la solución más simple es hacer al intervalo de guarda igual al retardo de dispersión del canal, durante el cual, no se transmite energía. Sin embargo, el canal es utilizado más eficientemente si algo se transmite durante dicho intervalo que contribuya a la energía de la señal sin la introducción de ISI. Ese algo es el llamado *prefijo cíclico* (CP). Y Consiste en la adición de los últimos  $N_g$  DFT coeficientes a cada bloque OFDM (ver figura 3.9), de tal manera que la longitud del

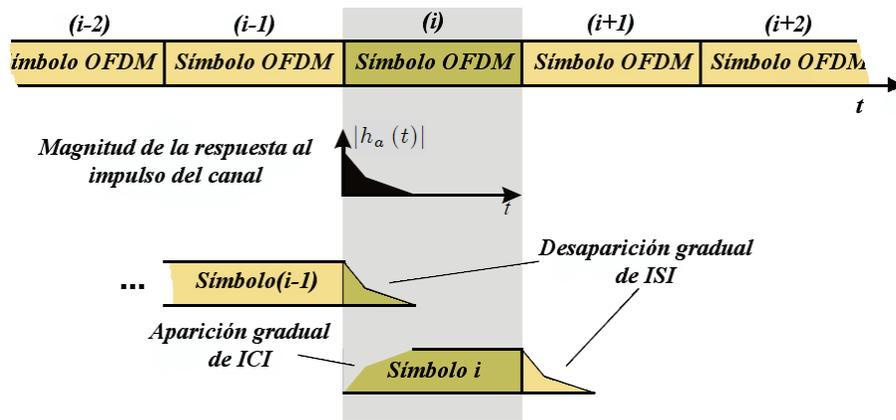


Figura 3.8: Interferencia ISI e ICI debidos a la respuesta del canal.

CP es por lo menos igual a la longitud del canal  $N_h$  en muestras. El intervalo de guarda de longitud  $N_g$  es una sobrecarga que resulta en una penalización de potencia y ancho de banda a causa de que consiste de símbolos de modulación redundantes. Sin embargo, el intervalo de guarda es útil para funciones de sincronización en el receptor, ya que éste, contiene símbolos de modulación repetidos a una separación conocida de muestras. La duración en tiempo de un símbolo OFDM es  $N + N_g$  veces más largo que un símbolo modulado en un sistema de portadora única.

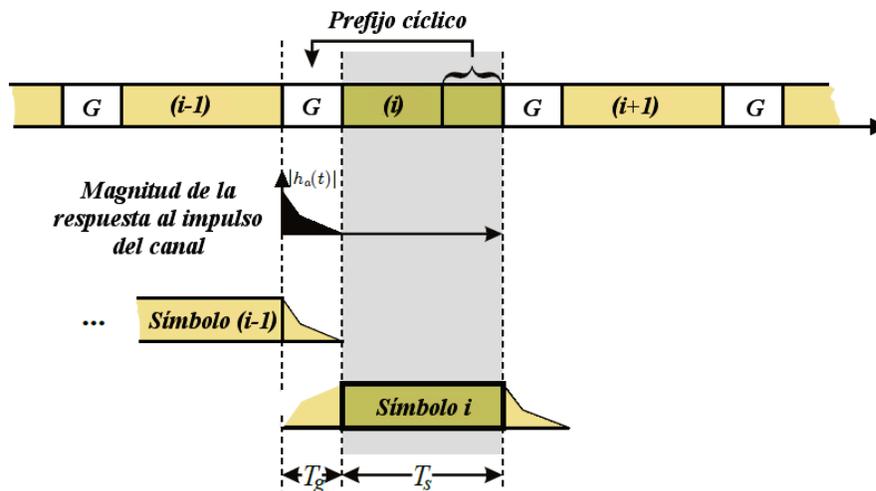


Figura 3.9: Intervalo de guarda/Prefijo cíclico.

### 3.4.4. Sincronización

La sincronización es un factor crítico en términos de prestaciones cuando se diseña un sistema de comunicaciones basado en OFDM, a causa de que OFDM es altamente sensible a errores de sincronización comparado con sistemas de portadora única [13]. Existen dos problemas fundamentales en el diseño de receptores OFDM. El primer problema que se

presenta es la sincronización en tiempo del sistema, y consiste en determinar cuando inicia un símbolo OFDM. El segundo problema, es el desajuste que existe entre el oscilador del transmisor con respecto al del receptor, provocando un desplazamiento en frecuencia de las subportadoras. Por consiguiente, entre los objetivos principales de la sincronización se incluyen, identificar el comienzo en tiempo de un símbolo OFDM y asegurar la ortogonalidad entre las subportadoras.

A continuación, se resumen en cuatro las principales tareas de sincronización a realizarse en un sistema basado en OFDM [1]:

- Sincronización de símbolo.
- Sincronización de trama.
- Estimación y corrección de frecuencia.
- Estimación de canal.

Las técnicas de sincronización y corrección de frecuencia se pueden clasificar en dos categorías: técnicas con agregado de datos y técnicas sin agregado de datos. Las técnicas con agregados de datos utilizan *secuencias de entrenamiento* o *símbolos piloto* que son insertados en la señal OFDM. Esta inserción de información adicional disminuye la eficiencia de la transmisión para ofrecer la posibilidad de sincronización, pero aunque estas técnicas no son eficientes, lo compensan con la alta precisión que pueden alcanzar en la estimación, además de una arquitectura simple, la cual es especialmente importante en las transmisiones por paquetes.

Por otro lado, las técnicas sin agregado de datos se basan en la redundancia que existe entre los símbolos OFDM. Esto es, emplean las propiedades de correlación del *prefijo cíclico*; ya que, los  $T_{CP}$  segundos iniciales, son los mismos segundos finales en un símbolo OFDM. En estas técnicas, el prefijo cíclico se utiliza para realizar la autocorrelación con un retraso  $T$ , y cuando a la salida del módulo de correlación se encuentra un *pico*, se evalúa conjuntamente la estimación del desplazamiento en tiempo del símbolo y el desplazamiento en frecuencia, debido a que la *fase* de la estadística en el instante  $t = \tau$  está relacionada con el desplazamiento en frecuencia. El valor de correlación puede ser expresado como

$$x(t) = \int_0^{T_{CP}} r(t - \tau)r^*(t - \tau - T)d\tau \quad (3.14)$$

donde  $r(t)$  es la señal OFDM recibida,  $x(t)$  es la salida del módulo de correlación y  $\tau$  denota el desplazamiento en tiempo.

### 3.5. Transformada Rápida de Fourier

Las señales se pueden clasificar dependiendo de las características de la variable tiempo y los valores que ésta pueda tomar. Las *señales en tiempo continuo* o *señales analógicas* están definidas para todos los valores de tiempo y pueden tomar cualquier valor en el intervalo continuo  $(a, b)$ , donde  $a$  puede ser  $-\infty$  y  $b$  puede ser  $\infty$ , ver figura 3.10. Matemáticamente, estas señales se describen como funciones continuas de la variable continua.

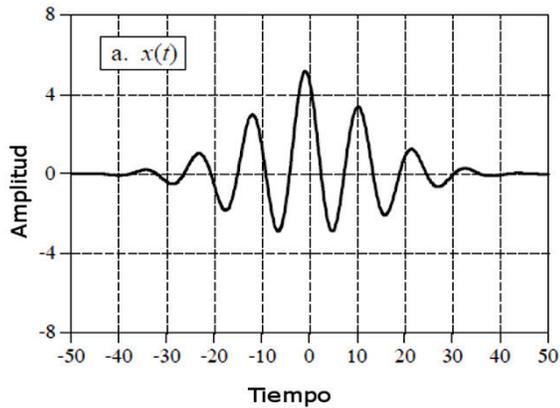


Figura 3.10: Señal continua.

Las *señales en tiempo discreto* están definidas para ciertos valores de tiempo, ver figura 3.11. Estos instantes de tiempo no necesitan estar equidistantes, aunque en la práctica se toman normalmente instantes equiespaciados conforme a intereses computacionales y matemáticos.

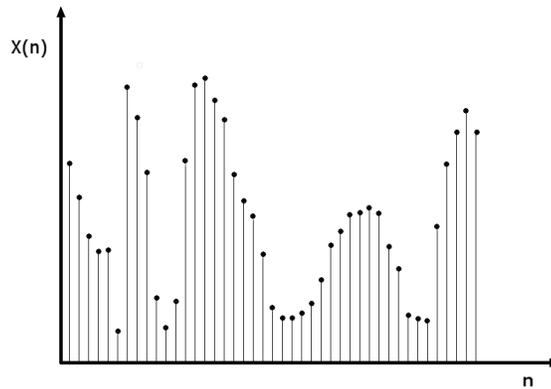


Figura 3.11: Señal discreta.

El análisis frecuencial de señales en tiempo discreto se realiza, normalmente, de forma más conveniente en procesadores de señales digitales, que pueden ser ordenadores digitales de propósito general o hardware digital especialmente diseñado. Para realizar el análisis frecuencial de una señal en tiempo discreto  $\{x(n)\}$ , convertimos la secuencia en el dominio del tiempo en una forma equivalente en el dominio de la frecuencia. Se sabe que tal forma viene dada por la transformada de Fourier,  $X(\omega)$ , de la secuencia  $\{x(n)\}$ . El análisis de Fourier es una familia de técnicas matemáticas, basadas en la descomposición de señales en senoides. Sin embargo,  $X(\omega)$  es una función continua de la frecuencia y, por lo tanto, no es una forma computacionalmente conveniente de la secuencia  $\{X(n)\}$  [33].

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (3.15)$$

La *transformada discreta de Fourier* es un método muy eficiente para determinar el espectro de frecuencia de una señal digitalizada. Permite convertir una secuencia de valores en el dominio del tiempo a una secuencia de valores en el dominio de la frecuencia. La *transformada inversa discreta de Fourier* (IDFT) realiza el proceso contrario. Las fórmulas para calcular la DFT y la IDFT son

$$\mathbf{DFT} \quad X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad k = 0, 1, 2, \dots, N-1 \quad (3.16)$$

$$\mathbf{IDFT} \quad X(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \quad n = 0, 1, 2, \dots, N-1 \quad (3.17)$$

Las expresiones (3.16) y (3.17) pueden ser reescritas como

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad k = 0, 1, 2, \dots, N-1 \quad (3.18)$$

$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-kn} \quad n = 0, 1, 2, \dots, N-1 \quad (3.19)$$

donde por definición,

$$W_N = e^{-j2\pi/N} \quad (3.20)$$

que es la raíz  $n$ -ésima de la unidad.

El inconveniente de realizar algoritmos que implementen tal cual estas fórmulas, es la cantidad de tiempo requerido para computar la salida, ya que si observamos, el cálculo de cada punto de la DFT implica  $N$  multiplicaciones complejas y  $(N-1)$  sumas complejas. Por tanto, los  $N$  puntos de la DFT pueden obtenerse tras  $N^2$  multiplicaciones complejas y  $N(N-1)$  sumas complejas.

Existen varias maneras de calcular la DFT, pero en la práctica, se utilizan un número de algoritmos más sofisticados, conocidos colectivamente como *Transformadas Rápidas de Fourier*, o FFTs. Los cuales, producen los mismos resultados que otros enfoques, pero son increíblemente más eficientes, a menudo, reduciendo el tiempo de cómputo en cientos de veces.

El cálculo directo de la DFT es básicamente ineficiente debido, fundamentalmente, a que no explota las propiedades de simetría del factor de fase  $W_N$ . Todos los algoritmos FFT explotan estas dos propiedades para mejorar su eficiencia. En particular, estas dos propiedades son:

Propiedad de simetría:

$$W_N^{k+N/2} = -W_N^k \quad (3.21)$$

Propiedad de periodicidad:

$$W_N^{k+N} = W_N^k \quad (3.22)$$

El algoritmo que se explora en este trabajo, se basa en la idea de *divide y vencerás*, en la cual una DFT de  $N$  puntos, es descompuesta en dos DFTs de  $\frac{N}{2}$  puntos. Cada DFT de  $\frac{N}{2}$  puntos se descompone a su vez en dos DFTs de  $\frac{N}{4}$  puntos y así sucesivamente. Al final de la descomposición se obtienen  $\frac{N}{2}$  DFTs de dos puntos cada una. La transformada más pequeña viene determinada por la base de la FFT. Para una FFT de base dos,  $N$  debe ser una potencia de dos y la transformada más pequeña es la DFT de dos puntos. Para implementar la FFT existen dos algoritmos: *diezmado en frecuencia* (DIF) y *diezmado en tiempo* (DIT).

En este trabajo se utiliza el diezmado en frecuencia, y la deducción del algoritmo se presenta a continuación.

### 3.5.1. Diezmado en frecuencia

Partiendo de la ecuación (3.18), se inicia dividiendo la fórmula en dos sumatorios, uno de los cuales contiene los primeros  $\frac{N}{2}$  puntos de datos y el otro los últimos  $\frac{N}{2}$  puntos de datos. Así obtenemos

$$\begin{aligned} X(k) &= \sum_{n=0}^{N/2-1} x(n)W_N^{kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{kn} \\ &= \sum_{n=0}^{N/2-1} x(n)W_N^{kn} + \sum_{n=0}^{N/2-1} x\left(n + \frac{N}{2}\right)W_N^{k(n+\frac{N}{2})} \\ &= \sum_{n=0}^{N/2-1} x(n)W_N^{kn} + W_N^{k\frac{N}{2}} \sum_{n=0}^{N/2-1} x\left(n + \frac{N}{2}\right)W_N^{kn} \end{aligned} \quad (3.23)$$

donde  $W_N^{k\frac{N}{2}}$  se saca fuera del segundo sumatorio por que no depende de  $n$ , además se tiene que

$$W_N^{k\frac{N}{2}} = e^{-jk\pi} = (e^{-j\pi})^k = (\cos \pi - j \sin \pi)^k = (-1)^k \quad (3.24)$$

por lo que la ecuación (3.23) puede reescribirse como

$$X(k) = \sum_{n=0}^{N/2-1} \left[ x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{kn} \quad (3.25)$$

Ahora bien, la ecuación (3.25) puede ser expresada en términos de los  $k$  pares y los  $k$  impares como sigue

$$X(2k) = \sum_{n=0}^{N/2-1} \left[ x(n) + (-1)^{2k} x\left(n + \frac{N}{2}\right) \right] W_N^{2kn} \quad (3.26)$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} \left[ x(n) + (-1)^{2k+1} x\left(n + \frac{N}{2}\right) \right] W_N^{(2k+1)n} \quad (3.27)$$

Sabiendo que  $(-1)^{2k} = 1$  y que  $(-1)^{2k+1} = -1$ , además

$$W_N^{2kn} = (e^{-j4\pi})^{\frac{kn}{N}} = (\cos 4\pi - j \sin 4\pi)^{\frac{kn}{N}} = (1)^{\frac{kn}{N}} = 1 \quad (3.28)$$

y

$$W_N^{(2k+1)n} = W_N^{(2kn+n)} = W_N^{2kn} W_N^n = W_N^n \quad (3.29)$$

las ecuaciones (3.26) y (3.27) pueden ser reescritas como

$$X(2k) = \sum_{n=0}^{N/2-1} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (3.30)$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (3.31)$$

Si definimos las secuencias de  $\frac{N}{2}$  puntos  $g1(n)$  y  $g2(n)$  como

$$g1(n) = x(n) + x\left(n + \frac{N}{2}\right) \quad (3.32)$$

$$g2(n) = \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \quad (3.33)$$

para  $n = 0, 1, 2, \dots, \frac{N}{2} - 1$ , entonces

$$X(2k) = \sum_{n=0}^{N/2-1} g1(n) \quad (3.34)$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} g2(n) \quad (3.35)$$

### 3.6. Resumen del capítulo

En este capítulo se ha descrito un SDC y los bloques principales que lo componen. Se han presentado brevemente algunos de los esquemas de modulación digital existentes, los cuales, se basan en la modificación de tres parámetros de una señal sinusoidal, a saber, *amplitud*, *frecuencia* y *fase*. Se ha mencionado también, que la forma más conveniente de representar un esquema de modulación digital es por medio de su *espacio de señal* o *constelación*, asociando a cada símbolo un valor de magnitud y uno de fase. Aunque resulta más práctico separar la señal en dos componentes independientes, conocidas como *componente en fase* y *componente en cuadratura*, donde este tipo de representación puede asociarse a la mayor parte de las técnicas de modulación digital. Por otro lado, se ha mostrado el funcionamiento de un modulador QAM, en el cual, un flujo de bits es separado en flujos más pequeños para generar un conjunto

*nivel-fase* representativo del flujo de bits recibido. Posteriormente este conjunto nivel-fase es utilizado para generar cada símbolo QAM.

En la parte central del capítulo, se detallan los conceptos en los que se fundamenta OFDM, mostrando que el *principio de ortogonalidad* entre señales permite que múltiples señales de información puedan ser transmitidas y detectadas sobre un canal de comunicación sin interferencia. Por otra parte, también se expone que uno de los mayores logros para resolver la complejidad de implementar un sistema OFDM es haciendo uso de la DFT, ya que permite una implementación completamente digital del proceso de modulación y demodulación OFDM. Además, las ventajas que proporciona una modulación multiportadora frente a una modulación de portadora única, hacen que OFDM represente la técnica adecuada para uso en ambientes PLC. Sin embargo, uno de los mayores inconvenientes de OFDM es su alta susceptibilidad a errores de sincronización.

Para finalizar el capítulo, se presentó el desarrollo del algoritmo de *diezmado en frecuencia* para el cálculo de la FFT. Este algoritmo corresponde al implementado en el presente trabajo como parte fundamental del proceso de modulación y demodulación OFDM.

# Capítulo 4

## Implementación del módem

En el presente capítulo, se describen los diferentes módulos que componen al módem PLC basado en OFDM. Se inicia con una descripción general de las etapas que lo constituyen (transmisor, receptor, etc.), para luego, especificar brevemente los recursos de hardware y software utilizados, siguiendo con los algoritmos implementados en cada etapa y finalizando con la explicación del módulo de adquisición diseñado.

### 4.1. Arquitectura general

En la figura 4.1 se puede observar la estructura general de la arquitectura propuesta para el módem PLC.

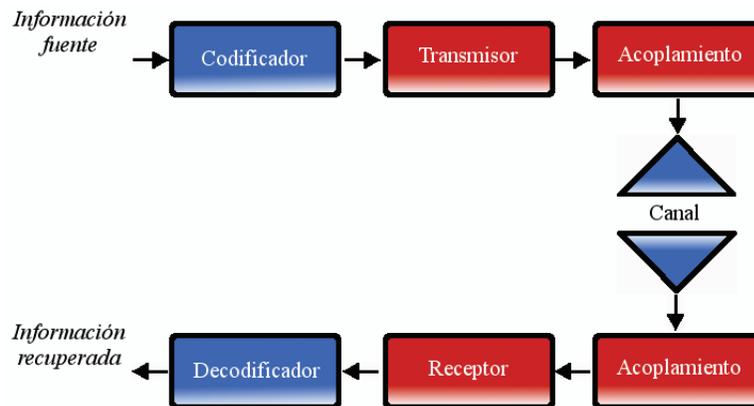


Figura 4.1: Arquitectura del módem propuesto.

Los bloques de codificación y de decodificación, se han omitido en la implementación del módem, ya que estos bloques corresponden a otro proyecto desarrollado de manera conjunta al presente trabajo[27].

En los párrafos siguientes, se detalla de manera general cada uno de los bloques de la arquitectura del módem mostrados en el esquema anterior, explicando brevemente las operaciones llevadas a cabo en los procesos de transmisión y recepción de información.

## 4.2. Transmisor

En la figura 4.2 se muestra el esquema del transmisor implementado. En un principio, los  $M$  bits que recibe el transmisor corresponden a bits sin codificar, dado que, como se mencionó anteriormente, el bloque de codificación no fue implementado. Estos  $M$  bits recibidos, son tomados en grupos de  $m_k$  bits para ser mapeados a un punto complejo  $p_k = a_k + jb_k$  de la constelación QAM. Posteriormente, un conversor a simetría hermitiana coloca los datos en el formato  $q_{k'} = \{0, p_k, 0, p_k^*\}$ , lo cual permite que al computar la IFFT, la parte imaginaria del resultado sea prácticamente cero, tomando en cuenta que el conversor D/A no manipula valores complejos. La conversión a simetría hermitiana genera  $2K + 2$  datos complejos, que luego son utilizados por la IFFT para realizar la modulación OFDM propiamente dicha, dando como resultado un símbolo OFDM con  $2K + 2$  valores reales. En la etapa siguiente, al símbolo OFDM se le agregan  $l$  datos que corresponden al prefijo cíclico, lo que origina un símbolo OFDM de tamaño  $2K + 2 + l$ . Una vez agregado el prefijo cíclico, el símbolo OFDM pasa al bloque de construcción de trama, donde formará parte de una trama OFDM de tamaño específico. Consecuentemente, a cada trama se le inserta un número determinado de símbolos piloto para propósitos de sincronización y recuperación de información del lado receptor. Acto seguido, la trama OFDM es enviada al conversor D/A para su transmisión.

