



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons  
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite  
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
Facultad de Ciencias e Ingeniería



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DEL PERÚ

# DISEÑO E IMPLEMENTACIÓN DE UN MODULADOR OFDM RECONFIGURABLE PARA LA TECNOLOGÍA SOFTWARE-DEFINED RADIO SOBRE UN FPGA

*Tesis para optar el título de Ingeniero Electrónico*

Presentado por:  
Oscar Igor Robles Palacios

Asesor:  
Carlos Silva Cárdenas

Lima – Perú  
2009

## RESUMEN

El presente trabajo realizó el diseño de un Modulador OFDM Reconfigurable orientado al desarrollo de la tecnología Software-Defined Radio sobre un FPGA mediante el uso de lenguaje de descripción de hardware VHDL. La simulación ideal del sistema se realizó en el software Matlab R2007 con los objetivos de comprender mejor el funcionamiento y tener un punto de referencia luego de obtener las simulaciones del circuito. Cabe mencionar que el sistema implementado no abarcó la etapa de banda base de la señal de entrada, pues se asume que llega una trama de datos de una determinada tasa de transferencia y con los rangos de voltajes TTL. Así también, la procedencia de la señal de entrada (datos, vídeo o audio) tampoco se contempló en el presente trabajo. En consecuencia, la tesis se concentró básicamente en el diseño del sistema y las pruebas correspondientes de funcionamiento.

Según esto, se procedió a dividir la arquitectura en sus principales unidades funcionales para poder detallar el esquema planteado para cada una de ellas. Así, se especifica las consideraciones y problemas encontrados en la etapa de diseño de cada una de ellas, y se evidencia su correcto funcionamiento mediante simulaciones aisladas de cada una de ellas. A continuación, se explica la arquitectura planteada para todo el sistema, en función a los bloques anteriormente detallados, mostrando finalmente la simulación de toda la arquitectura que asegura el éxito en la etapa de diseño. Sin embargo, además de toda la arquitectura se tuvo que implementar pequeños bloques externos para tener el ambiente apropiado para las pruebas finales del circuito. Así, se tuvo que manejar tanto el puerto VGA para obtener la señal analógica, y uno de los PLLs internos del FPGA para poder obtener la frecuencia de reloj necesaria para el sistema.

El Modulador OFDM Reconfigurable se implementó en la tarjeta de desarrollo DE2, la cual contiene el FPGA Cyclone II EP2C35F672C6 de la compañía Altera, a una frecuencia de 200 MHz (obtenida a partir del oscilador de 50 MHz y el PLL embebido del FPGA). Para la verificación del funcionamiento, se utilizó básicamente los resultados de la simulación y otras herramientas del software Quartus II 8.0 Web Edition, y principalmente, el análisis de la señal obtenida mediante un osciloscopio. La señal obtenida posee un SFDR de 63 dBc, lo cual es un resultado aceptable. Adicionalmente, se obtuvo que la máxima latencia del sistema, una vez ya acabada la etapa de lectura de datos, es de 19.23 ns. Esto demuestra la gran velocidad de procesamiento y eficiencia del sistema implementado.



*A Víctor y Myriam, mis padres, quienes no siempre me entendieron, pero siempre me apoyaron.*

*A Daniel, Vladimir y Karina, cuyos lazos de hermandad demostraron superar toda distancia.*

*A Jorge, Jaime, Miguel, Eduart, Mario, Cynthia, Taliana, Erick y Roddy, por brindarme su amistad y hacerme sentir como en casa en esta ciudad extraña.*

*A José, Sergio, Stephany, Karla, Ana María, Fernando, Paolo y Hugo, cuya amistad siempre me hizo sentir de vuelta en mi ciudad natal.*

*A Daniel, Willy, Jorge, Enrique, Karina y Francis, quienes me acogieron como uno más de su promoción y alegraron mis días universitarios.*