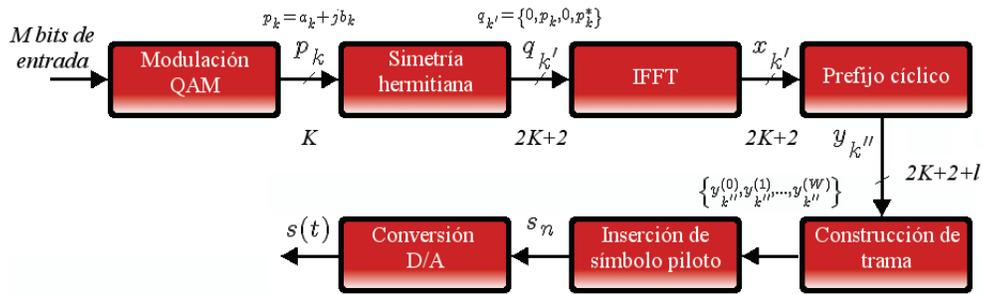


Figura 4.2: Transmisor.

## 4.3. Receptor

Básicamente, el receptor realiza la operación inversa al transmisor, aunque para la correcta recuperación de la información precisa de etapas adicionales.

En la figura 4.3 se puede observar el esquema correspondiente al receptor implementado. Como se puede notar, la señal  $r(t)$  que recibe el receptor puede o no ser información enviada por el transmisor, donde además, el grado de perturbación de la señal depende de las condiciones del canal. Por lo tanto, la primera tarea del receptor consiste en determinar si la señal recibida pertenece a una señal útil que pueda contener información enviada por el transmisor, y si éste es el caso, localizar el inicio de la información enviada. Por consiguiente, después de la etapa de conversión A/D, donde la señal  $r(t)$  es discretizada en  $r_n$ , se tiene el bloque de correlación IG, en el cual, el valor máximo de correlación de  $r_n$ , al ser comparado con un valor de umbral, indica aproximadamente el inicio de la información. Al proceso anterior

se le conoce comunmente como sincronización burda y posibilita la localización del inicio de la información en un rango de  $\pm v$  muestras.

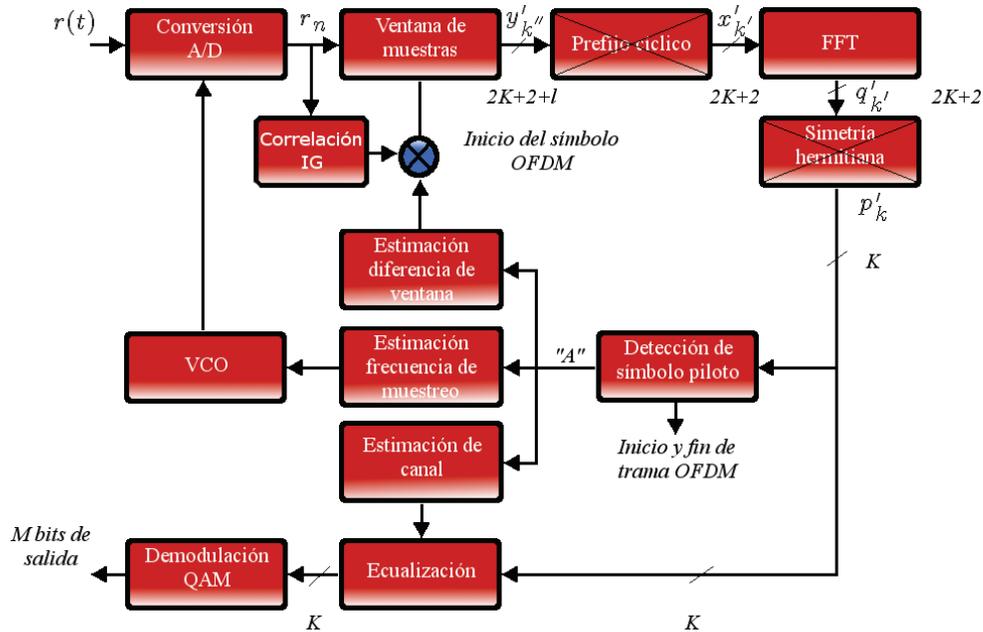


Figura 4.3: Receptor.

Con el resultado de la *sincronización burda* se ajusta una ventana de muestras con longitud  $2K + 2 + l$ , esto es, de longitud de un símbolo OFDM, incluyendo al intervalo de guarda. Posteriormente, al símbolo OFDM contenido en la ventana se le remueve el prefijo cíclico, se le aplica la FFT para la demodulación OFDM y se elimina la simetría hermitiana; quedando de esta manera únicamente la señal analítica  $p'_k = a_k + jb_k$ . Más tarde, la señal analítica pasa por las etapas de ecualización y de detección de símbolo piloto.

En la etapa de detección de símbolo piloto, la señal analítica recibida es comparada con la señal analítica de un símbolo piloto conocido como símbolo A. Cuando la señal recibida corresponde a la señal del símbolo A, hay tres procesos que se llevan a cabo:

- Estimación de diferencia de ventana
- Estimación de frecuencia de muestreo
- Estimación de canal

La estimación de diferencia de ventana se basa en la observación de fase de la señal recibida para estimar el desplazamiento en tiempo  $\hat{\epsilon}_t$  de la ventana actual, y una vez conociendo el desplazamiento, se realiza la corrección de la posición de la ventana aplicando un corrimiento de  $v$  muestras al flujo de datos recibidos. Por otro lado, la estimación de frecuencia de muestreo emplea la señal de dos símbolos piloto recibidos para estimar la diferencia de la frecuencia del reloj, basándose en el hecho de que dos símbolos piloto diferentes pueden sufrir rotaciones con diferente fase, y que al ser comparadas, permiten estimar el valor de la diferencia de reloj, para luego, con este valor, adecuar la frecuencia de muestreo del conversor A/D, por medio del

control de un VCO. Por último, la estimación de canal utiliza la señal recibida para estimar la respuesta del canal subportadora a subportadora, aplicando el criterio ML en el dominio de la frecuencia. El resultado de la estimación después es utilizado por la etapa de ecualización para compensar la distorsión introducida por el canal.

A los procesos anteriores en conjunto se les conoce como procesos de *sincronización fina*, donde el enfoque empleado para la estimación se basa en el principio de rotación de fase en el dominio espectral, es decir, por la comparación de los defasamientos observados sobre diferentes subportadoras. De esta manera, la correcta detección del símbolo piloto juega un papel muy importante, y si bien, el símbolo piloto está pensado principalmente para propósitos de sincronización fina, también es un indicador que permite discernir el inicio o final de una trama sin tener que realizar el procesamiento de los datos para ello.

Como parte final del receptor se encuentra la etapa de demodulación QAM. En esta etapa, cada punto  $p'_k = a_k + jb_k$  de la señal analítica, previamente compensada por la etapa de ecualización, se decodifica en grupos  $m_k$  bits, donde el número de bits en error de los  $M$  bits de salida, dependen de las condiciones del canal y de lo efectivo que resulten las etapas de sincronización fina.

## 4.4. Acoplamiento

La etapa de acoplamiento es la encargada de superponer una señal de comunicación sobre una señal de potencia, lo cual implica que el circuito de acoplamiento debe ser cuidadosamente diseñado para una óptima compatibilidad entre los dos sistemas. Por un lado, la etapa de acoplamiento manipula una señal de baja frecuencia y alta potencia como lo es la señal que distribuye la energía eléctrica, y por el otro, una señal de alta frecuencia y baja potencia, como lo es la señal de comunicación. En la figura 4.4 se presenta el circuito propuesto para la etapa de acoplamiento del módem. El circuito proviene de [22] y la descripción de las partes más representativas se presenta a continuación.

El circuito de acoplamiento mostrado en el esquema 4.4 es conocido como de modo diferencial o también de modo balanceado, y corresponde a la configuración más ampliamente adoptada por la mayoría de los fabricantes de módems para acoplar la señal de comunicación a la red eléctrica. En esta configuración, la señal de comunicación es aplicada al cable de fase, mientras que el retorno es por el cable de neutro. El aislamiento galvánico se logra por medio del transformador T1, lo cual evita la conexión física del módem a la red eléctrica. T1 además de servir como aislamiento, actúa como un filtro pasa-altos, a la vez que se aprovecha para lograr el acoplamiento de impedancias entre la línea eléctrica y el módem. El capacitor C1 es un capacitor de alta frecuencia y alto voltaje, con características típicas de los capacitores supresores de radio-interferencia, permitiendo el paso de la señal de comunicación, a la vez que filtra la senoide de la red eléctrica. L1 actúa como un filtro de la señal de comunicación, impidiendo el paso de esta a la fuente de alimentación del sistema. R1, en combinación con los diodos D1 y D2, protegen al circuito de comunicación de transitorios rápidos. El fusible y V1 son los encargados de proteger a todo el sistema de corto circuitos, sobre tensiones y sobre voltajes.

Cabe señalar, que la etapa de acoplamiento no fue construida, debido a que las pruebas realizadas con el módem sólo son a nivel de simulación. Aunque la construcción del circuito propuesto se deja como trabajo futuro.

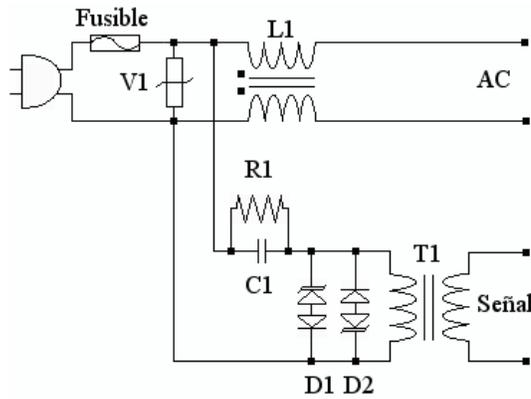


Figura 4.4: Circuito de acoplamiento.

## 4.5. Hardware

A continuación se describen los recursos de hardware empleados en el desarrollo del módem.

### 4.5.1. TMS320C6416T DSK

El kit de desarrollo TMS320C6416T DSK es una herramienta de hardware de bajo costo que permite evaluar la funcionalidad del procesador digital de señales TMS320C6416 de Texas Instruments. El diagrama de bloques del kit se muestra en la figura 4.5.

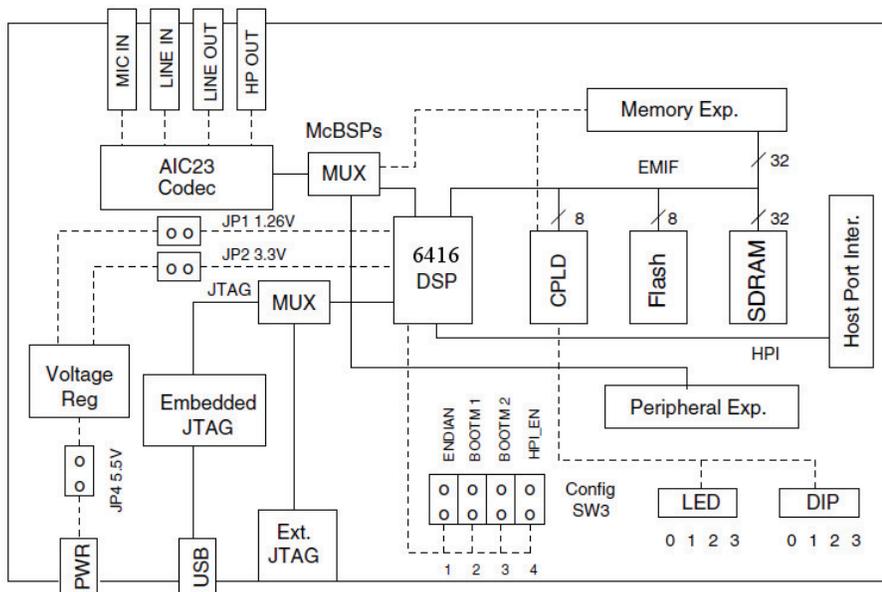


Figura 4.5: Diagrama de bloques del TMS320C6416T DSK.

Entre las principales características del hardware que incluye el kit se encuentran las siguientes:

- Procesador digital de señales (DSP) con velocidad de hasta 1GHz.
- Códec AIC23 para procesamiento de audio.
- 16Mb de SDRAM.
- 512Kb de memoria flash no volátil.
- Interfaz USB para emulación JTAG.
- Opciones de arranque configurables.
- Leds y dip switches de propósito general.

#### 4.5.2. Spartan 3E Breakout Board

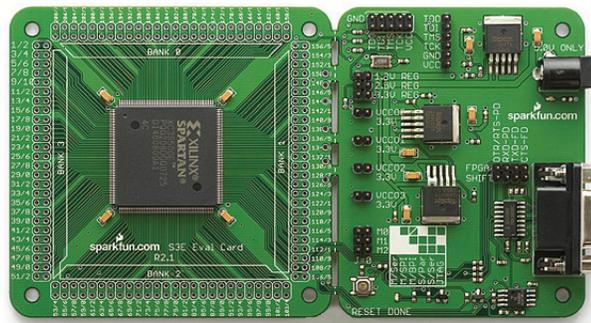


Figura 4.6: Módulo de evaluación Xilinx Spartan 3E.

Las características principales de esta tarjeta son las siguientes:

- Xilinx Spartan 3E XC3S500E-PQG208.
- 16Mb PROM SPI.
- Interfaz JTAG.
- Reguladores de voltaje de 1.2, 2.5 y 3.3V.

#### 4.5.3. ADS5485EVM

El módulo ADS5485EVM provee una plataforma para la evaluación del convertidor analógico-digital ADS5485 de Texas Instruments. Este convertidor tiene una resolución de 16 bits, con una arquitectura *pipeline* que le permiten tasas de hasta 200 MSPS operando con voltajes de 5V y 3.3V.

#### 4.5.4. DAC5672EVM

El módulo DAC5672EVM proporciona la interfaz necesaria para la evaluación del convertidor digital-analógico DAC5672. Este convertidor tiene una resolución de 14 bits con una tasa de hasta 275 MSPS operando desde 3 a 3.6V.

## 4.6. Software

### 4.6.1. Code Composer Studio 3.3

Code Composer Studio(CCS) es un ambiente de desarrollo especialmente diseñado para ayudar a los programadores en la creación y prueba en tiempo real, de las aplicaciones embebidas para una gran gama de familias de dispositivos de Texas Instruments. Proporciona herramientas para configuración, construcción, depuración, exploración y análisis de los programas. Las fases en el ciclo de desarrollo en CCS se muestran en la figura 4.7.

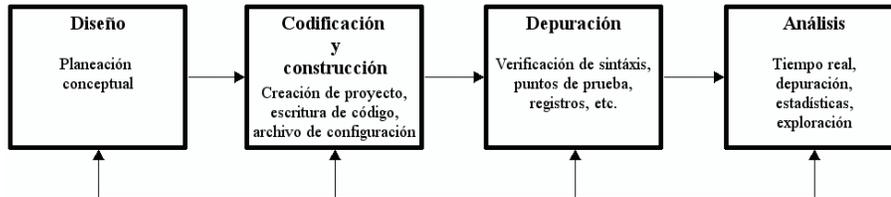


Figura 4.7: Fases del ciclo de desarrollo en CCS.

### 4.6.2. Xilinx ISE Desing Suite 12.2

Xilinx ISE Design Suite representa un conjunto de herramientas desarrolladas por la compañía Xilinx para el análisis y síntesis de diseños descritos en diagramas esquemáticos o en lenguaje de descripción de hardware (HDL). El conjunto de herramientas disponible, permiten hacer la verificación, simulación, análisis, síntesis y programación de los diseños.

### 4.6.3. MATLAB

MATLAB (Laboratorio de matrices) es un lenguaje de alto nivel diseñado principalmente para la computación numérica, con capacidades para manejo de matrices, implementación de algoritmos, creación de interfaces de usuario, graficación de funciones y datos, entre otros. Es ampliamente utilizado en escuelas y centros de investigación, así como en la industria. Las principales áreas del conocimiento que abarca son la ciencia, ingeniería y economía. Combina la excelente capacidad computacional con capacidades gráficas fáciles de usar. Contiene un gran conjunto de funciones programadas (archivos m) para la generación, análisis, procesamiento y despliegue de señales. Bibliotecas complementarias (toolboxes) que suplementan a la biblioteca de funciones de MATLAB en áreas de aplicaciones específicas.

## 4.7. Algoritmos

En esta sección se presentan los algoritmos y su respectivo pseudocódigo de la etapas más relevantes del diseño propuesto. Sin embargo, en el CD que se entrega adjunto a este documento, se incluyen los códigos en lenguaje C y MATLAB.

### 4.7.1. Modulador y demodulador QAM

El modulador QAM es el encargado de generar los símbolos de modulación para cada una de las portadoras de OFDM. Estos símbolos de modulación son de la forma  $s_k = a_k + jb_k$  y son generados tomando grupos de  $m_k$  bits de un flujo de  $M$  bits. Los primeros  $\frac{m_k}{2}$  bits son codificados en la componente en *fase* ( $a_k$ ) y los  $\frac{m_k}{2}$  bits restantes son codificados en la componente en *cuadratura* ( $b_k$ ). Los  $m$  bits que se requieren para formar un símbolo de modulación dependen de la selección del formato QAM a utilizar. Por ejemplo, para una modulación 16-QAM, se necesitan  $m = 4$  bits para generar un símbolo de modulación, ya que  $2^m = 2^4 = 16$ . Los formatos QAM que el modulador permite generar son 4-QAM, 16-QAM, 64-QAM, 128-QAM y 256-QAM.

Para una robustez mayor frente al ruido, el modulador implementa la codificación de los  $m$  bits en código Gray. Esto permite que símbolos de modulación adyacentes difieran únicamente en un bit, de esta manera, si los efectos del ruido causan que la constelación se mueva del umbral de decisión, únicamente 1 bit de los  $m$  bits estará en error.

A continuación, se presenta el pseudocódigo para el algoritmo del modulador implementado. La función **BIN2GRAY()** realiza la conversión de binario a código Gray y la función **LEVELMAP()** mapea el nivel QAM de acuerdo al código Gray.

**Algoritmo 4.7.1:** MODULATOR( $M, m$ )

```

 $N \leftarrow 2^m$ 
for  $k \leftarrow 0$  to  $N$ 
  do
     $m_k \leftarrow M(k)$ 
     $m_{k_1} \leftarrow m_k(0, \dots, \frac{m}{2} - 1)$ 
     $m_{k_2} \leftarrow m_k(\frac{m}{2}, \dots, m - 1)$ 
     $G_a \leftarrow \text{BIN2GRAY}(m_{k_1})$ 
     $G_b \leftarrow \text{BIN2GRAY}(m_{k_2})$ 
     $a_k \leftarrow \text{LEVELMAP}(G_a)$ 
     $b_k \leftarrow \text{LEVELMAP}(G_b)$ 
     $s_k \leftarrow a_k + jb_k$ 

```

Por otra parte, el demodulador es el encargado de recuperar los bits mapeados en cada uno de los símbolos de modulación recibidos. El problema fundamental en el demodulador es decidir en base a la señal recibida  $r_k = a_k + b_k$ , que señal, del espacio de señales  $S = \{s_0, s_1, \dots, s_i\}$ ,  $i = 0, 1, \dots, N - 1$  fue enviada del lado transmisor, donde  $N = 2^m$  es el número de señales de la constelación QAM. Para resolver el problema, el demodulador utiliza la distancia mínima dada en la ecuación (4.1) como regla de decisión para determinar a que punto de la constelación QAM pertenece la señal recibida.

$$\delta_{ML}(r_k) = \arg \min \|r_k - s_i\| \quad (4.1)$$

En seguida se presenta el pseudocódigo para el algoritmo del demodulador implementado. La función **LEVEL2GRAY()** utiliza la regla de decisión para determinar el código Gray de la componente de fase y cuadratura de la señal recibida. La función **GRAY2BIN()** realiza la conversión de código Gray a código binario.

**Algoritmo 4.7.2:** DEMODULATOR( $r_k, m$ )

```

 $N \leftarrow 2^m$ 
for  $k \leftarrow 0$  to  $N$ 
  do
     $a_k \leftarrow \Re\{r_k\}$ 
     $b_k \leftarrow \Im\{r_k\}$ 
     $G_a \leftarrow \text{LEVEL2GRAY}(a_k)$ 
     $G_b \leftarrow \text{LEVEL2GRAY}(b_k)$ 
     $m_{k_1} \leftarrow \text{GRAY2BIN}(G_a)$ 
     $m_{k_2} \leftarrow \text{GRAY2BIN}(G_b)$ 
     $m_k \leftarrow (m_{k_1}, m_{k_2})$ 
     $M(k) \leftarrow m_k$ 

```

**4.7.2. FFT, IFFT**

La FFT e IFFT implementadas se basan en el algoritmo de diezmado en frecuencia (DIF) presentado en el capítulo anterior, donde las ecuaciones (4.2) y (4.3) implementan la *mariposa* más simple del algoritmo.

$$g1(n) = x(n) + x\left(n + \frac{N}{2}\right) \quad (4.2)$$

$$g2(n) = \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \quad (4.3)$$

Para la implementación, los factores de fase  $W_N^n$  son previamente calculados y mapeados a memoria. Esto permite disminuir el tiempo de ejecución del algoritmo. El cómputo de la FFT e IFFT utiliza la técnica de cálculo *in-place*, es decir, que el mismo buffer de entrada sirve como buffer de almacenamiento de los cálculos de salida del algoritmo, eficientando de esta manera el uso de la memoria disponible en el dispositivo utilizado. Por otro lado, el orden de los datos se encuentran en *bit reversing*, por lo que se implementa la función para el reordenamiento de los mismos. Además, para el caso de la IFFT se aplica un factor de escalamiento.

A continuación se preseta el pseudocódigo para el algoritmo de la FFT e IFF. La función **BITREVERSING()** se encarga del reordenamiento de los datos de salida y la función **SCALE()** aplica el factor de escalamiento a los datos para el caso de la IFFT.

**Algoritmo 4.7.3:** FFT( $Y, N$ )

```

NumStages ← log2(N)
LegDiff ← N/2
Step ← MAXFFT/N
for i ← 0 to NumStages
  do {
    n ← 0
    for j ← 0 to LegDiff
      do {
        for UpperLeg ← j to N
          do {
            LowerLeg ← UpperLeg + LegDiff
            Y[UpperLeg] ← Y[UpperLeg] + Y[LowerLeg]
            Y[LowerLeg] ← {Y[UpperLeg] - Y[LowerLeg]} WNn
            UpperLeg ← UpperLeg + 2 * LegDiff
          }
        n ← n + Step
      }
    LegDiff ← LegDiff/2
    Step ← Step * 2
  }
BITREVERSING(Y)

```

**Algoritmo 4.7.4:** IFFT( $Y, N$ )

```

NumStages ← log2(N)
LegDiff ← N/2
Step ← MAXFFT/N
for i ← 0 to NumStages
  do {
    n ← 0
    for j ← 0 to LegDiff
      do {
        for UpperLeg ← j to N
          do {
            LowerLeg ← UpperLeg + LegDiff
            Y[UpperLeg] ← Y[UpperLeg] + Y[LowerLeg]
            Y[LowerLeg] ← {Y[UpperLeg] - Y[LowerLeg]} (WNn)*
            UpperLeg ← UpperLeg + 2 * LegDiff
          }
        n ← n + Step
      }
    LegDiff ← LegDiff/2
    Step ← Step * 2
  }
BITREVERSING(Y)
SCALE(Y, N)

```

**4.7.3. Módulo de correlación IG**

El módulo de correlación aprovecha las propiedades del prefijo cíclico para identificar el inicio de un símbolo OFDM. La función de correlación sigue un criterio de máxima correlación, y se basa en el estimador ML para  $\delta_t$  sobre un canal gaussiano definido como

$$\delta_t^{ML} = \arg \max_{\delta_t} \left( \Psi(\delta_t) - \frac{\rho}{2} \Phi(\delta_t) \right) \quad (4.4)$$

con

$$\begin{cases} \Psi(\delta_t) = \sum_{m=\delta_t}^{\delta_t+L-1} r_n r_{n+N} \\ \Phi(\delta_t) = \sum_{m=\delta_t}^{\delta_t+L-1} r_n^2 + r_{n+N}^2 \end{cases}$$

donde  $\rho = \frac{SNR}{SNR+1}$ ,  $\Psi(\delta_t)$  es la correlación de la señal recibida con una versión retardada de la misma y  $\Phi(\delta_t)$  es un término de energía.

Si se mantiene una SNR muy pequeña, es decir,  $\rho \rightarrow 0$ , se obtiene una versión subóptima del módulo de correlación ML mostrado en (4.4).

$$\delta_t^{MC} = \arg \max_{\delta_t} (\Psi(\delta_t)) \quad (4.5)$$

En la práctica, el cálculo de  $\Psi(\delta_t)$  únicamente se realiza al inicio del proceso, para luego ser actualizado con cada muestra recibida tomando en cuenta que

$$\Psi(\delta_t + 1) = \Psi(\delta_t) + r_{\delta_t+L} r_{\delta_t+L+N} - r_{\delta_t} r_{\delta_t+N} \quad (4.6)$$

lo cual reduce significativamente el costo computacional.

El pseudocódigo que se muestra en seguida realiza la sincronización burda de una trama OFDM, empleando para ello la función de máxima correlación **CROSSCORRELATION()**. Se emplea una ventana de observación de  $2N+L$  muestras y se asume que el pico de correlación existe cuando la ventana contiene  $N+L$  muestras de un símbolo OFDM. Además, se ha utilizado un valor de umbral para detectar el inicio de la trama en presencia de ruido.

**Algoritmo 4.7.5:** SYNC\_COARSE( $m, r, N, L$ )

```

Ψ(δt) = CROSSCORRELATION(m, r, N, L)
for i ← 0 to 2N + L
do
  Ψ(δt + 1) = Ψ(δt) + rδt+Lrδt+L+N - rδtrδt+N
  if (Ψ(δt) > Threshold) and (Ψ(δt) > Ψmax)
  then
    Ψmax = Ψ(δt)
    δtmax = δt
  else if (Ψmax > Threshold) and (Ψ(δt) < Threshold)
  then
    exit for
  m ← m + 1
return (δtmax)

```

#### 4.7.4. Detector de símbolo piloto

El detector utiliza la atenuación dada por  $G = \frac{Y}{X}$  y el error del tal atenuación para saber cuando un símbolo recibido corresponde al símbolo piloto. Las ecuaciones siguientes son las implementadas en el detector

$$g = \frac{1}{N} \sum_{i=0}^{N-1} \frac{Y(i)}{X(i)} \quad (4.7)$$

$$v = \frac{1}{N} \sum_{i=0}^{N-1} \left( \frac{Y(i)}{X(i)} - g \right)^2 \quad (4.8)$$

$$e = \sqrt{v} \quad (4.9)$$

donde  $Y$  es la magnitud del símbolo recibido,  $X$  la magnitud del símbolo piloto,  $N$  la longitud de un símbolo,  $g$  es la ganancia promedio,  $v$  la varianza promedio y  $e$  el error.

El pseudocódigo siguiente muestra el algoritmo del detector. La función **SYNCFINE()** utiliza el símbolo piloto recibido para los procesos de sincronización fina.

**Algoritmo 4.7.6:** SYMBOLDETECT( $Y_n, X_A$ )

```

 $g \leftarrow \text{GAIN}(Y_n, X_A)$ 
 $v \leftarrow \text{VARIANCE}(Y_n, X_A, g)$ 
 $e \leftarrow \text{sqr}(v)$ 
if ( $g > \text{ThGain}$ ) and ( $e < \text{ThError}$ )
  then {SYNCFINE( $Y_n$ )}

```

#### 4.7.5. Estimador de diferencia de ventana

El estimador de diferencia de ventana se basa en las diferencias de fase del símbolo piloto recibido  $y_p$  y del símbolo piloto conocido  $x_A$  para estimar el desplazamiento de la ventana actual. El estimador es del tipo LLS (Linear Last Square) y busca determinar la pendiente de la recta que se acerca mejor a los elementos del vector  $\Delta\phi_p$  que contiene las diferencias de fase. En [28] se puede encontrar el estudio completo del estimador.

Las expresiones para el estimador se muestran a continuación

$$\hat{\epsilon}_t^{GLLS}(p) = \frac{1}{2\pi} \frac{N_u \sum_{k \in H} k \Delta\phi_k(p) - \sum_{k \in H} k \sum_{k \in H} \Delta\phi_k(p)}{N_u \sum_{k \in H} k^2 - (\sum_{k \in H} k)^2} \quad (4.10)$$

$$\hat{\epsilon}_t^{GLLS}(p) = \frac{1}{2\pi} \text{grad}_H [\Delta\phi_p] \quad (4.11)$$

donde  $N_u$  es el número de subportadoras útiles.

El pseudocódigo que a continuación se presenta muestra el algoritmo implementado para la estimación de diferencia de ventana. La función **UNWRAP()** aplica un algoritmo de despliegue de fase antes de la aplicación del estimador GLLS.

**Algoritmo 4.7.7:** SYNC\_FINE( $y_p, x_A$ )

```

 $X_A^H \leftarrow \text{diag}(x_A)$ 
 $\Delta\phi_p \leftarrow \angle(X_A^H y_p)$ 
 $\Delta\phi_p \leftarrow \text{UNWRAP}(\Delta\phi_p)$ 
 $\hat{\epsilon}_t(p) \leftarrow \frac{1}{2\pi} \text{grad}_H [\Delta\phi_p]$ 
 $\delta_t \leftarrow \delta_t - \hat{\epsilon}_t$ 

```

#### 4.7.6. Estimador de frecuencia de muestreo

Al igual que el estimador de diferencia de ventana, este estimador se basa en la observación de la diferencia de fase para poder estimar la desviación en la frecuencia de muestreo. Pero, a diferencia del estimador de diferencia de ventana, el estimador de frecuencia de muestreo

utiliza la información de dos símbolos piloto recibidos para llevar a cabo la estimación, ya que, en principio, dos símbolos piloto diferentes pueden sufrir rotaciones con diferente fase, y que al ser comparadas, permiten estimar el valor de la diferencia de reloj. Las expresiones para el estimador se presentan a continuación

$$\hat{\epsilon}_f^{GLLS}(p) = \frac{1}{2\pi} \frac{N}{M(N+D)+D} \frac{N_u \sum_{k \in H} k \Delta\varphi_k(p) - \sum_{k \in H} k \sum_{k \in H} \Delta\varphi_k(p)}{N_u \sum_{k \in H} k^2 - (\sum_{k \in H} k)^2} \quad (4.12)$$

$$\hat{\epsilon}_f^{GLLS}(p) = \frac{1}{2\pi} \frac{N}{M(N+D)+D} \text{grad}_H [\Delta\varphi_p] \quad (4.13)$$

#### 4.7.7. Estimador de canal

El estimador de canal se basa en una implementación del sistema de comunicaciones en *tramas*. Es decir, ráfagas de  $N$  símbolos OFDM a un tiempo. Donde agregando *símbolos piloto* al inicio de cada trama, posibilita la estimación de canal, asumiendo que la trama es corta, y que además, el canal se mantiene estático en el intervalo de duración de la trama. De esta manera, si la señal recibida es

$$Y(k) = H(k)X(k) + Z(k) \quad (4.14)$$

donde  $k$  representa el índice de la subportadora,  $H$  el canal,  $X$  el símbolo piloto y  $Z$  el ruido. Entonces la manera más simple de estimar el estado del canal viene dado por

$$H_{en} = \frac{Y}{X} \quad (4.15)$$

Sin la presencia de ruido, la expresión (4.15) proporciona una correcta estimación del estado del canal, pero cuando el ruido se hace presente, la estimación dada puede resultar errónea. Para hacer frente a los problemas de ruido y obtener una respuesta más aproximada a las condiciones del canal, se ha implementado una técnica de eliminación de ruido por *ventaneo* en el dominio temporal, ver [21]. Esta técnica consiste en transformar la respuesta en frecuencia de las condiciones del canal al dominio temporal, para después realizar un *ventaneo*, con el fin de conservar únicamente las muestras dentro del tiempo de dispersión máxima dado por la longitud del intervalo de guarda, asumiendo que las muestras restantes representan únicamente ruido.

El pseudocódigo para la estimación de la respuesta del canal se presenta a continuación, donde  $H_{en}$  es la respuesta en frecuencia del canal con ruido,  $N$  la longitud del símbolo OFDM y  $L$  la longitud del intervalo de guarda.

**Algoritmo 4.7.8:** ESTIMATION( $H_{en}, N, L$ )

```

 $L_h \leftarrow L/2$ 
 $h_{tn} \leftarrow \text{IFFT}(H_{en})$ 
 $h_{tu}(0 \text{ to } L_h - 1) \leftarrow h_{tn}(0 \text{ to } L_h - 1)$ 
 $h_{tu}(N - L_h \text{ to } N - 1) \leftarrow h_{tn}(N - L_h \text{ to } N - 1)$ 
 $H_e \leftarrow \text{FFT}(h_{tu})$ 

```

#### 4.7.8. Ecualizador

El ecualizador utiliza la respuesta estimada del canal  $H_e$  para compensar los efectos del canal sobre la señal recibida por medio de la expresión siguiente

$$Y_g = Y_n * \frac{1}{H_e} \quad (4.16)$$

donde  $Y_g$  es el símbolo compensado,  $Y_n$  es el símbolo recibido y  $H_e$  es la respuesta en frecuencia del canal.

#### 4.8. Módulo de adquisición

El módulo de adquisición es implementado en el FPGA Spartan-3E y se emplea el bus EMIF del DSP TMS320C6416T como interfaz para el intercambio de información a través de canales EDMA. El módulo se compone de un controlador EMIF, controlador de ADC, controlador de DAC y FIFOs asíncronas para el almacenamiento temporal de datos. La figura 4.8 muestra el diagrama de bloques del módulo de adquisición.

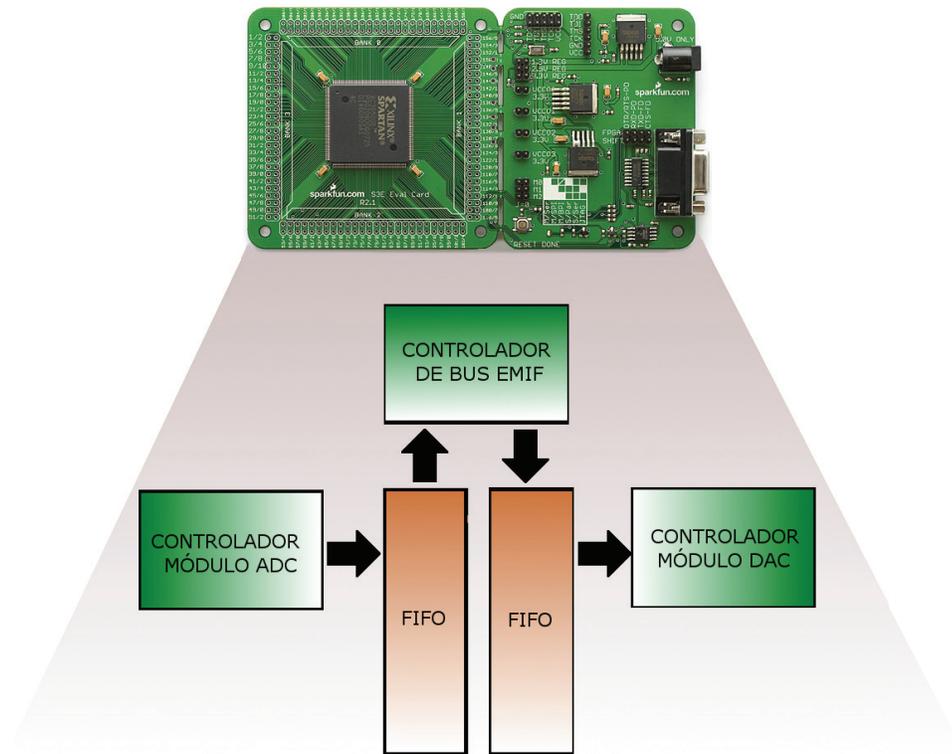


Figura 4.8: Diagrama de bloques del módulo de adquisición.

##### 4.8.1. Controlador EMIF

El controlador EMIF es el encargado del manejo de las señales generadas por el bus EMIFA en las transferencias de información entre el módulo de adquisición y la memoria del

DSP. Para ello, hace uso del espacio de direcciones CE2 del EMIFA como espacio para tales transferencias. Por otro lado, las peticiones de transferencias que el controlador es capaz de soportar son las siguientes:

- Petición de lectura. Esta petición se recibe cuando se requiere una transferencia de datos desde la memoria del DSP hacia el módulo de adquisición.
- Petición de escritura. Esta petición se recibe cuando se requiere una transferencia de datos desde el módulo de adquisición hacia la memoria del DSP.
- Petición de comando. Esta petición se recibe cuando el DSP requiere enviar un comando hacia el módulo de adquisición.

Cada una de las peticiones anteriores corresponde a una dirección en el espacio de direcciones CE2. Esto permite que, empleando un decodificador de direcciones, se pueda identificar el tipo de petición recibida. Una vez identificada la petición, el controlador utiliza las mismas señales del bus para generar las señales de habilitación, reloj, etc., que requiere el módulo de adquisición para llevar a cabo la operación solicitada.

En la figura 4.9 se puede observar el controlador implementado.

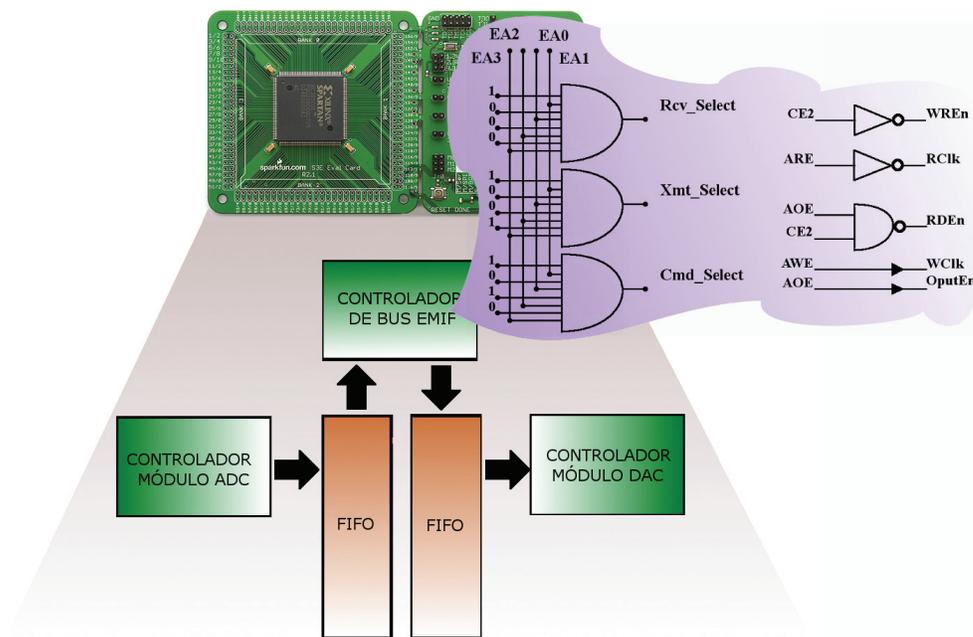


Figura 4.9: Controlador EMIF.

#### 4.8.2. Controlador ADC

El controlador ADC explota algunas de las características disponibles en el dispositivo FPGA para el control del convertor A/D ADS5485. Entre estas características se encuentran las siguientes:

- Drivers diferenciales. Estos drivers son empleados para la interfaz entre las líneas de datos y de control del convertor A/D con el FPGA.

- DCM - Digital Clock Manager. El DCM es utilizado para generar las señales de reloj necesarias para el control del conversor A/D.
- IDDR2 - Double Data Rate Input D Flip-Flop. El IDDR2 es utilizado para la captura de datos en los flancos de subida y de bajada de la señal DRY del conversor A/D.

Para el proceso de captura, el controlador hace uso de las señales DRY\_P y DRY\_M procedentes del conversor A/D. Estas señales indican cuando una muestra válida se encuentra presente en el bus de datos del conversor. Las señales anteriores, sirven como entrada de reloj al DCM encargado de generar las señales de reloj para el IDDR2. El IDDR2 es un componente con dos entradas de reloj, C1 y C2, las cuales sirven para capturar el dato D presente a su entrada en cada flanco positivo de estas señales. La figura 4.10 muestra la estructura del controlador ADC.

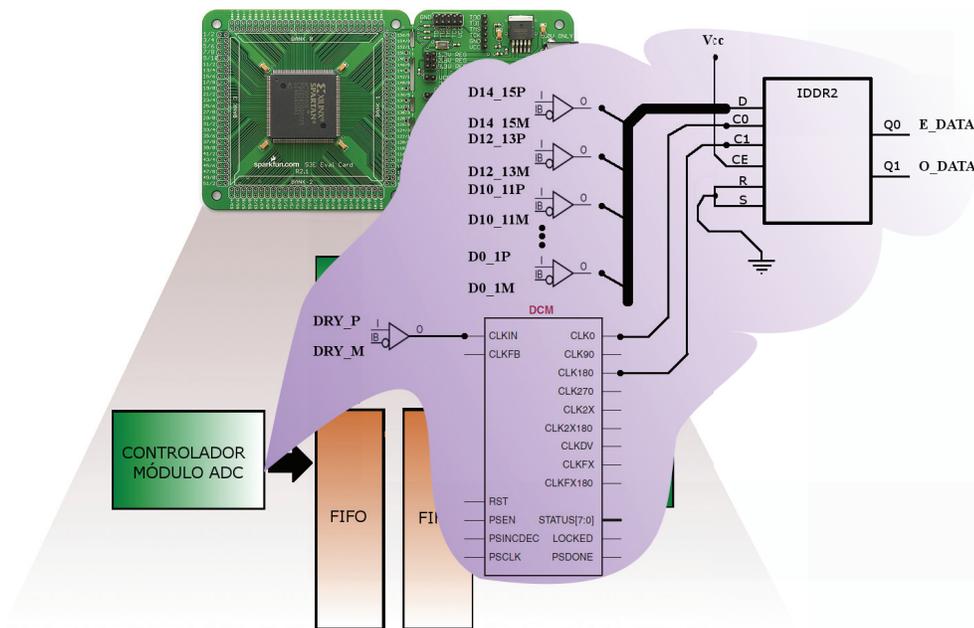


Figura 4.10: Controlador ADC.