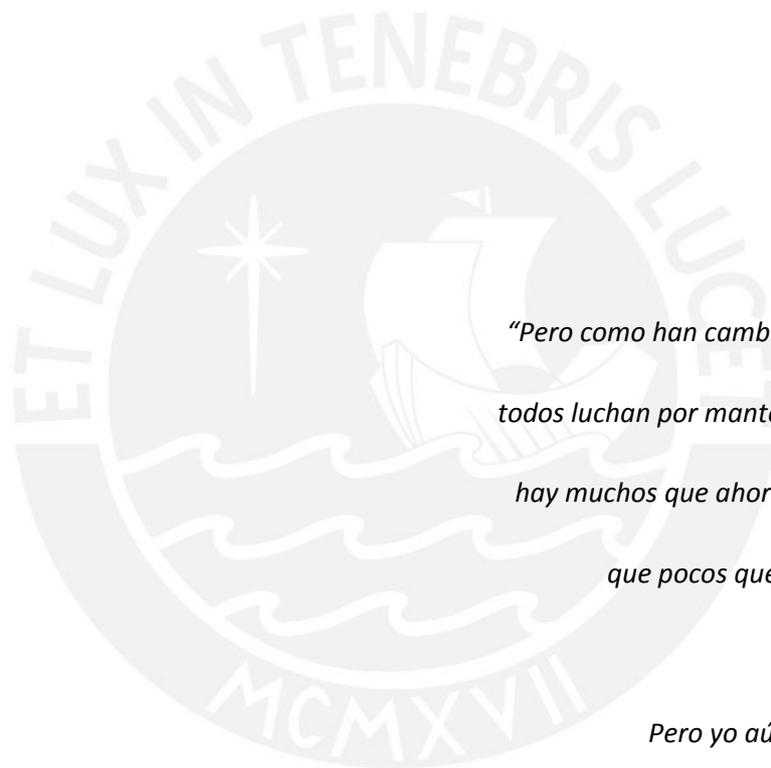
*A Kristy...juntos o separados, siempre sacó a relucir la mejor parte de mí.*

*A mi asesor, el Dr. Carlos Silva, por todo el apoyo brindado y sabiduría compartida.*

*A mi Grupo de Microelectrónica (GuE), por mostrarme el camino hacia mi verdadero futuro.*

*A todos, mencionados o no, pues ayudaron a convertirme en la persona que soy ahora.*

*Gracias!!!*



*“Pero como han cambiado los tiempos  
todos luchan por mantener sus puestos  
hay muchos que ahora son ingenieros  
que pocos quedaron rockeros.*

*Pero yo aún sigo cantando  
y lo voy a seguir haciendo  
una lección me dio la vida  
tenés que hacer lo que el corazón diga”*

*Enanitos Verdes – Aún sigo cantando*

## INDICE

<u>INTRODUCCIÓN</u> .....	i
<u>CAPÍTULO 1: ASPECTOS GENERALES DE SISTEMAS SDR</u> .....	1
1.1. El concepto de Software-Defined Radio.....	1
1.2. Esquema general de un SDR.....	2
1.3. Técnicas de Síntesis Digital Directa en SDRs.....	3
1.4. Implementaciones.....	5
1.4.1. USRP y GNU Radio.....	5
1.4.2. KUAR: Kansas University Agile Radio.....	7
<u>CAPÍTULO 2: TRANSMISORES OFDM</u> .....	9
2.1. Modulación OFDM.....	9
2.1.1. Transformada de Fourier: DFT y FFT.....	11
2.2. Etapas de un transmisor OFDM.....	11
2.3. Técnicas de implementación de moduladores OFDM.....	12
2.3.1. Uso de DSP (Digital Signal Processor).....	12
2.3.2. Uso de DUC (Digital Up-Converter).....	13
<u>CAPÍTULO 3: PLANTEAMIENTOS PARA LA IMPLEMENTACIÓN DEL MODULADOR OFDM RECONFIGURABLE</u> .....	15
3.1. Modelo Teórico.....	15
3.2. Definiciones Operativas.....	17
3.3. Hipótesis de la Investigación.....	19
3.3.1. Hipótesis Principal.....	19
3.3.2. Hipótesis Secundaria.....	19
3.4. Objetivos de la Investigación.....	20
3.4.1. Objetivo Principal.....	20
3.4.2. Objetivos Específicos.....	20
3.5. Metodología de la Investigación.....	21
3.6. Análisis y Determinación de Requerimientos.....	24