### 4.8.3. Controlador DAC

En la figura 4.11 se muestra la máquina de estados que utiliza el controlador DAC para el manejo del conversor D/A.

En esta máquina de estados, el proceso inicia cuando se detecta que el buffer de transmisión se encuentra lleno, en tal caso, se genera la señal que habilita la lectura de datos del buffer para su envío al conversor D/A. Una vez que ya no hay más datos en el buffer, se deshabilita la señal de lectura y se espera por nuevos datos en el buffer. El proceso se repite cada vez que el DSP coloca un bloque de datos en el buffer para su transmisión.

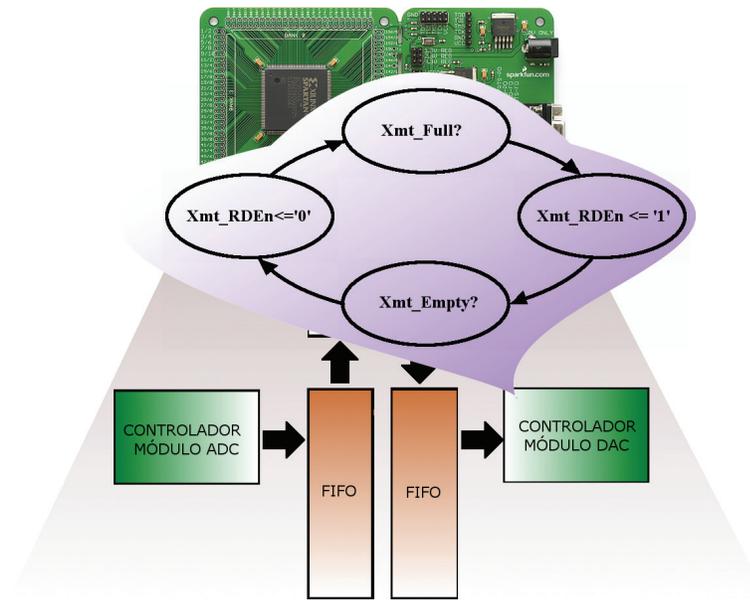


Figura 4.11: Controlador DAC.

#### 4.8.4. FIFO

Las FIFO corresponden a buffers que el módulo de adquisición utiliza para el almacenamiento temporal de bloques de datos que se requieren transferir desde o hacia la memoria del DSP. Estas FIFO son del tipo asíncrona, esto significa que las operaciones de lectura y escritura se realizan en diferentes dominios de reloj. Además, contienen la lógica de control necesaria para el manejo de las operaciones de lectura y escritura, proporcionando indicaciones de estado e indicaciones adicionales para su interfaz.

Las indicaciones disponibles en las FIFO son las siguientes:

- Empty.- Indica que la memoria interna se encuentra vacía y que no hay datos disponibles para las peticiones de lectura.
- Full.- Indica que la memoria ya no tiene espacio para nuevos datos y que cada nueva petición de escritura no será tomada en cuenta.
- Half.Full.- Indica que la memoria interna se encuentra llena a la mitad y que puede seguir recibiendo peticiones de lectura y escritura.

La figura 4.12 muestra la estructura de las FIFO implementadas.

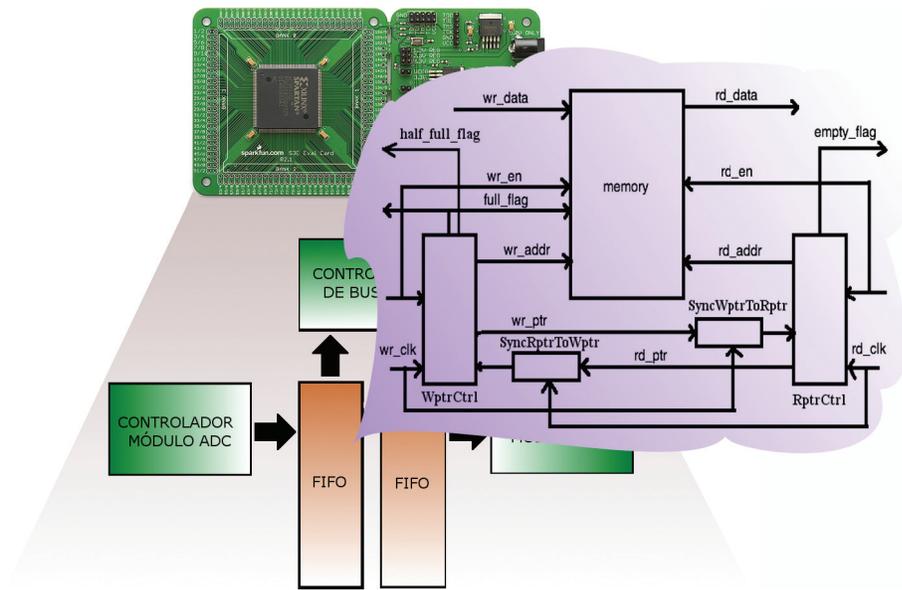


Figura 4.12: Estructura de la FIFO asíncrona.

En el anexo A de este documento se incluye el código VHDL del diseño de la etapa de adquisición. Sin embargo, las pruebas a esta etapa no se adjuntan, debido a problemas de ruido con las conexiones del hardware utilizado. Estos problemas pueden ser corregidos con la construcción en tarjeta de circuito impreso del diseño, lo cual se deja como trabajo futuro.

## 4.9. Resumen del capítulo

En este capítulo se ha presentado la estructura del módem desarrollado, describiendo a grandes rasgos cada una de las etapas que lo componen. Como se pudo notar, son muchos los algoritmos que se requieren para que la señal de información pueda salvar las dificultades impuestas por el medio eléctrico. Además, las características del hardware y software que se requieren exigen cumplir ciertos criterios para su implementación satisfactoria.

# Capítulo 5

## Pruebas y resultados

En este capítulo, se muestran los experimentos realizados y los resultados obtenidos en el desarrollo del módem PLC. En la parte inicial se delinea el procedimiento seguido en cada uno de los experimentos, indicando su objetivo, así como las condiciones y especificaciones en los que se llevan a cabo. La parte restante del capítulo está destinado a mostrar los resultados encontrados en cada experimento.

### 5.0.1. Procedimiento experimental

En la figura 5.1 se muestra el diagrama general de conexiones del sistema utilizado para las pruebas. Como se puede observar, hay dos secciones principales. La primera sección corresponde al uso de una computadora personal con MATLAB y CCS v3.4 para simulación de códigos y despliegue de resultados. Además, la computadora sirve como medio de comunicación con el kit del DSP para envío y recepción de datos utilizando el módulo de intercambio de datos en tiempo real (RTDX). La segunda sección corresponde al kit de desarrollo para el DSP, el cual sirve como herramienta para la experimentación con el sistema OFDM implementado. Cada uno de los experimentos que se muestran en seguida se enfocan en evaluar la etapa de recepción, ya que esta etapa es fundamental para la correcta recuperación de información.

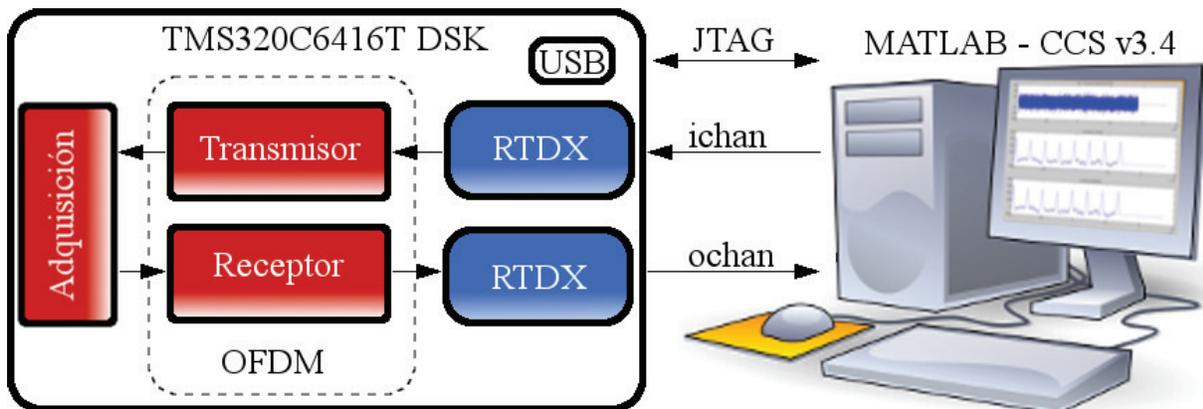


Figura 5.1: Diagrama general de conexiones del sistema.

El objetivo principal de los experimentos es evaluar el desempeño de los algoritmos implementados en cada una de las etapas del receptor del módem. Para esto, se han diseñado los siguientes experimentos:

- Experimento 1. Prueba de los algoritmos recepción en un ambiente ideal, esto es, sin la presencia de ruido ni distorsiones producidas por el canal de comunicación.
- Experimento 2. Prueba del receptor OFDM en presencia de ruido blanco para una relación señal a ruido (SNR) dada.
- Experimento 3. Prueba de los algoritmos de correlación y detección de símbolo piloto para una SNR dada.

Cada uno de los experimentos con el transreceptor fueron realizados fijando los parámetros mostrados en la tabla 5.1.

Tabla 5.1: Parámetros del transreceptor OFDM

Parámetro	Valor
Frecuencia del sistema - $f_s$	48KHz
Subportadoras - $K$	63
Longitud del IG - $L$	19 muestras
FFT - $N$	128
Longitud total del símbolo OFDM - $N + L$	147 muestras
Modulación	16-QAM
Bits por símbolo OFDM	252

El objetivo del experimento 1, es verificar que los algoritmos programados para el proceso de recepción funcionen correctamente. Para llevar a cabo este experimento, se auxilió de las conexiones que se muestran en la figura 5.2.

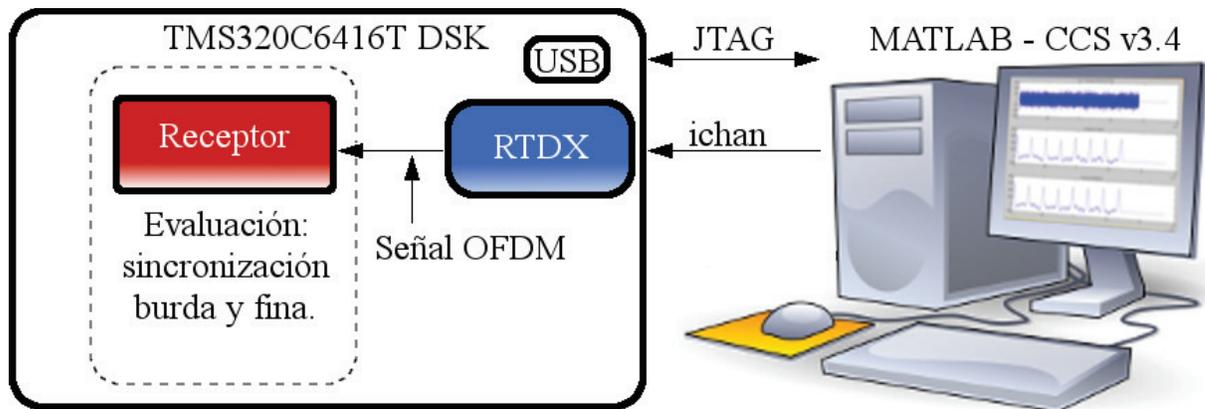


Figura 5.2: Diagrama de conexiones para el experimento 1.

En este experimento, se genera la señal OFDM desde MATLAB, cuya estructura se muestra en seguida

$$[S_0, S_1, S_0, S_1, S_0, S_1, S_0, S_1, \mathbf{S}_A, S_0, S_1, S_0, S_1, S_0, S_1, S_0, S_1, \mathbf{S}_A]$$

donde  $S_A$  es la señal del símbolo piloto,  $S_0$  y  $S_1$  son las señales de símbolos aleatorios.

Esta señal generada desde MATLAB es enviada al DSP por medio del módulo RTDX empleando el canal de entrada *ichan*. Una vez recibida la señal, se le envía al módulo de recepción OFDM, donde se realiza el proceso de sincronización burda y fina. El comportamiento de la sincronización burda y fina se observa tanto en CCS como en MATLAB. En la figura 5.3 se puede ver la señal OFDM utilizada en el experimento 1.

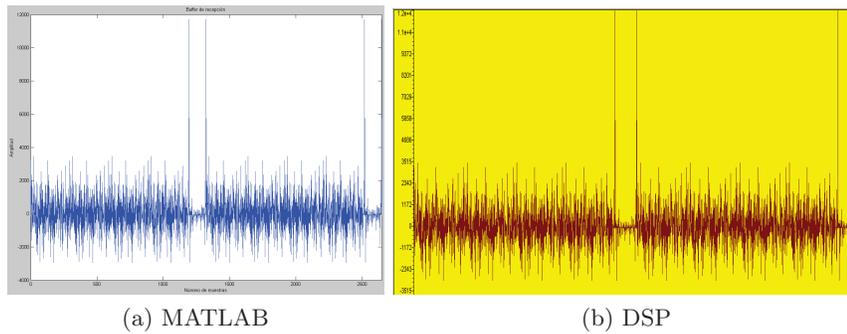


Figura 5.3: Señal OFDM de referencia.

El objetivo del experimento 2, es medir los valores de la relación señal a ruido para los cuales el receptor OFDM es capaz de funcionar. En este experimento se hizo uso de las conexiones mostradas en la figura 5.4. Donde se asume una sincronización perfecta del sistema,

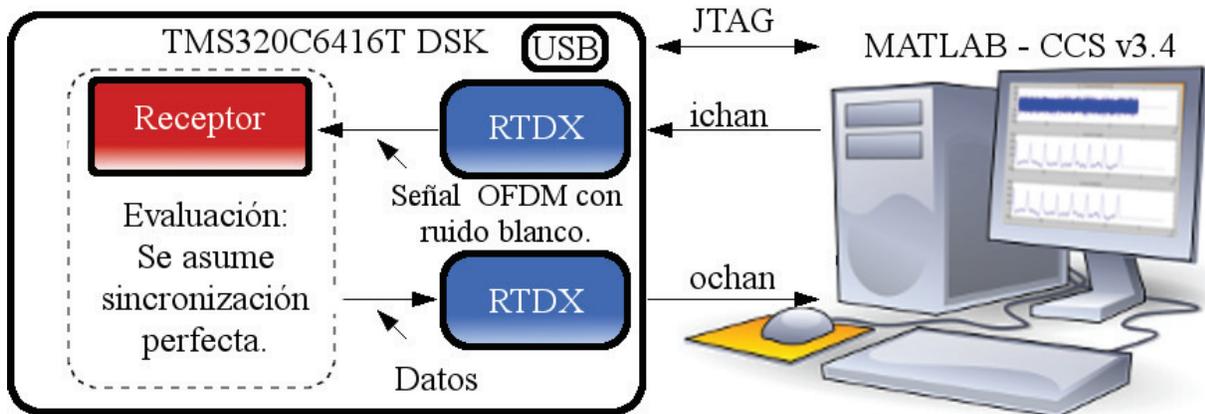


Figura 5.4: Diagrama de conexiones para el experimento 2.

es decir, que el receptor conoce con antelación el inicio de cada trama OFDM recibida. La finalidad es determinar el BER del receptor OFDM, evitando posibles errores introducidos por el proceso de sincronización. La métrica utilizada para medir el BER del sistema es como

sigue:

$$BER = \frac{No. errores}{Bits totales} \quad (5.1)$$

Para este experimento, en MATLAB se generan tramas OFDM con datos de prueba procedentes de una imagen. Cada una de las tramas se envía al DSP por medio del canal *ichan* del módulo RTDX. Posteriormente, se pasa al módulo de recepción para la realización de la demodulación OFDM y el desmapeo de bits. Una vez recuperados los datos de la señal OFDM recibida, éstos se envían hacia MATLAB para el cálculo del BER por medio del canal *ochan* del módulo RTDX.

El objetivo del experimento 3, en primer lugar, es analizar el grado de error que presenta la función de correlación al momento de determinar el inicio de una trama OFDM. En segundo lugar, es observar la variación del error y ganancia que presenta el detector de símbolo piloto. Para el caso de la función de correlación, se desea conocer cuales son los valores máximos de error que se presentan al definir el inicio de cada trama OFDM. Por otra parte, en lo que respecta al detector de símbolo piloto, se desea saber cual es el comportamiento del error y la ganancia asumiendo que la función de correlación detectó correctamente el inicio de cada trama OFDM.

Al igual que para el experimento 2, en este experimento se construyen tramas OFDM con datos de prueba procedentes de una imagen. Sin embargo, en este experimento, cada trama debe pasar por el proceso de sincronización burda y fina. La figura 5.5 muestra las conexiones utilizadas.

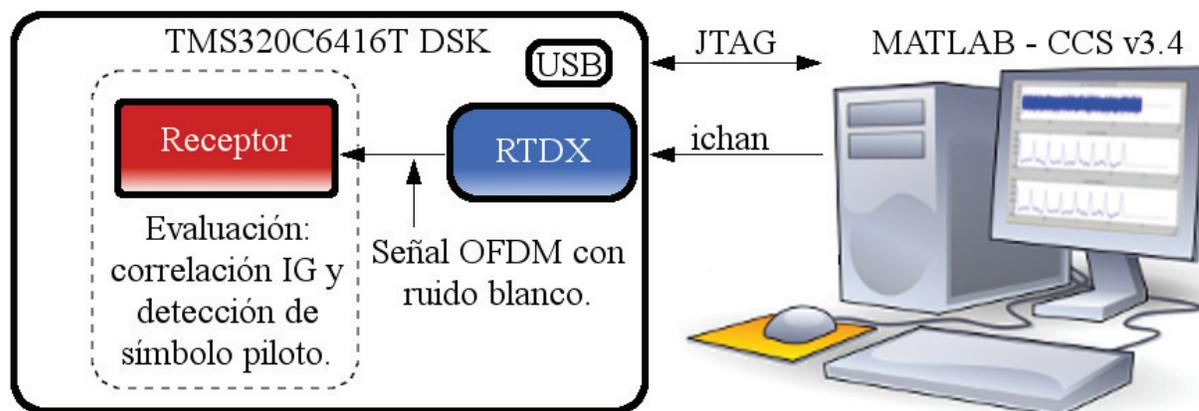


Figura 5.5: Diagrama de conexiones para el experimento 3.

Para los experimentos 2 y 3, se construyeron 38 tramas OFDM con 40 símbolos OFDM cada una, donde del total de símbolos, 8 corresponden al símbolo piloto. La estructura de cada trama se presenta a continuación:

$$[S_A, S_0, S_1, S_2, S_3, S_A, S_4, S_5, S_6, S_7, S_A, \dots, S_k]$$

donde  $S_A$  es la señal del símbolo piloto,  $S_k$  para  $k = 0, 1, \dots, 31$  son los símbolos generados con datos procedentes de una imagen de prueba desde MATLAB. Los experimentos se realizan variando los niveles de  $SNR = \{0, 2, 4, 6, \dots, 20\}$  para el total de tramas. La tabla 5.2 resume las especificaciones de los experimentos 2 y 3.

Tabla 5.2: Parámetros del experimento

Parámetro	Valor
Tramas OFDM	39
Símbolos por trama	32
Símbolos piloto por trama	8
Total de símbolos por trama	40
Total de símbolos del experimento	1560
Bits por símbolo OFDM	252
Bits totales por trama	8064
Bits totales del experimento	314496

En las figuras 5.6 y 5.7 se pueden observar ejemplos de tramas OFDM generadas desde MATLAB y utilizadas en los experimentos 2 y 3. La figura 5.6 corresponde a una trama OFDM sin la adición de ruido, mientras que la figura 5.7 representa una trama OFDM con ruido gaussiano para una SNR de 0 dB.

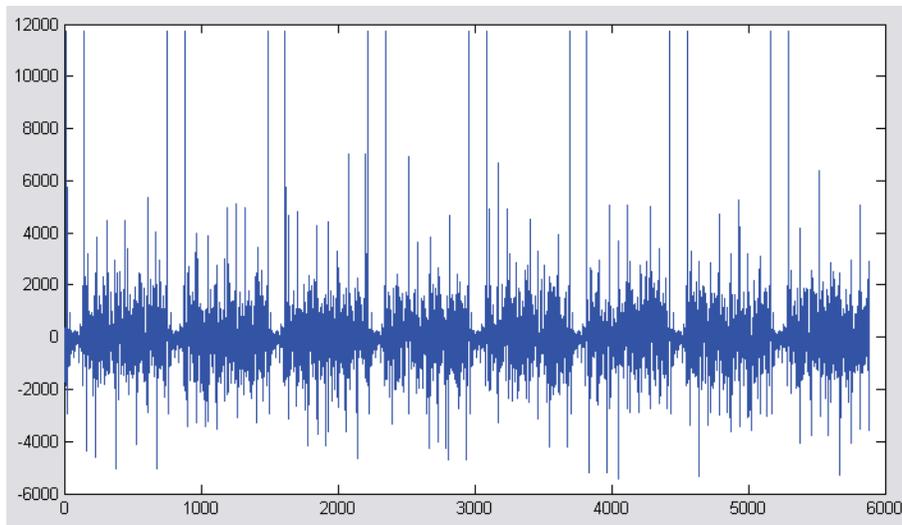


Figura 5.6: Señal OFDM sin ruido.

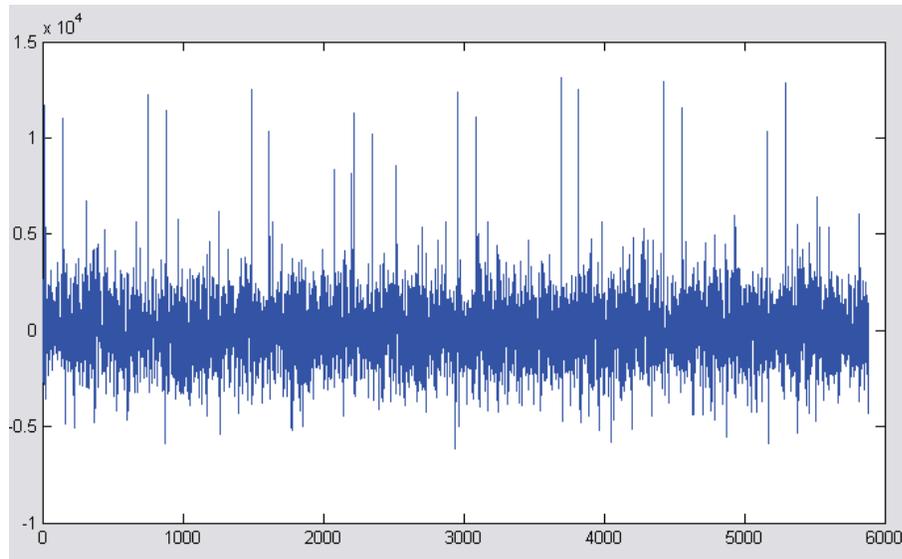


Figura 5.7: Señal OFDM para una SNR de 0 dB.

### 5.0.2. Resultados del experimento 1

En la figura 5.8, se pueden apreciar los resultados de la correlación IG para una ventana de  $2N + L$  muestras. Donde el valor máximo de correlación indica el inicio de un símbolo OFDM. En esta prueba, se eligió arbitrariamente un índice de 47 muestras como inicio para la correlación. Esto significa, de acuerdo a la estructura dada de la señal de prueba, que el inicio del primer símbolo no será detectado, ya que se ha saltado el número de muestras que contienen al intervalo de guarda. Lo anterior fue comprobado cuando el valor máximo de correlación se determinó en el índice de muestra 290, que es aproximadamente el inicio del segundo símbolo OFDM dentro de la señal de prueba.

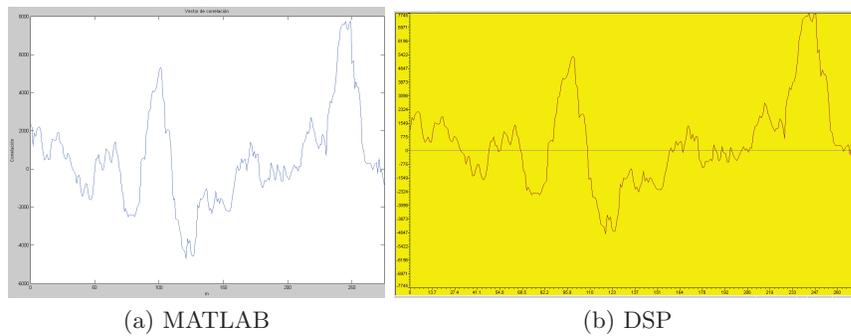


Figura 5.8: Señal de correlación.

La figura 5.9 muestra las gráficas de error y ganancia del detector de símbolo piloto. En estas gráficas, se puede notar claramente cuando fue detectado un símbolo piloto. Debido a que los valores de error y ganancia son menores comparados a los valores de los símbolos aleatorios de la señal de prueba. En cada figura, la gráfica de la parte superior corresponde a los valores de ganancia y la gráfica de la parte inferior a los valores de error.

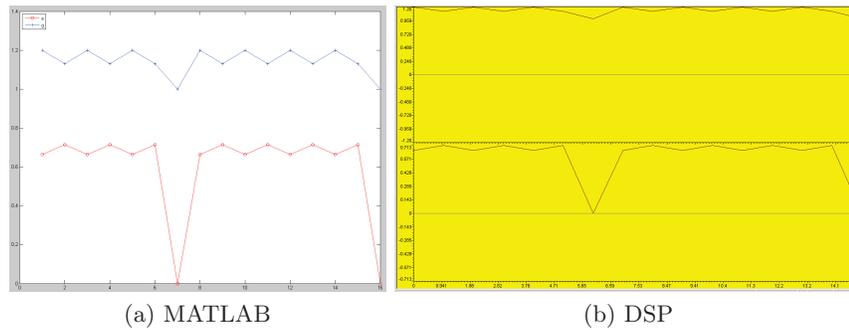


Figura 5.9: Gráfica de error y ganancia para el detector de símbolo piloto.

En las figuras 5.10, 5.11 y 5.12 se puede apreciar la señal real, magnitud y fase, respectivamente, del símbolo piloto sincronizado en forma burda. En cada caso, la señal de la parte superior corresponde a la señal original del símbolo piloto, y la señal de la parte inferior, representa la señal recibida. En la figura 5.10 se puede notar el desplazamiento en tiempo de la señal recibida, a causa de que la correlación IG presenta un error de  $\pm v$  muestras del inicio del símbolo. Por otro lado, se puede advertir, de acuerdo a la figura 5.11 que el desplazamiento en tiempo de la señal recibida no afecta significativamente la magnitud de la señal. No así para el caso de la fase (ver figura 5.12), dado que cualquier desplazamiento en el dominio del tiempo, representa un desplazamiento de fase en el dominio de la frecuencia.

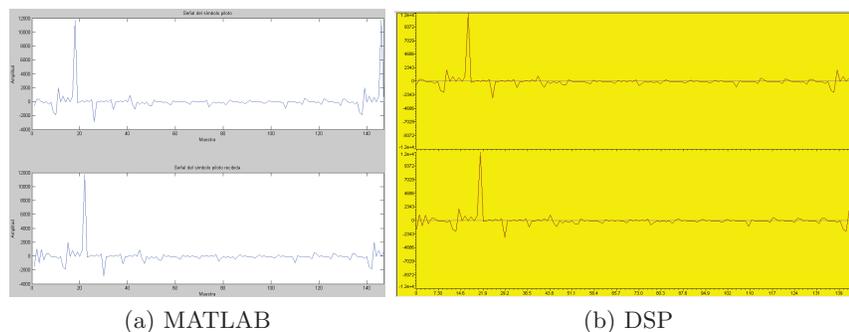


Figura 5.10: Señal del símbolo piloto.

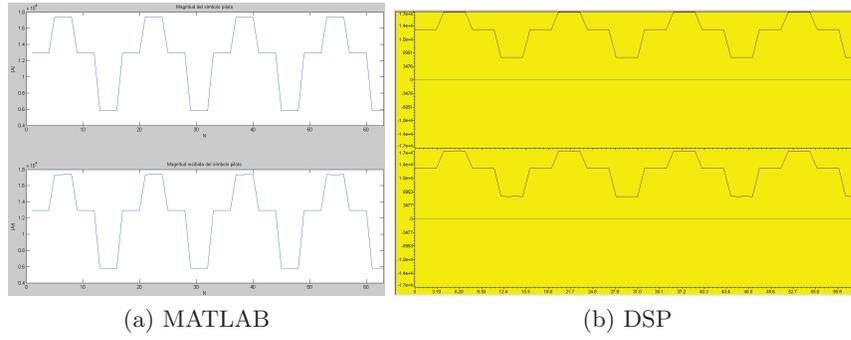


Figura 5.11: Magnitud recibida del símbolo piloto.

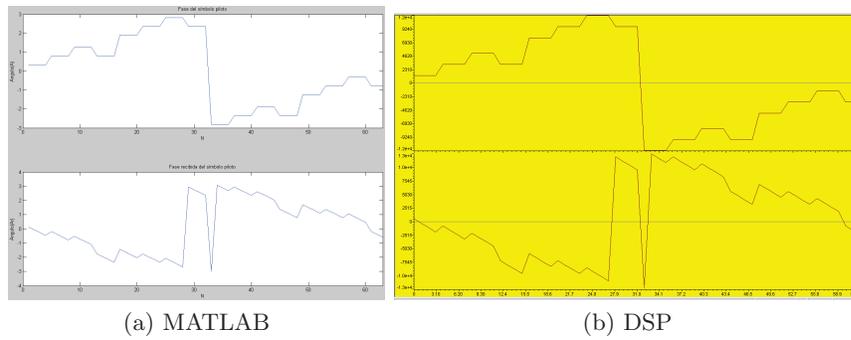


Figura 5.12: Fase recibida del símbolo piloto.

Las figuras 5.13, 5.14 y 5.15 muestran las señales del símbolo piloto después del proceso de sincronización fina. Como se puede observar en la figura 5.15, la fase del símbolo piloto queda corregida con el ajuste fino de la ventana FFT. Esto es, después de aplicar el *estimador de diferencia de ventana*.

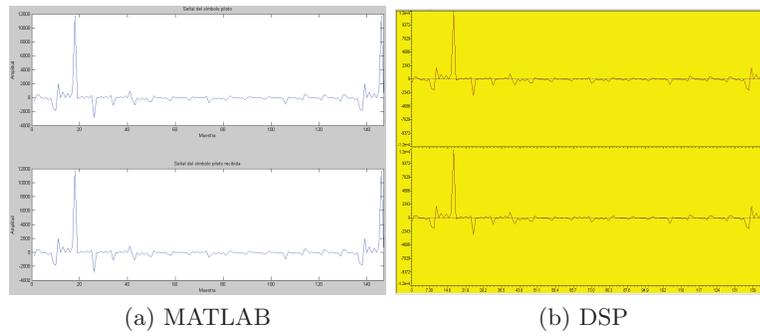


Figura 5.13: Señal del símbolo piloto después de la sincronización.

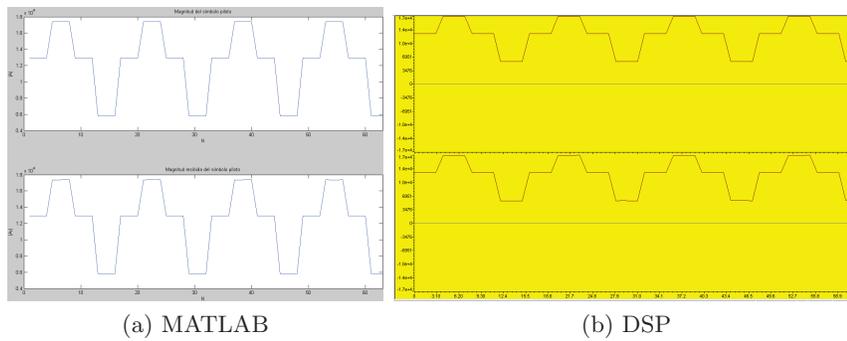


Figura 5.14: Magnitud del símbolo piloto después de la sincronización.

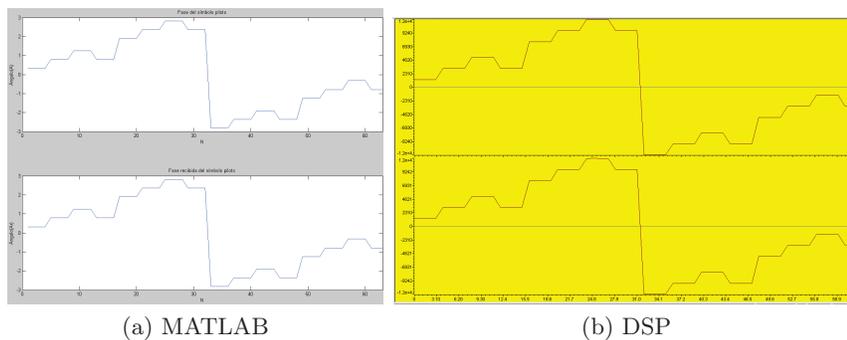


Figura 5.15: Fase del símbolo piloto después de la sincronización.

En la figura 5.16 del lado izquierdo se puede observar el desplazamiento de la constelación QAM debido al desplazamiento en tiempo de la señal recibida, y del lado derecho, la corrección del desplazamiento después del proceso de sincronización fina.

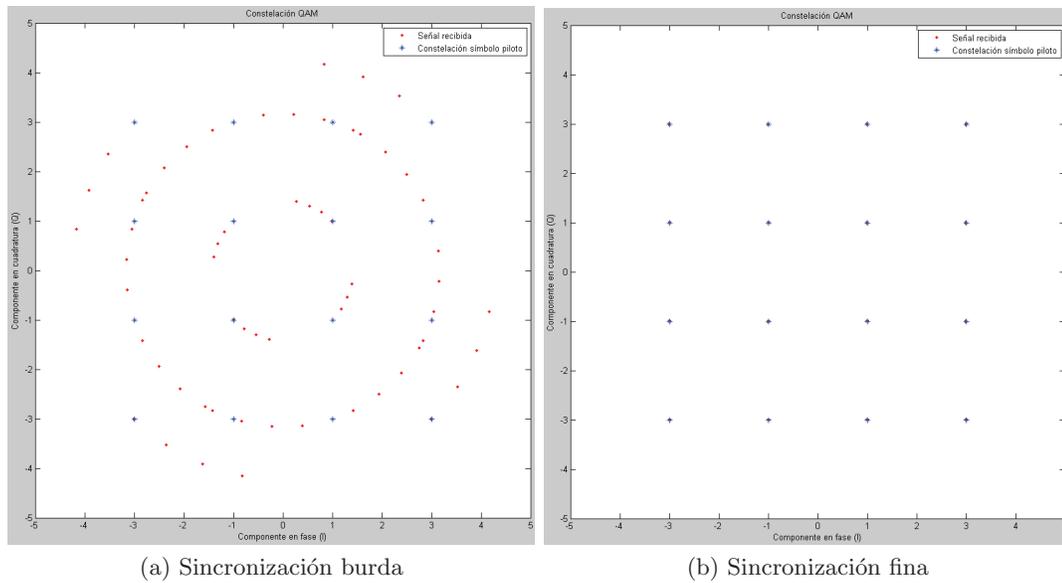


Figura 5.16: Constelación QAM con sincronización burda y fina.

### 5.0.3. Resultados del experimento 2

La figura 5.17 muestra la gráfica de la tasa de error por bit del receptor OFDM para diferentes valores de la relación señal a ruido. Como se puede observar, para tener un bit en error por cada 10 bits transmitidos, se requiere de una relación señal a ruido de 10dB, esto es, se requiere que la potencia de la señal OFDM transmitida sea aproximadamente 10 veces mayor a la potencia de la señal de ruido.

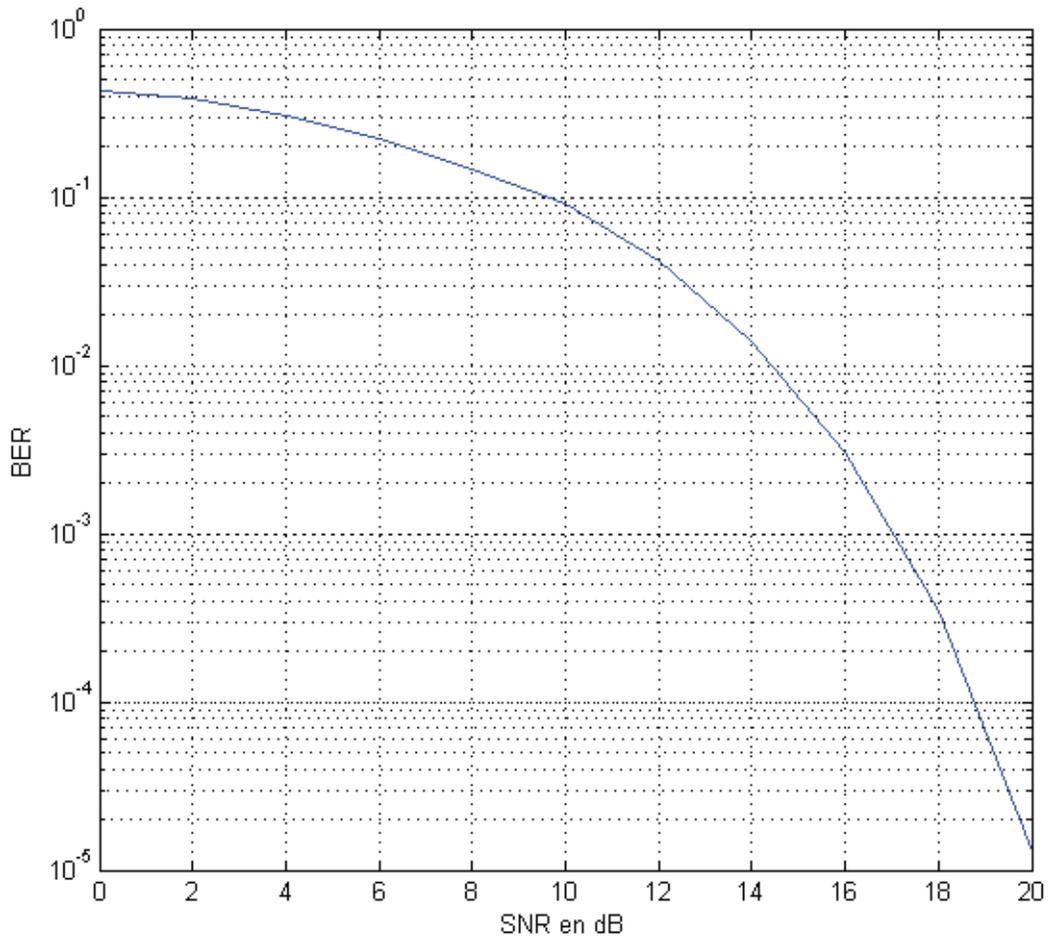


Figura 5.17: Tasa de error por bit para una SNR dada.

### 5.0.4. Resultados del experimento 3

En la figura 5.18 se pueden observar la magnitud del error (muestras), presentado por la función de correlación  $\Psi(\delta_t)$  al momento de determinar el inicio de una trama OFDM. Como se puede apreciar, la magnitud del error se mantiene dentro de la longitud del intervalo de guarda ( $L = 19$ ). Además, la magnitud del error disminuye con relación al incremento de la relación señal a ruido, mateniendose una magnitud de error promedio en 8 muestras a partir de los 4dB, para el rango de la relación señal a ruido evaluada.