3.7. Consideraciones para el diseño.....	25
--	----

#### CAPÍTULO 4: DISEÑO DE LA ARQUITECTURA DEL MODULADOR OFDM SOBRE EL FPGA CYCLONE II DE ALTERA.....

4.1. Diseño del modulador OFDM reconfigurable en software.....	28
4.2. Diseño del modulador OFDM reconfigurable sobre el FPGA Cyclone II de Altera.....	30
4.2.1. Descripción y diseño de la arquitectura del Sintetizador Digital Directo (DDS).....	32
4.2.1.1. Acumulador de Fase.....	34
4.2.1.2. Convertidor Fase-Amplitud.....	35
4.2.2. Descripción y diseño de la arquitectura del Mapeador de Constelaciones (Mapeador I&Q).....	37
4.2.2.1. Cargador de Datos del Registro.....	41
4.2.2.2. Extractor de Selectores.....	43
4.2.2.3. Selector de Constelaciones.....	44
4.2.3. Descripción y diseño de la arquitectura del Convertidor Serial/Paralelo.....	46
4.2.4. Descripción y diseño de la arquitectura de la Unidad de Control.....	48
4.2.5. Descripción y diseño de la arquitectura del Modulador OFDM Reconfigurable.....	49

#### CAPÍTULO 5: IMPLEMENTACIÓN Y RESULTADOS.....

5.1. Simulaciones.....	51
5.2. Implementación.....	57

#### CONCLUSIONES.....

#### RECOMENDACIONES.....

#### BIBLIOGRAFÍA.....

#### ANEXOS

## INTRODUCCIÓN

Actualmente, los Procesadores Digitales de Señales (DSPs: Digital Signal Processors) implementados sobre dispositivos de lógica programable están tomando mayor relevancia en el mundo de las telecomunicaciones digitales. Esto se deriva del hecho que el procesamiento digital de la señal está siendo expandido incluso hasta la zona de las antenas, y ya no solamente a la etapa de banda base. Por tanto, las fracciones de muestreo y la rapidez de procesamiento necesarias no pueden ser alcanzadas por Procesadores Digitales de Señales Programables (PDSPs: Programmable DSPs) convencionales. Son en estas circunstancias que se convierte en idónea la implementación de un DSP sobre un dispositivo de lógica programable, pues además de tener mayores prestaciones en los aspectos antes mencionados, otorgan un mayor nivel de reconfiguración del Front-End digital del sistema.

Por los motivos expuestos anteriormente, el uso de los dispositivos de lógica programable y los lenguajes de descripción de hardware ha aumentado paulatinamente debido a que contribuyen a la reducción de costos de los procesos de diseño. Así, estos dispositivos se están convirtiendo en elementos básicos del flujo de diseño de hardware digital.

La tecnología de Radios Definidos por Software (SDR: Software-Defined Radios) fue creada para obtener tanto una permanente comunicación dentro de diferentes bandas de frecuencia del espectro con un solo dispositivo, como adaptabilidad frente a las nuevas innovaciones de componentes y equipos. Básicamente, se debe transferir a software muchas de las funciones antiguamente atribuidas al hardware.

Debido a la aparición de esta tecnología se ha visto necesario la implementación de transmisores y receptores altamente configurables. En consecuencia, la implementación de algoritmos de modulación variables es una de las principales tareas para alcanzar los objetivos de esta tecnología. Es por este motivo que muchos diseñadores ya han hecho diversos trabajos acerca de moduladores y transmisores reconfigurables, siendo el algoritmo más común el de Multiplexación Ortogonal por División de Frecuencias (OFDM: Orthogonal Frequency Division

Multiplexing). La popularidad de esta técnica de modulación se debe a su eficiencia y su robustez, lo que ha derivado en el aumento de su uso para distintas comunicaciones inalámbricas.

Por lo expuesto anteriormente, el objetivo de este trabajo es el diseño e implementación de un modulador OFDM reconfigurable sobre un dispositivo de lógica programable pensado para la tecnología SDR.



## CAPÍTULO 1: ASPECTOS GENERALES DE SISTEMAS SDR

### 1.1. El concepto de Software-Defined Radio.

Paulatinamente, los sistemas de comunicación han ido alcanzando mayores niveles de complejidad y de flexibilidad, siendo la tendencia más fuerte y actual es buscar un solo dispositivo que se pueda adaptar a diferentes tipos de radioenlaces. De esta necesidad es que nace la tecnología SDR (Software-Defined Radios o Radios Definidos por Software).