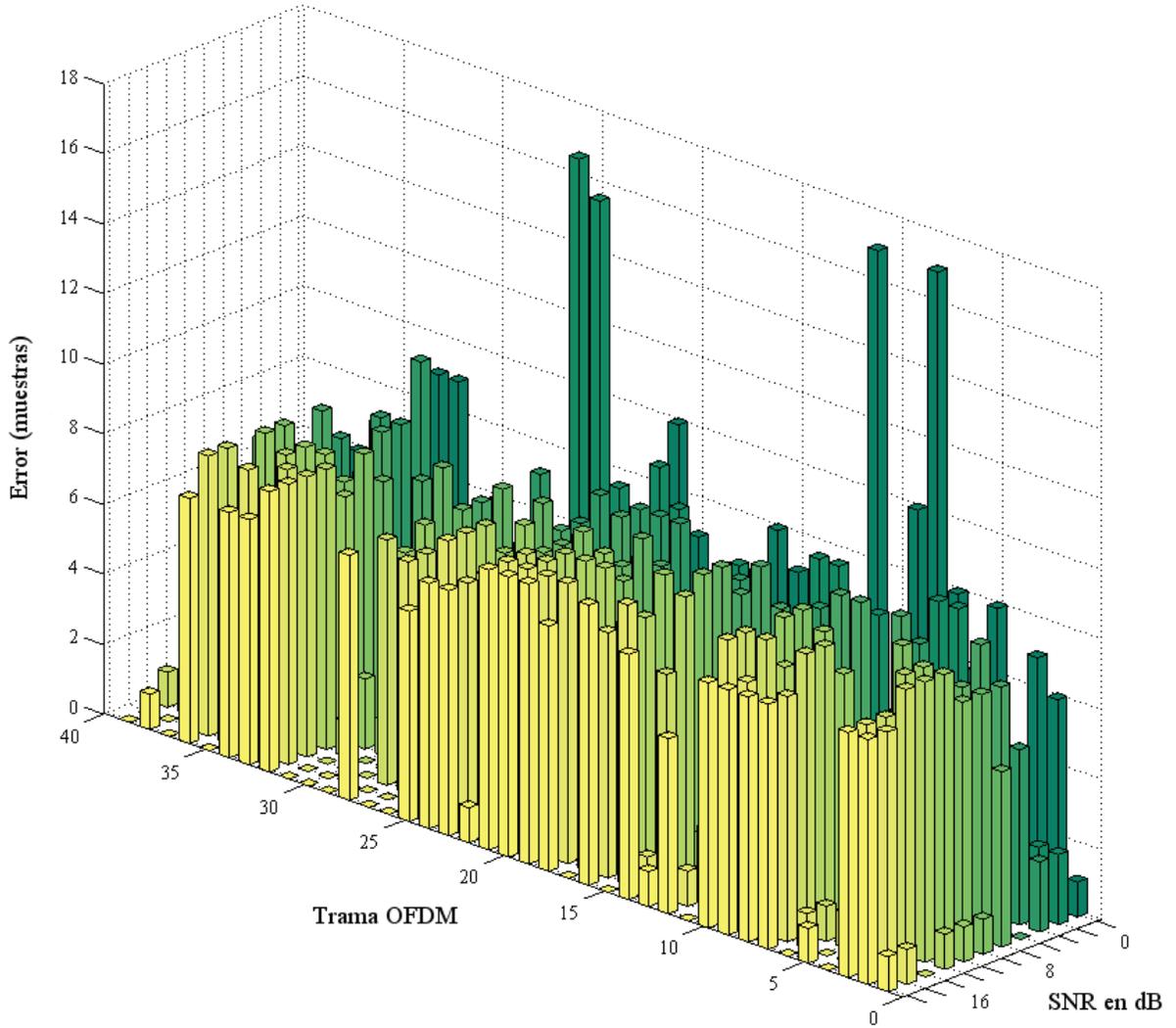


Figura 5.18: Error en muestras de la función de correlación para una SNR dada.

La figura 5.19 muestra el error (muestras), de la función de correlación  $\Psi(\delta_t)$  al determinar el inicio de una trama OFDM. En esta ocasión, se muestra si la función de correlación determinó el inicio de la trama OFDM en un número de muestras antes o después del valor de muestra correcto. Un valor negativo del error indica que la trama fue detectada después del valor de muestra correcto, caso contrario si el error tiene un valor positivo mayor a cero.

Observando la figura 5.18 y la figura 5.19 se puede notar que los valores más altos de error se presentan cuando la función de correlación  $\Psi(\delta_t)$  detecta el inicio de la trama OFDM después del valor correcto de muestra. Tomando en cuenta esta información, se pueden ajustar los valores de umbral que permitan disminuir el nivel de error para la detección del inicio de la trama OFDM.

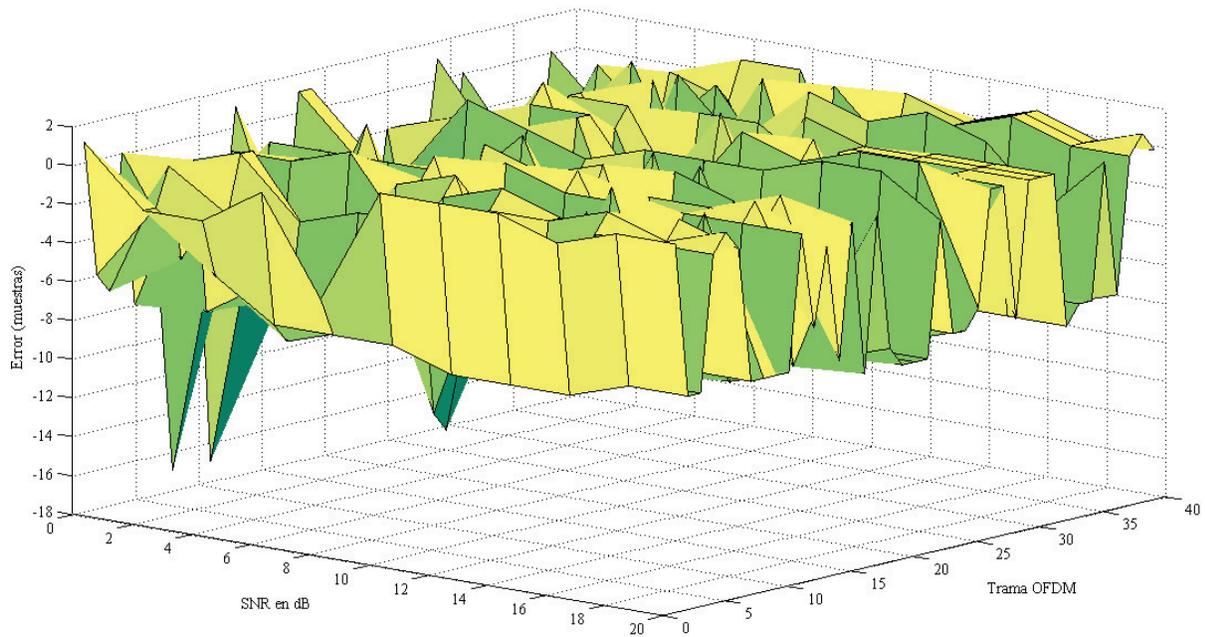


Figura 5.19: Error en muestras de la función de correlación para una SNR dada.

En la figura 5.20 se muestran las gráficas para las funciones de error y ganancia utilizados en la detección del símbolo piloto. En este experimento, el receptor conoce exactamente el inicio de cada uno de los símbolos piloto contenidos en la trama OFDM recibida. Como se podría esperar, los valores de error tienden a cero con el incremento de la relación señal a ruido; por otra parte, los valores de ganancia tienden a uno, lo cual significa que la señal del símbolo piloto recibido se parece cada vez más a la señal conocida por el receptor.

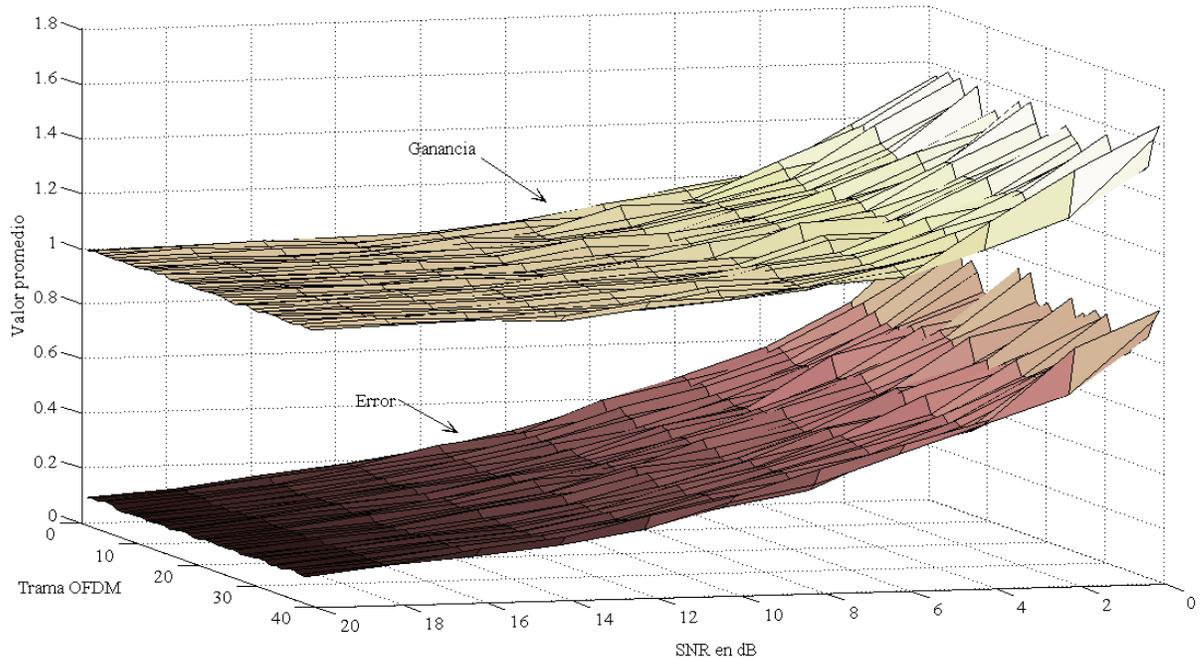


Figura 5.20: Comportamiento del *error* y la *ganancia* para una SNR dada.

## 5.1. Resumen del capítulo

En el capítulo se han presentado los experimentos diseñados para evaluar el funcionamiento de los diferentes algoritmos que componen al receptor para la plataforma del módem PLC. Se ha mostrado como los efectos del desplazamiento en tiempo afectan la magnitud y la fase de los símbolos OFDM. De igual manera, se ha visto el comportamiento de la función de correlación y del detector de símbolo piloto frente a determinado valor de la relación señal a ruido. Asimismo, se ha proporcionado la tasa de error por bit del sistema para diferentes valores de la relación señal a ruido.

## Capítulo 6

# Conclusiones y trabajo futuro

El capítulo reseña las aportaciones y las conclusiones obtenidas durante el desarrollo del trabajo. Inicia señalando las aportaciones obtenidas de acuerdo a los objetivos planteados, continuando con las conclusiones extraídas de los experimentos realizados y finalizando con la mención de los puntos más sobresalientes a atacar para mejorar el diseño planteado.

### 6.1. Conclusiones

En el presente trabajo, se ha realizado la implementación en hardware y software de los algoritmos aptos para un módem que utiliza el medio eléctrico como canal de comunicación. Cada uno de los módulos que aquí fueron presentados, son los más característicos en un diseño utilizando OFDM. Además, la gran mayoría de los fabricantes actuales que basan la capa física de sus dispositivos en OFDM, utilizan variantes de las técnicas que aquí fueron presentadas.

Las aportaciones del trabajo se puede resumir como sigue:

- Una plataforma de hardware y software para la evaluación de algoritmos en un ambiente PLC. En esta plataforma, la parte de software se compone de los códigos del transreceptor OFDM desarrollados en lenguaje C y MATLAB. Los códigos en MATLAB permiten la simulación del comportamiento del transreceptor antes de su programación en hardware. Por otro lado, los códigos en lenguaje C, proveen la funcionalidad del transreceptor para su implementación en el DSP TMS320C6416T de Texas Instruments. No obstante, con ligeros cambios a estos códigos, se puede migrar el transreceptor a cualquier otro dispositivo, debido a que el lenguaje C es en la actualidad, uno de los lenguajes más utilizados en la programación de dispositivos. Por otro parte, el hardware se compone del módulo de adquisición diseñado en el dispositivo FPGA Spartan 3E, el cual, permitiría virtualmente la prueba de algoritmos a frecuencias en el rango de 5Mhz a 30MHz. Que son frecuencias típicas para módems PLC en banda ancha.
- Módulo de adquisición. Este módulo diseñado es el hardware requerido para que el transreceptor pueda trabajar en banda ancha. Lo cual permitiría tasas de transmisión de información del orden de los Megabits (en un rango de 10Mb/s a 200Mb/s).
- Un módem PLC basado en OFDM, a nivel de simulación. La implementación en el DSP TMS320C6416T del transreceptor OFDM que reúne los algoritmos reportados en

la literatura como los más adecuados para trabajar en ambientes PLC, representan al módem PLC basado en OFDM. Lo cual ratifica la hipótesis planteada en este trabajo.

De las pruebas realizadas en la simulación del módem PLC se concluye lo siguiente:

- En el diseño desarrollado, el sistema requiere de una SNR de aproximadamente 10dB para una tasa de error de 1 bit de cada 10 bits transmitidos.
- La detección del *símbolo piloto* juega un papel fundamental en el sistema desarrollado, ya que de ésta, dependen en gran medida las etapas de sincronización fina y de estimación de canal. De los algoritmos implementados para esta etapa, un valor del *error* cercano a cero y un valor de *ganancia* cercano a uno, indican una buena detección del *símbolo piloto*.
- Para la función de correlación utilizada  $\Psi(\delta_t)$  en el proceso de sincronización burda, la información del inicio de un símbolo OFDM queda confinado en un número de muestras no mayor a la longitud del prefijo cíclico.

## 6.2. Trabajo futuro

Como trabajo futuro se resaltan los siguientes puntos:

- Realizar el diseño en tarjeta de circuito impreso del módulo de adquisición, ya que esto permitiría las pruebas en banda ancha.
- Construir la etapa de acoplamiento al canal eléctrico.
- Realizar el diseño del control de un VCO y aplicar el algoritmo de estimación de diferencia de reloj de muestreo.
- Utilizar la opción del DSP BIOS del TMS320C6416T para hacer un diseño basado en tareas del transreceptor OFDM.
- Apegarse al estándar IEEE1901 para comunicaciones en banda ancha.

# Referencias bibliográficas

- [1] ABOLTINS A., KLAVINS D. *Synchronization for OFDM-based Communication System: A Brief Overview*. Riga Technical University 50th International Scientific Conference, 2009.
- [2] BAHAI R.S.A., SALTZBERG R. B. *Multi-Carrier Digital Communications, Theory and Applications of OFDM*. New York: Kluwer Academic Publishers. 2002.
- [3] BASAK K., NIZAMETTIN C., MEHMET S. *A DSP based DMT Modem For Power Line Communications*. International Symposium on Power-line Communications and its Applications. ISPLC. 2001.
- [4] CARDUCCI L., CERVETTO M., PEZOIMBURU P. *Implementación de un transceiver OFDM para BPL*. Universidad de Buenos Aires, Buenos Aires, Argentina. 2008.
- [5] CARLSON A. B., CRILLY P. B., RUTLEDGE J. C. *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*. McGraw-Hill, Fourth Edition, 2002.
- [6] CASTILLE K. *TMS320C6000 EMIF to External FIFO Interface*. Application Report SPRA543. Texas Instruments, May 1999.
- [7] CORTÉS, J. A. *Modulation and Multiple Access Techniques for Indoor Broadband Power-Line Communications*. PhD thesis, Universidad de Málaga, Málaga, España, 2007.
- [8] DICK C., HARRIS F. *FPGA IMPLEMENTATION OF AN OFDM PHY*. SDR 03 Technical Conference and Product Exposition. 2003.
- [9] FAZEL K., KAISER S. *Multi-Carrier and Spread Spectrum Systems*. John Wiley & Sons Inc. 2003.
- [10] FINGER R. *Using TI FIFOs to Interface High-Speed Data Converters With TMS320 DSPs*. Application Report SDMA003. Texas Instruments, June 2001.
- [11] GARCIA H., ALARCON V. *A Power-Line Communication Modem based on OFDM*. International Conference on Electrical, Communications, and Computers. 2009.
- [12] GARCÍA J. *A Survey on BPL Communications Standards*. International Journal of Computer Science and Network Security, Vol. 8 No. 10, October 2008.
- [13] GRANADO J., TORRALBA A., BAENA-LECUYER V., CHÁVEZ J., *Estimadores de desviación de frecuencia y tiempo en OFDM*. Procedente de URSI 2003, Coruña(España), 2003.
- [14] HAI Y., SHENGLI Z., ZHIJIE J. and BAOSHENG L. *A DSP Implementation of OFDM Acoustic Modem*. Montréal, Québec, Canada. 2007.
- [15] JIE M., QIANG G., WU Y., YAO F. *The realization of OFDM Power Line Modem with DSP*. 6th International Conference on Signal Processing. 2002.

- [16] KAMMEYER K-D., SCHMIDT H. *OFDM: An Old Idea Solves New Problems*. International Symposium on Theoretical Electrical Engineering(ISTET01), Linz, Austria, August 2001.
- [17] KARDIN A. *Design and Implementation of OFDM Transmitter And Receiver on FPGA Hardware*. Universiti Teknologi Malaysia. November, 2005.
- [18] KATAYAMA Masaaki. *Introduction to Robust, Reliable, and High-Speed Power-Line Communication System*. IEICE Trans. Fundamentals. Vol.E84-A, No. 12, December 2001.
- [19] KATSIS P. L., PAPADOPOULOS G. D., PAVLIDOU F.-N . *Comparison of Coded Orthogonal Frequency Division Multiplexing and Multicarrier Code Division Multiple Access Systems for Power Line Communications*. International Journal of Communication Systems, Volume 17, Issue 9, November 2004.
- [20] KWON B. *Broadband over Power Lines (BPL): Developments and Policy Issues*. OECD Digital Economy Papers, No. 157, OECD Publishing. June 2009.
- [21] LAGUNA, G. *Aplicación de la transformada de ondeleta para la estimación de canales PLC* Phd thesis, Centro de Investigación en Computación, México, D. F., Junio, 2010.
- [22] LAGUNA, G. *Circuito de acoplamiento para la comunicación por la línea eléctrica doméstica*. Congreso de Ingeniería Eléctrica y Electrónica Aplicada, CIEEA. In 9o. , p.p. 1-4, S.P., México, Marzo 2008.
- [23] LAWREY P. *Adaptive Techniques for Multiuser OFDM* James Cook University. December 2001.
- [24] LI Y., GORDON S. *Frequency Division Multiplexing for Wireless Communications*. Signals and Communications Technology. Springer, 2006.
- [25] LUSHBAUGH L., SAFAVIAN S. R. *Broadband over Power Lines (BPL)*. Bechtel Corporation Technical Journal, Volume 5, Number 1, January 2007.
- [26] MALLELA N. and SHAHNASSER H. *Broadband over Power Lines: Challenges Ahead*. San Francisco State University, CA, 2007.
- [27] MARTÍNEZ A. *Implementación de códigos correctores de errores en un módem para la comunicación por la red eléctrica*. Master thesis, Centro de Investigación en Computación, México, D.F.,2011.
- [28] MATTHIEU CRUSSIÈRE. *Étude et optimisation de communications à haut-débit sur lignes d'énergie: exploitation de la combinaison OFDM/CDMA*. PhD thesis, Institut National des Sciences Appliquées de Rennes (INSA), Rennes, Francia, 2005.
- [29] MEKURIA G., HISRCH H. *Powerline Communication: Untapped Broadband Infrastructure in Developing Countries*. University of Duisburg-Essen, Germany, 2007.
- [30] PAVLIDOU N., VINCK H., YAZDANI J., HONORY B. *Power Line Communications: State of the Art and Future Trends*. IEEE Communications Magazine, April 2003, pp. 34-40.
- [31] PAZ H., CASTELLANOS G. *Noise and Interference in Power Line Channels*. Electronics, Robotics and Automotive Mechanics Conference. 2009.
- [32] PRASAD R. *OFDM for Wireless Communications Systems*. Artech House, Inc. Universal Personal Communications Series. 2004.
- [33] PROAKIS J. G., MANOLAKIS D. G. *Tratamiento digital de señales: principios, algoritmos y aplicaciones*, Tercera edición, Prentice Hall, Madrid, 1998.

- [34] PROAKIS J. G., SALEHI M. *Digital Communications*. McGraw-Hill, Fifth Edition, 2008.
- [35] RASTISLAV R. *THE DESIGN OF A PLC MODEM AND ITS IMPLEMENTATION USING FPGA CIRCUITS*. Journal of ELECTRICAL ENGINEERING, Vol. 60, No. 1, 43-47. 2009.
- [36] ROHLING H. *OFDM : Concepts for Future Communication Systems*. Signals and Communications Technology. Springer, 2011.
- [37] RÓKA, R., DLHÁŇ S. *Modelling Of Transmission Channels Over The Low-Voltage Power Distribution Network*. Journal of Electrical Engineering, Vol. 56, No. 9-10, 2005, 1-9.
- [38] SALMAN M., EL-SHAFEI M. *Power Line Communications: An Overview of Applications, Technology and the HomePlug Standards*. King Fahd University of Petroleum and Minerals, Dhahran, KSA.
- [39] SÁNCHEZ M., CORTÉS J., DÍEZ L., CAÑETE F., TORRES L. *Performance analysis of OFDM modulation on indoor PLC channels in the frequency band up to 210 MHz*. International Symposium on Power-Line Communications and its Applications (ISPLC), 2010.
- [40] SARTENAER T. *Multiuser communications over frequency selective wired channels and applications to the powerline access network*. PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgique, 2004.
- [41] SIMOES L., GERALD J. A. B. *A Communication System for Power Lines*. CHIPIDEA - Microelectrónica, S.A. Porto Salvo, Portugal.
- [42] SOTIRIS H., PANTELIS D., EVAGELOS K. and ATHANASE A. *Design and Implementation of OFDM Transceiver Framework Using MATLAB and the TMS320C6713 DSK*. EUROCON, Serbia Montenegro, Belgrade. 2005.
- [43] SOUHA S., ARWA B., FETHI T., CHIHEB R. *OFDM modem design and implementation for narrowband Powerline Communications*. International Conference on Design & Technology of Integrated Systems in Nanoscale Era. 2010.
- [44] SPRITE R. L. *The Future of Broadband Over Power Line*. Department of Electrical Engineering, Volume 4, Issue 2, Spring 2006.
- [45] SUTTERLIN P., DOWNEYA W. *Power Line Communication Tutorial-Challenges and Technologies*. Echelon Corporation. Palo Alto, California, USA.
- [46] TROYA A. L. *Synchronization and Channel Estimation in OFDM Systems: Algorithms for Efficient Implementation of WLAN Systems*. PhD thesis, IHP(Institute for High Performance microelectronics), Frankfurt, Germany, 2004.
- [47] VAN D., VAN L., VAN X., HUY D., HUU T. and TIEN H. *Implementation of an OFDM system based on the TMS320C6416 DSP*. The International Conference on Advanced Technologies for Communications. 2009.
- [48] WEINSTEIN S. B. *THE HISTORY OF ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING*. IEEE Communications Magazine, November 2009, pp 26-35.
- [49] YANG Y., LAMBERT F., DIVAN D. *A Survey on Technologies for Implementing Sensor Networks for Power Delivery Systems*. School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA.
- [50] YUAN N. *An Equalization Technique for High Rate OFDM Systems*. Masters Thesis, University of Saskatchewan, Canada, 2003.

[51] ZIMMERMAN M. AND DOSTERT K. *A Multipath Model for the Powerline Channel*. IEEE Transactions on Communications, Vol. 50, No. 4, April 2002.

[52] Sitio oficial del estándar IEEE1901 <http://grouper.ieee.org/groups/1901/>

# Anexo

# Anexo A

## Código VHDL de la etapa de adquisición

### A.1. Módulo principal

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ADQMOD is
  port(
    --Sistema
    Reset      : in std_logic;
    Clk        : in std_logic;

    --Dsp
    DspData    : inout std_logic_vector(15 downto 0);
    DspAddr    : in std_logic_vector(3 downto 0);
    DspCE2     : in std_logic;
    DspARE     : in std_logic;
    DspAWE     : in std_logic;
    DspAOE     : in std_logic;
    DspInt0    : out std_logic; --Interrupción Rx

    --Adc
    Clk2ADC    : out std_logic;
    Dx_y_P    : in std_logic_vector(7 downto 0);
    Dx_y_M    : in std_logic_vector(7 downto 0);
    Dry_P     : in std_logic;
    Dry_M     : in std_logic;

    --Dac
    Clk2DAC    : out std_logic;
    Data2DAC   : out std_logic_vector(13 downto 0);

    --FIFO 0 (Rx)
    FOEmpty    : out std_logic;
    FOFull     : out std_logic;
    FOHalf_Full : out std_logic;

    --FIFO 1 (Tx)
    F1Empty    : out std_logic;
    F1Full     : out std_logic;
  );
end ADQMOD;
```

```

architecture Structural of ADQMOD is
    constant DATA_WIDTH : integer:=16;
    constant ADDR_WIDTH  : integer:=12;

    --Sistema
    signal S_SysReset      : std_logic;
    signal S_OutSel       : std_logic_vector(1 downto 0):=B"00";

    --Dsp
    signal S_DspDataOut   : std_logic_vector(DATA_WIDTH-1 downto 0):=X"0000";
    signal S_DspDataIn   : std_logic_vector(DATA_WIDTH-1 downto 0):=X"0000";

    --Generación de señales de reloj
    signal S_Clk0         : std_logic;
    signal S_Clk180      : std_logic;
    signal S_Clk4X       : std_logic;
    signal S_ClkRdyDs    : std_logic;
    signal S_ClkRdy      : std_logic;
    signal S_ClkRdy180   : std_logic;
    signal S_ClkRdy270   : std_logic;
    signal S_DCM1Locked  : std_logic;
    signal S_DCM0Locked  : std_logic;

    -- Decodificador
    signal S_WClk        : std_logic:= '0';
    signal S_RClk        : std_logic:= '0';
    signal S_RDEN        : std_logic:= '0';
    signal S_WREN        : std_logic:= '0';
    signal S_OputEn      : std_logic:= '1';
    signal S_F0Select    : std_logic:= '0';
    signal S_F1Select    : std_logic:= '0';
    signal S_CmdSelect   : std_logic:= '0';

    -- DAC
    signal S_ADCCData    : std_logic_vector(DATA_WIDTH-1 downto 0);
    signal S_Clk2DAC     : std_logic;
    signal S_Xmt         : std_logic;

    -- FIFO 0
    signal S_F0WClk      : std_logic:= '0';
    signal S_FORClk     : std_logic:= '0';
    signal S_FORDEnable  : std_logic:= '0';
    signal S_F0WREnable  : std_logic:= '0';
    signal S_F0Empty     : std_logic:= '0';
    signal S_F0Full      : std_logic:= '0';
    signal S_F0Half_Full : std_logic:= '0';
    signal S_F0DataOut   : std_logic_vector(DATA_WIDTH-1 downto 0);
    signal S_F0DataIn   : std_logic_vector(DATA_WIDTH-1 downto 0);

    -- FIFO 1
    signal S_F1WClk      : std_logic:= '0';
    signal S_F1RClk     : std_logic:= '0';
    signal S_F1RDEnable  : std_logic:= '0';
    signal S_F1WREnable  : std_logic:= '0';
    signal S_F1Empty     : std_logic:= '0';
    signal S_F1Full      : std_logic:= '0';
    signal S_F1Half_Full : std_logic:= '0';
    signal S_F1DataOut   : std_logic_vector(DATA_WIDTH-1 downto 0);
    signal S_F1DataIn   : std_logic_vector(DATA_WIDTH-1 downto 0);

    -- Comandos
    signal S_CmdReg      : std_logic_vector(DATA_WIDTH-1 downto 0):= X"0000";
    signal S_SoftReset   : std_logic;

```

```

-- Componentes
component Decoder is
port(
  EA      : in std_logic_vector(3 downto 0);
  CE2     : in std_logic;
  AWE     : in std_logic;
  ARE     : in std_logic;
  AOE     : in std_logic;
  WClk    : out std_logic;
  RClk    : out std_logic;
  RDEn    : out std_logic;
  WREn    : out std_logic;
  OputEn  : out std_logic;
  FOSelect : out std_logic;
  F1Select : out std_logic;
  CmdSelect : out std_logic
);
end component;

component Timing is
port(
  Reset    : in std_logic;
  Clk      : in std_logic;
  Dry_P    : in std_logic;
  Dry_M    : in std_logic;
  Clk0     : out std_logic;
  Clk180   : out std_logic;
  Clk4X    : out std_logic;
  ClkRdyDs : out std_logic;
  ClkRdy   : out std_logic;
  ClkRdy180 : out std_logic;
  ClkRdy270 : out std_logic;
  DCM0Locked : out std_logic;
  DCM1Locked : out std_logic
);
end component;

component ADC is
port(
  Reset    : in std_logic;
  ClkRdy   : in std_logic;
  ClkRdy180 : in std_logic;
  Dx_y_P   : in std_logic_vector(7 downto 0);
  Dx_y_M   : in std_logic_vector(7 downto 0);
  Data     : out std_logic_vector(DATA_WIDTH-1 downto 0)
);
end component;

component aFIFO is
generic
(
  DATA_WIDTH :integer := 16;
  ADDR_WIDTH  :integer := 12
);
port(
  Reset      : in std_logic;
  WClk       : in std_logic;
  RClk       : in std_logic;
  RDataPort  : out std_logic_vector(DATA_WIDTH-1 downto 0);
  WDataPort  : in std_logic_vector(DATA_WIDTH-1 downto 0);
  WR_En      : in std_logic;
  RD_En      : in std_logic;
  Empty      : out std_logic;
  Full       : out std_logic;

```

```

    Half_Full    : out std_logic
);
end component;

component DAC is
port(
    Reset       : in std_logic;
    Clk4X       : in std_logic;
    Xmt         : in std_logic;
    Empty       : in std_logic;
    RDEnable    : out std_logic
);
end component;

begin
-- Decodificador
SysDecoder : Decoder
port map(
    EA          => DspAddr,
    CE2         => DspCE2,
    AWE         => DspAWE,
    ARE         => DspARE,
    AOE         => DspAOE,
    WClk        => S_WClk,
    RClk        => S_RClk,
    RDEn        => S_RDEn,
    WREn        => S_WREn,
    OputEn      => S_OputEn,
    F0Select    => S_F0Select,
    F1Select    => S_F1Select,
    CmdSelect   => S_CmdSelect
);

-- Generador de reloj
SysTiming : Timing
port map(
    Reset       => S_SysReset,
    Clk         => Clk,
    Dry_P       => Dry_P,
    Dry_M       => Dry_M,
    Clk0        => S_Clk0,
    Clk180     => S_Clk180,
    Clk4X       => S_Clk4X,
    ClkRdyDs   => S_ClkRdyDs,
    ClkRdy      => S_ClkRdy,
    ClkRdy180  => S_ClkRdy180,
    ClkRdy270  => S_ClkRdy270,
    DCMOLocked => S_DCMOLocked,
    DCM1Locked => S_DCM1Locked
);

-- Controlador del ADC
SysADC: ADC
port map(
    Reset       => S_SysReset,
    ClkRdy      => S_ClkRdy,
    ClkRdy180  => S_ClkRdy180,
    Dx_y_P     => Dx_y_P,
    Dx_y_M     => Dx_y_M,
    Data       => S_ADData
);

--FIFO Asíncrona de puerto dual
SysAFiFo1: aFiFo
generic map

```

```

(
  DATA_WIDTH => DATA_WIDTH,
  ADDR_WIDTH => ADDR_WIDTH
)
port map(
  Reset      => S_SysReset,
  WClk       => S_FOWClk,
  RClk       => S_FORClk,
  RDataPort  => S_F0DataOut,
  WDataPort  => S_F0DataIn,--S_ADCCData,
  WR_En      => S_FOWREnable,
  RD_En      => S_FORDEnable,
  Empty      => S_F0Empty,
  Full       => S_F0Full,
  Half_Full  => S_F0Half_Full
);

-- FIFO Asíncrona de puerto dual
SysAFiFo2: aFiFo
generic map
(
  DATA_WIDTH => DATA_WIDTH,
  ADDR_WIDTH => ADDR_WIDTH
)
port map(
  Reset      => S_SysReset,
  WClk       => S_F1WClk,
  RClk       => S_F1RClk,
  RDataPort  => S_F1DataOut,
  WDataPort  => S_F1DataIn,
  WR_En      => S_F1WREnable,
  RD_En      => S_F1RDEnable,
  Empty      => S_F1Empty,
  Full       => S_F1Full,
  Half_Full  => S_F1Half_Full
);

-- Controlador del DAC
SysDAC: DAC
port map
(
  Reset      => Reset,
  Clk4X      => S_Clk4X,--S_Clk2DAC,
  Xmt        => S_Xmt,
  Empty      => S_F1Empty,
  RDEnable   => S_F1RDEnable
);

-- Manejo de indicación al DAC
S_Xmt <= S_F1Full or S_F1Half_Full;

-- Manejo de la señal de reset
S_SysReset <= Reset or S_SoftReset;

-- Manejo del bus bidireccional
S_DspDataIn <= DspData;

with S_OputEn select
DspData <= S_DspDataOut when '0',
          (others => 'Z') when others;

-- Selección de la fuente de datos al DSP
S_OutSel <= S_F0Select & S_CmdSelect;

with S_OutSel select

```

```

S_DspDataOut <= S_F0DataOut when "10",
               S_CmdReg when "01",
               (others=>'Z') when others;

-- Señal de interrupción al DSP
DspInt0 <= S_F0Half_Full when S_DCM1Locked = '1' else 'Z';

-- Señales de la FIFO 0
S_FORDEnable <= S_RDEn and S_F0Select;
S_F0WREnable <= S_DCM1Locked;
S_F0WClk <= S_ClkRdy270;
S_FORClk <= S_RC1k;
S_F0DataIn <= S_ADCCData;
FOEmpty <= S_F0Empty;
FOFull <= S_F0Full;
FOHalf_Full <= S_F0Half_Full;

-- Señales de la FIFO 1
S_F1WREnable <= S_WREn and S_F1Select;
S_F1WClk <= S_WClk;
S_F1RClk <= S_Clk180;
S_F1DataIn <= S_DspDataIn when (S_F1Select = '1') else (others=>'Z');
F1Empty <= S_F1Empty;
F1Full <= S_F1Full;

-- Señal de reloj para el ADC
Clk2ADC <= S_Clk0;

-- Señal de reloj para el DAC
Clk2DAC <= S_Clk4X;
Data2DAC <= S_F1DataOut(DATA_WIDTH-3 downto 0);

-- Recepción de comando
process(S_WClk)
begin
    if(rising_edge(S_WClk))then
        if(S_WREn = '1' and S_CmdSelect = '1')then
            S_CmdReg <= S_DspDataIn;
        end if;
    end if;
end process;

-- Comando de reset del sistema
S_SoftReset <= S_CmdReg(0);

end Structural;

```

## A.2. Módulo del decodificador

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Decoder is
    Port(
        EA      : in std_logic_vector(3 downto 0);
        CE2     : in std_logic;
        AWE     : in std_logic;
        ARE     : in std_logic;
        AOE     : in std_logic;
        WClk    : out std_logic;
        RClk    : out std_logic;
    );
end Decoder;

```

```

        RDEn      : out std_logic;
        WREn      : out std_logic;
        OputEn    : out std_logic;
        F0Select  : out std_logic;
        F1Select  : out std_logic;
        CmdSelect : out std_logic
    );
end Decoder;

architecture Behavioral of Decoder is
begin

    --ARE, AWE y CE son activos en bajo

    -- Habilitación de escritura
    WREn    <= not CE2;

    -- Habilitación de lectura
    RDEn    <= AOE nand CE2;

    -- Señal de reloj para escritura
    WClk    <= AWE;

    -- Señal de reloj para lectura
    RClk    <= not ARE;

    -- Habilitación del bus bidireccional para salida
    OputEn  <= AOE;

    -- Selección de FIFO 0
    F0Select <= '1' when EA(3 downto 0) = "1000" else '0'; -- 0xA0100000

    -- Selección de FIFO 1
    F1Select <= '1' when EA(3 downto 0) = "1001" else '0'; -- 0xA0120000

    -- Selección de comando
    CmdSelect <= '1' when EA(3 downto 0) = "1010" else '0'; -- 0xA0140000

end Behavioral;

```

### A.3. Módulo de generación de las señales de reloj

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

library UNISIM;
use UNISIM.VComponents.all;

entity Timing is
    port(
        Reset      : in  std_logic;
        Clk        : in  std_logic;
        Dry_P      : in  std_logic;
        Dry_M      : in  std_logic;
        Clk0       : out std_logic;
        Clk180    : out std_logic;
        Clk4X     : out std_logic;
        ClkRdyDs  : out std_logic;
        ClkRdy    : out std_logic;
        ClkRdy180 : out std_logic;
        ClkRdy270 : out std_logic;
        DCMOLocked : out std_logic;
    );
end Timing;

```

```

    DCM1Locked : out std_logic
);
end Timing;

architecture Behavioral of Timing is

    -- Señales de salida
    signal S_Clk0      : std_logic;
    signal S_Clk180   : std_logic;
    signal S_Clk4X    : std_logic;
    signal S_ClkRdyDs : std_logic;
    signal S_ClkRdy   : std_logic;
    signal S_ClkRdy180 : std_logic;
    signal S_ClkRdy270 : std_logic;

    -- Buffers globales
    signal S_Clk0Buf   : std_logic;
    signal S_Clk180Buf : std_logic;
    signal S_Clk4XBuf  : std_logic;
    signal S_ClkRdyBuf : std_logic;
    signal S_ClkRdy180Buf : std_logic;
    signal S_ClkRdy270Buf : std_logic;

    -- Señal de Locked del DCM
    signal S_DCM0Locked : std_logic;
    signal S_DCM1Locked : std_logic;
    signal S_DCM0LockedInv : std_logic;

    -- Para la lógica de reset
    signal FDS_Q_OUT   : std_logic;
    signal FD1_Q_OUT   : std_logic;
    signal FD2_Q_OUT   : std_logic;
    signal FD3_Q_OUT   : std_logic;
    signal OR3_0_OUT   : std_logic;
    signal DCM0Reset   : std_logic;
    signal DCM1Reset   : std_logic;

begin

    ClkRdyDs    <= S_ClkRdyDs;
    ClkRdy      <= S_ClkRdyBuf;
    ClkRdy180   <= S_ClkRdy180Buf;
    ClkRdy270   <= S_ClkRdy270Buf;
    Clk0        <= S_Clk0Buf;
    Clk180      <= S_Clk180Buf;
    Clk4X       <= S_Clk4XBuf;

    DCM0Locked   <= S_DCM0Locked;
    DCM1Locked   <= S_DCM1Locked;
    S_DCM0LockedInv <= not S_DCM0Locked;
    DCM0Reset    <= Reset;

    -- Definición de buffers
    CLK0_BUF_INST: BUFG
    port map(I => S_Clk0, 0 => S_Clk0Buf);

    CLK180_BUF_INST: BUFG
    port map(I => S_Clk180, 0 => S_Clk180Buf);

    CLK4X_BUF_INST: BUFG
    port map(I => S_Clk4X, 0 => S_Clk4XBuf);

    IBUFGDS_INST : IBUFGDS
    generic map (DIFF_TERM => TRUE, IBUF_DELAY_VALUE => "0", IOSTANDARD => "LVDS_25")
    port map(I => Dry_P, IB => Dry_M, 0 => S_ClkRdyDs);

```

```

CLKRDY_BUF_INST: BUFG
port map(I => S_ClkRdy, 0 => S_ClkRdyBuf);

CLKRDY180_BUF_INST: BUFG
port map(I => S_ClkRdy180, 0 => S_ClkRdy180Buf);

CLKRDY270_BUF_INST: BUFG
port map(I => S_ClkRdy270, 0 => S_ClkRdy270Buf);

-- Manejadores de reloj
DCM_SP_INST0 : DCM_SP
generic map(
  CLK_FEEDBACK          => "1X",
  CLKDV_DIVIDE          => 2.0,
  CLKFX_DIVIDE          => 1,
  CLKFX_MULTIPLY        => 4,
  CLKIN_DIVIDE_BY_2    => FALSE,
  CLKIN_PERIOD          => 166.667,--200.000,
  CLKOUT_PHASE_SHIFT   => "NONE",
  DESKEW_ADJUST        => "SOURCE_SYNCHRONOUS",
  DFS_FREQUENCY_MODE   => "LOW",
  DLL_FREQUENCY_MODE   => "LOW",
  DUTY_CYCLE_CORRECTION => TRUE,
  FACTORY_JF           => x"C080",
  PHASE_SHIFT          => 0,
  STARTUP_WAIT         => FALSE)
port map(
  CLKFB   => S_Clk0Buf,
  CLKIN   => Clk,
  DSSSEN  => '0',
  PSCLK   => '0',
  PSEN    => '0',
  PSINCDEC => '0',
  RST     => DCM0Reset,
  CLKDV   => open,
  CLKFX   => S_Clk4X,
  CLKFX180 => open,
  CLK0    => S_Clk0,
  CLK2X   => open,
  CLK2X180 => open,
  CLK90   => open,
  CLK180  => S_Clk180,
  CLK270  => open,
  LOCKED  => S_DCM0Locked,
  PSDONE  => open,
  STATUS  => open
);

DCM_SP_INST1 : DCM_SP
generic map(
  CLK_FEEDBACK          => "1X",
  CLKDV_DIVIDE          => 2.0,
  CLKFX_DIVIDE          => 1,
  CLKFX_MULTIPLY        => 4,
  CLKIN_DIVIDE_BY_2    => FALSE,
  CLKIN_PERIOD          => 166.667,
  CLKOUT_PHASE_SHIFT   => "NONE",
  DESKEW_ADJUST        => "SOURCE_SYNCHRONOUS",
  DFS_FREQUENCY_MODE   => "LOW",
  DLL_FREQUENCY_MODE   => "LOW",
  DUTY_CYCLE_CORRECTION => TRUE,
  FACTORY_JF           => x"C080",
  PHASE_SHIFT          => 0,
  STARTUP_WAIT         => FALSE)

```

```

port map(
  CLKFB  => S_ClkRdyBuf,
  CLKIN  => S_ClkRdyDs,
  DSSEN  => '0',
  PSCLK  => '0',
  PSEN   => '0',
  PSINCDEC => '0',
  RST    => DCM1Reset,
  CLKDV  => open,
  CLKFX  => open,
  CLKFX180 => open,
  CLKO   => S_ClkRdy,
  CLK2X  => open,
  CLK2X180 => open,
  CLK90  => open,
  CLK180 => S_ClkRdy180,
  CLK270 => S_ClkRdy270,
  LOCKED => S_DCM1Locked,
  PSDONE => open,
  STATUS => open
);

```

-- Generación de la señal de reset

```

FDS_INST0 : FDS
port map(
  C => S_ClkRdyDs,
  D => '0',
  S => '0',
  Q => FDS_Q_OUT
);

```

```

FD1_INST0 : FD
port map(
  C => S_ClkRdyDs,
  D => FDS_Q_OUT,
  Q => FD1_Q_OUT
);

```

```

FD2_INST0 : FD
port map(
  C => S_ClkRdyDs,
  D => FD1_Q_OUT,
  Q => FD2_Q_OUT
);

```

```

FD3_INST0 : FD
port map(
  C => S_ClkRdyDs,
  D => FD2_Q_OUT,
  Q => FD3_Q_OUT
);

```

```

OR2_INST0 : OR2
port map(
  I0 => S_DCM0LockedInv,
  I1 => OR3_O_OUT,
  O  => DCM1Reset
);

```

```

OR3_INST0 : OR3
port map(
  I0 => FD3_Q_OUT,
  I1 => FD2_Q_OUT,
  I2 => FD1_Q_OUT,

```

```

    0 => OR3_0_OUT
);
end Behavioral;

```

## A.4. Módulo de la FIFO asíncrona

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;

entity Afifo is
generic
(
    DATA_WIDTH: integer:=16;
    ADDR_WIDTH: integer:=12
);
port(
    Reset      : in  std_logic;

    WClk      : in  std_logic;
    RClk      : in  std_logic;

    RDataPort : out std_logic_vector(DATA_WIDTH-1 downto 0);
    WDataPort : in  std_logic_vector(DATA_WIDTH-1 downto 0);

    WR_En     : in  std_logic;
    RD_En     : in  std_logic;

    Empty     : out std_logic;
    Full      : out std_logic;
    Half_Full : out std_logic
);
end Afifo;

architecture structural of Afifo is

    -- Tamaño de la FIFO
    constant FIFO_DEPTH :integer := 2**ADDR_WIDTH;

    type RAM is array (integer range <>)of std_logic_vector (DATA_WIDTH-1 downto 0);
    signal Mem : RAM (0 to FIFO_DEPTH-1);

    -- Componente para el manejo del apuntador de lectura
    component RptrCtrl is
generic
(
    ADDR_WIDTH: integer := ADDR_WIDTH
);
port(
    Reset      : in  std_logic;
    RClk      : in  std_logic;
    GryRptr   : out std_logic_vector(ADDR_WIDTH  downto 0); -- utilizado para sincronizacion
    RptrAddr  : out std_logic_vector(ADDR_WIDTH-1 downto 0); -- utilizado para direccionamiento de la fifo
    SyncGryWpnr : in  std_logic_vector(ADDR_WIDTH  downto 0); -- utilizado para generar las banderas
    RptrInc   : in  std_logic;
    Empty     : out std_logic
);
end component;

    -- Componente para el manejo del apuntador de escritura
    component WpnrCtrl is

```

```

generic
(
  ADDR_WIDTH: integer:= ADDR_WIDTH
);
port(
  Reset      : in  std_logic;
  WClk       : in  std_logic;
  GryWptr    : out std_logic_vector(ADDR_WIDTH  downto 0);-- utilizado para sincronización
  WptrAddr   : out std_logic_vector(ADDR_WIDTH-1 downto 0);-- utilizado para direccionamiento de la FIFO
  SyncGryRptr : in  std_logic_vector(ADDR_WIDTH  downto 0);-- utilizado para generar las indicaciones
  WptrInc    : in  std_logic;
  Full       : out std_logic;
  Half_Full  : out std_logic
);
end component;

-- Señales auxiliares
signal S_GryRptr   : std_logic_vector(ADDR_WIDTH downto 0);
signal S_GryWptr   : std_logic_vector(ADDR_WIDTH downto 0);
signal S_RptrInc   : std_logic;
signal S_WptrInc   : std_logic;
signal S_WptrAddr  : std_logic_vector(ADDR_WIDTH-1 downto 0);
signal S_RptrAddr  : std_logic_vector(ADDR_WIDTH-1 downto 0);
signal S_Empty     : std_logic;
signal S_Full      : std_logic;
signal S_Half_Full : std_logic;

begin

  Afifo_RptrCtrl: RptrCtrl
  generic map
  (
    ADDR_WIDTH => ADDR_WIDTH
  )
  port map(
    Reset      => Reset,
    RClk       => RClk,
    GryRptr    => S_GryRptr,
    RptrAddr   => S_RptrAddr,
    SyncGryWptr => S_GryWptr,
    RptrInc    => S_RptrInc,
    Empty      => S_Empty
  );

  Afifo_WptrCtrl: WptrCtrl
  generic map
  (
    ADDR_WIDTH => ADDR_WIDTH
  )
  port map(
    Reset      => Reset,
    WClk       => WClk,
    GryWptr    => S_GryWptr,
    WptrAddr   => S_WptrAddr,
    SyncGryRptr => S_GryRptr,
    WptrInc    => S_WptrInc,
    Full       => S_Full,
    Half_Full  => S_Half_Full
  );

  -- Incremento del apuntador de lectura
  S_RptrInc <= RD_En;

  -- Incremento del apuntador de escritura
  S_WptrInc <= WR_En;

```

```

-- FIFO vacía
Empty      <= S_Empty;

-- FIFO llena
Full       <= S_Full;

-- FIFO llena por la mitad
Half_Full <= S_Half_Full;

-- Operación de lectura
process (RClk)
begin
    if (rising_edge(RClk)) then
        if (RD_En = '1' and S_Empty = '0') then
            RDataPort <= Mem(conv_integer(S_RptrAddr));
        end if;
    end if;
end process;

-- Operación de escritura
process(WClk)
begin
    if(rising_edge(WClk)) then
        if(WR_En = '1' and S_Full = '0') then
            Mem(conv_integer(S_WptrAddr)) <= WDataPort;
        end if;
    end if;
end process;
end structural;

```

#### A.4.1. Módulo de sincronización del apuntador de lectura

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.all;
use IEEE.STD_LOGIC_ARITH.all;

entity RptrCtrl is
    generic
    (
        ADDR_WIDTH: integer:= 12 -- N+1
    );
    port(
        Reset      : in  std_logic;
        RClk       : in  std_logic;
        GryRptr    : out std_logic_vector(ADDR_WIDTH  downto 0); -- utilizado para sincronización
        RptrAddr   : out std_logic_vector(ADDR_WIDTH-1 downto 0); -- utilizado para direccionamiento de la FIFO
        SyncGryWptr : in  std_logic_vector(ADDR_WIDTH  downto 0); -- utilizado para generar las indicaciones
        RptrInc    : in  std_logic;
        Empty      : out std_logic
    );
end RptrCtrl;

architecture Behavioral of RptrCtrl is

    component GrayCounter is
        generic(
            COUNTER_WIDTH :integer := ADDR_WIDTH
        );
        port(
            Reset      : in  std_logic;
            Clk        : in  std_logic;

```

```

    Enable    : in  std_logic;
    GrayCount : out std_logic_vector(COUNTER_WIDTH downto 0)
);
end component;

-- Señales auxiliares
signal S_GryRptr      : std_logic_vector(ADDR_WIDTH downto 0);
signal S_RptrInc     : std_logic;
signal S_Empty       : std_logic;
signal S_HybGryRptr  : std_logic_vector(ADDR_WIDTH downto 0);
signal S_HybSyncGryWptr : std_logic_vector(ADDR_WIDTH downto 0);

begin

-- Componente para conversión a código Gray
GrayCounter_Rptr: GrayCounter
generic map
(
    COUNTER_WIDTH => ADDR_WIDTH
)
port map(
    Reset    => Reset,
    Clk     => RClk,
    Enable   => S_RptrInc,
    GrayCount => S_GryRptr
);

-- Incremento del apuntador dual
S_RptrInc <= '1' when (RptrInc = '1' and S_Empty = '0') else '0';

-- Apuntador dual en código Gray
GryRptr <= S_GryRptr;

-- Se obtiene apuntador Gray híbrido
S_HybGryRptr <= S_GryRptr(ADDR_WIDTH) & (S_GryRptr(ADDR_WIDTH)
xor S_GryRptr(ADDR_WIDTH-1)) & S_GryRptr(ADDR_WIDTH-2 downto 0);

-- Se obtiene apuntador síncrono de lectura híbrido
S_HybSyncGryWptr <= SyncGryWptr(ADDR_WIDTH) & (SyncGryWptr(ADDR_WIDTH)
xor SyncGryWptr(ADDR_WIDTH-1)) & SyncGryWptr(ADDR_WIDTH-2 downto 0);

--Apuntador Gray híbrido para direccionamiento de memoria
RptrAddr<= S_HybGryRptr(ADDR_WIDTH-1 downto 0);

-- Se genera la señal de FIFO vacía
S_Empty<= '1' when (S_HybGryRptr = S_HybSyncGryWptr) else '0';

-- Indicación de FIFO vacía
Empty <= S_Empty;

end Behavioral;

```

#### A.4.2. Módulo de sincronización del apuntador de escritura

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.all;
use IEEE.STD_LOGIC_ARITH.all;

entity WptrCtrl is
generic
(

```

```

    ADDR_WIDTH: integer:= 12
);
port(
    Reset      : in  std_logic;
    WClk       : in  std_logic;
    GryWptr    : out std_logic_vector(ADDR_WIDTH  downto 0);
    WptrAddr   : out std_logic_vector(ADDR_WIDTH-1 downto 0);
    SyncGryRptr : in  std_logic_vector(ADDR_WIDTH  downto 0);
    WptrInc    : in  std_logic;
    Full       : out std_logic;
    Half_Full  : out std_logic
);
end WptrCtrl;

architecture Behavioral of WptrCtrl is

    component GrayCounter is
    generic(
        COUNTER_WIDTH :integer := ADDR_WIDTH
    );
    port(
        Reset      : in  std_logic;
        Clk        : in  std_logic;
        Enable     : in  std_logic;
        GrayCount  : out std_logic_vector(COUNTER_WIDTH downto 0)
    );
    end component;

    signal S_GryWptr      : std_logic_vector(ADDR_WIDTH downto 0);
    signal S_WptrInc      : std_logic;
    signal S_Full         : std_logic;
    signal S_Half_Full    : std_logic;
    signal S_EqualAddr    : std_logic;
    signal S_HybGryWptr   : std_logic_vector(ADDR_WIDTH  downto 0);
    signal S_HybSyncGryRptr : std_logic_vector(ADDR_WIDTH  downto 0);
    signal S_WptrAddr     : std_logic_vector(ADDR_WIDTH-1 downto 0);
    signal S_SyncRptrAddr : std_logic_vector(ADDR_WIDTH-1 downto 0);

begin

    GrayCounter_Wptr: GrayCounter
    generic map
    (
        COUNTER_WIDTH => ADDR_WIDTH
    )
    port map(
        Reset      => Reset,
        Clk        => WClk,
        Enable     => S_WptrInc,
        GrayCount  => S_GryWptr
    );

    -- Incremento del apuntador dual
    S_WptrInc <= '1' when (WptrInc = '1' and S_Full = '0') else '0';

    -- Apuntador dual en código Gray
    GryWptr <= S_GryWptr;

    -- Apuntador para direccionamiento de escritura
    WptrAddr <= S_WptrAddr;

    -- Se obtiene apuntador Gray híbrido
    S_HybGryWptr <= S_GryWptr(ADDR_WIDTH) & (S_GryWptr(ADDR_WIDTH)
        xor S_GryWptr(ADDR_WIDTH-1)) & S_GryWptr(ADDR_WIDTH-2 downto 0);

```

```

-- Se obtiene apuntador sincrono de lectura híbrido
S_HybSyncGryRptr <= SyncGryRptr(ADDR_WIDTH) & (SyncGryRptr(ADDR_WIDTH)
xor SyncGryRptr(ADDR_WIDTH-1)) & SyncGryRptr(ADDR_WIDTH-2 downto 0);

--Apuntador Gray híbrido para direccionamiento de escritura
S_WptrAddr <= S_HybGryWptr(ADDR_WIDTH-1 downto 0);

--Apuntador Gray híbrido para direccionamiento de lectura
S_SyncRptrAddr <= S_HybSyncGryRptr(ADDR_WIDTH-1 downto 0);

-- Comprobamos si las direcciones son iguales
S_EqualAddr <= '1' when (S_WptrAddr = S_SyncRptrAddr) else '0';

-- Se genera la señal de FIFO llena
S_Full <= '1' when ((S_HybGryWptr(ADDR_WIDTH) /= S_HybSyncGryRptr(ADDR_WIDTH)) and (S_EqualAddr = '1')) else '0';

-- Se genera la señal de FIFO llena a la mitad
S_Half_Full <= '1' when (S_HybSyncGryRptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "00")
and ( S_HybGryWptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "01") else

'1' when (S_HybSyncGryRptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "01")
and ( S_HybGryWptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "10") else

'1' when (S_HybSyncGryRptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "10")
and ( S_HybGryWptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "11") else

'1' when (S_HybSyncGryRptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "11")
and ( S_HybGryWptr(ADDR_WIDTH downto ADDR_WIDTH - 1) = "00") else

'0';

-- Señal de FIFO llena
Full <= S_Full;

-- Señal de FIFO llena a la mitad
Half_Full <= S_Half_Full;

end Behavioral;

```

### A.4.3. Módulo de conversión a código Gray

```

library ieee;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.all;
use IEEE.STD_LOGIC_ARITH.all;

entity GrayCounter is
generic (
COUNTER_WIDTH :integer := 10
);
port(
Reset      :in  std_logic;
Clk       :in  std_logic;
Enable    :in  std_logic;
GrayCount :out std_logic_vector(COUNTER_WIDTH downto 0)
);
end entity;

architecture rtl of GrayCounter is

-- Señales auxiliares
signal S_BinCnt      : std_logic_vector(COUNTER_WIDTH downto 0);
signal S_BinCntNxt  : std_logic_vector(COUNTER_WIDTH downto 0);
signal S_GryCnt      : std_logic_vector(COUNTER_WIDTH downto 0);

```

```

    signal S_GryCntNxt : std_logic_vector(COUNTER_WIDTH downto 0);
begin

    process (Clk,Reset)
    begin
        if (Reset = '1') then
            S_GryCnt <= (others=>'0');
            GrayCount <= (others=>'0');
        elsif(rising_edge(Clk))then
            if(Enable = '1')then
                GrayCount <= S_GryCntNxt;
                S_GryCnt <= S_GryCntNxt;
            end if;
        end if;
    end process;

    -- Gray A Binario
    S_BinCnt <= S_GryCnt xor ('0'& S_BinCnt(COUNTER_WIDTH downto 1));

    -- Incremento
    S_BinCntNxt <= S_BinCnt + 1;

    -- Binario a Gray
    S_GryCntNxt <= S_BinCntNxt xor ('0'& S_BinCntNxt(COUNTER_WIDTH downto 1));

end architecture;

```

## A.5. Módulo controlador del ADC

```

library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;

library UNISIM;
    use UNISIM.VComponents.all;

entity ADC is
    port(
        Reset      : in std_logic;
        ClkRdy     : in std_logic;
        ClkRdy180 : in std_logic;
        Dx_y_P     : in std_logic_vector(7 downto 0);
        Dx_y_M     : in std_logic_vector(7 downto 0);
        Data       : out std_logic_vector(15 downto 0)
    );
end ADC;

architecture Behavioral of ADC is
    -- Señales auxiliares
    signal Dx_y_out: std_logic_vector(7 downto 0);
    signal O_Data:std_logic_vector(7 downto 0);
    signal E_Data:std_logic_vector(7 downto 0);
begin

    -- Manejo de las entradas diferenciales
    LOOP0 : for i in 0 to 7 generate
        IBUFDS_INST : IBUFDS
        generic map (DIFF_TERM => TRUE,IBUF_DELAY_VALUE => "0", IFD_DELAY_VALUE => "AUTO",IOSTANDARD => "LVDS_25")
        port map (I => Dx_y_P(i), IB => Dx_y_M(i),O => Dx_y_out(i));
    end generate;

    -- Captura de datos en los flancos positivo y negativo de la señal de reloj
    LOOP1: for j in 0 to 7 generate
        IDDR2_INST : IDDR2

```

```

generic map(
  DDR_ALIGNMENT => "NONE",
  INIT_Q0 => '0',
  INIT_Q1 => '0',
  SRTYPE => "ASYNC")
port map (
  Q0 => E_Data(j),
  Q1 => O_Data(j),
  C0 => ClkRdy,
  C1 => ClkRdy180,
  CE => '1',
  D => Dx_y_out(j),
  R => Reset,
  S => '0'
);
end generate;

-- Recuperación de una muestra de 16 bits
Data(0) <= E_Data(0);
Data(2) <= E_Data(1);
Data(4) <= E_Data(2);
Data(6) <= E_Data(3);
Data(8) <= E_Data(4);
Data(10) <= E_Data(5);
Data(12) <= E_Data(6);
Data(14) <= E_Data(7);
Data(1) <= O_Data(0);
Data(3) <= O_Data(1);
Data(5) <= O_Data(2);
Data(7) <= O_Data(3);
Data(9) <= O_Data(4);
Data(11) <= O_Data(5);
Data(13) <= O_Data(6);
Data(15) <= O_Data(7);

end Behavioral;

```

## A.6. Módulo controlador del DAC

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity DAC is
  port(
    Reset      : in std_logic;
    Clk4X      : in std_logic;
    Xmt        : in std_logic;
    Empty      : in std_logic;
    RDEnable   : out std_logic
  );
end DAC;

architecture Behavioral of DAC is

  -- Señal auxiliar
  signal S_Enable : std_logic:= '0';

begin

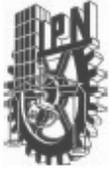
  -- Proceso para la transmisión de datos
  process(Clk4X,Reset)
  begin
    if(Reset = '1')then

```

```
S_Enable <= '0';
elsif(rising_edge(Clk4X))then
  if(Xmt = '1')then
    S_Enable <= '1';
  elsif(Empty = '1') then
    S_Enable <= '0';
  end if;
end if;
end process;

-- Señal de habilitación para la FIFO de transmisión
RDEnable <= S_Enable;

end Behavioral;
```



# INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

## ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 14:00 horas del día 15 del mes de diciembre de 2011 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

*Centro de Investigación en Computación*

para examinar la tesis titulada:

### "Módem para la comunicación por la red eléctrica"

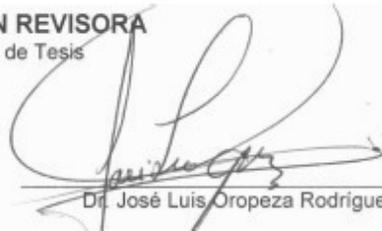
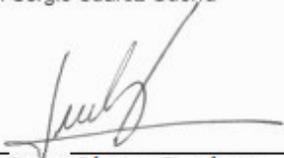
Presentada por el alumno:

<b>ANTONIO</b> Apellido paterno	<b>MÉNDEZ</b> Apellido materno	<b>RAÚL</b> Nombre(s)
Con registro:		B 0 9 1 6 7 3

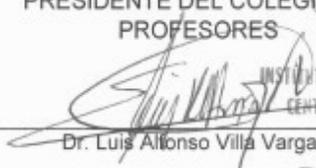
aspirante de: **MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

#### LA COMISIÓN REVISORA Directores de Tesis

 _____ Dr. Ricardo Barrón Fernández	 _____ Dr. José Luis Oropeza Rodríguez
 _____ Dr. Sergio Suárez Guerra	 _____ Dr. Oleksiy Pogrebnyak
 _____ Dr. Luis Pastor Sánchez Fernández	

#### PRESIDENTE DEL COLEGIO DE PROFESORES

  
\_\_\_\_\_  
Dr. Luis Alfonso Villa Vargas

  
INSTITUTO POLITÉCNICO NACIONAL  
CENTRO DE INVESTIGACIÓN  
EN COMPUTACIÓN  
DIRECCIÓN



**INSTITUTO POLITÉCNICO NACIONAL**

*SECRETARÍA DE INVESTIGACIÓN Y PROGRADO*

*CARTA DE CESIÓN DE DERECHOS*

En la Ciudad de México, del día 15 del mes de diciembre del año 2011, el que suscribe **Raúl Antonio Méndez**, alumno del programa de **Maestría en Ciencias en Ingeniería de Cómputo**, con número de registro **B091673**, adscrito al **Centro de Investigación en Computación**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del **Dr. Ricardo Barrón Fernández** y el **Dr. José Luis Oropeza Rodríguez**, y cede los derechos del trabajo intitulado: **“Módem para la comunicación por la red eléctrica”**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual. Gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: [amendezb09@sagitario.cic.ipn.mx](mailto:amendezb09@sagitario.cic.ipn.mx), [bechguishe@hotmail.com](mailto:bechguishe@hotmail.com). Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

A handwritten signature in black ink, appearing to read 'Raúl', is positioned above the printed name.

**Raúl Antonio Méndez**