Existen muchas definiciones de la tecnología SDR, a pesar de que todos aquellos que han dado una de esas definiciones resaltan la necesidad de esclarecer los propósitos y los objetivos de esta tecnología. Sin embargo, quizás la mejor definición que se le puede dar es la siguiente: implementación en software del terminal de usuario, el cual es capaz de adaptarse dinámicamente al ambiente del radioenlace en el que se encuentra [1]. Esto implica que muchas de las funcionalidades de la interfaz de radio serán definidas mediante software, a diferencia de implementaciones pasadas en las cuales estas funcionalidades se plasman sobre hardware dedicado. En otras palabras, un transceptor SDR debe permitir la manipulación de sus principales características a través de software [1]. Entre estas características están:

- Banda de frecuencia
- Ancho de Banda
- Esquemas de modulación y codificación
- Recursos de Radio y Gestión de movilidad
- Aplicaciones de Usuario

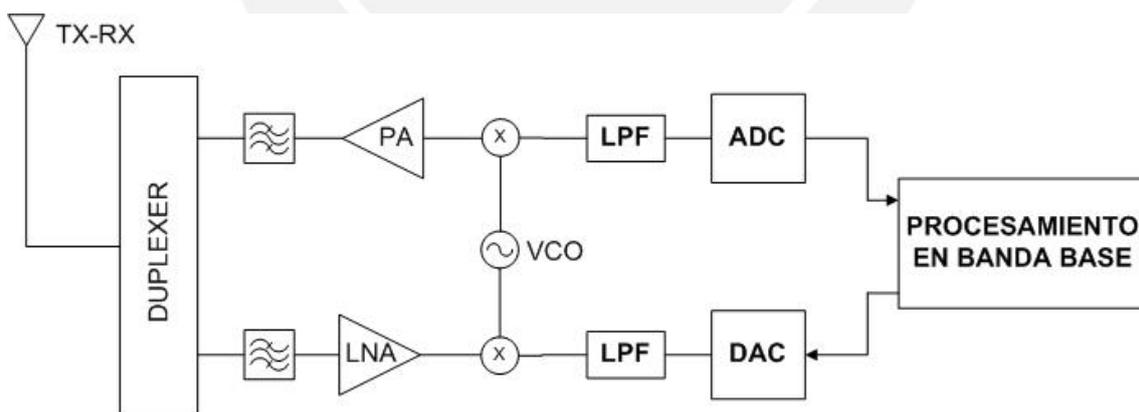
Así, lo que se pretende con los sistemas SDR es obtener un dispositivo que sea multi-servicio, multi-estándar, multi-banda, y en todo aspecto reconfigurable mediante una interfaz de software [1]. En consecuencia de esta necesidad de reconfigurabilidad, la antigua opción de facto para el diseño de transceptores, ASIC (Application Specific Integrated Circuits o Circuito Integrado de Aplicación Específica), está siendo reemplazada por el uso de dispositivos de lógica programable y el uso de procesadores para la implementación de los algoritmos y protocolos de codificación y modulación.

Cabe decir también, que esta tecnología SDR tiene como visión final el desarrollo e implementación de Radios Cognitivos (CR: Cognitive Radios) le darían total autonomía y flexibilidad a los sistemas de transmisión para adaptarse a las características requeridas.

## 1.2. Esquema general de un SDR.

Un dispositivo SDR consiste en un transceptor reconfigurable mediante software. Esto implica que debe ser capaz de recibir y transmitir en simultáneo y en todo momento, con las características establecidas mediante la interfaz de software. Ya que el lograr este nivel de flexibilidad implica un gran cambio en el diseño del hardware involucrado en el proceso de implementación tradicional de transmisores y receptores, es necesario tener claro, al menos a nivel conceptual, las partes y etapas básicas de un dispositivo SDR. Así, de un diagrama o esquema general se podrá tener un mayor entendimiento de las prestaciones, alcances y limitaciones de este tipo de sistemas.

A pesar de las variaciones en cuanto al enfoque del diseño de estos dispositivos en contraste con el diseño clásico de transceptores basados en hardware dedicado, los transceptores SDR comparten con sus antepasados varias de las etapas y unidades funcionales analógicas. Sin embargo, cabe mencionar que la principal diferencia entre ambos es que los dispositivos SDR siempre buscarán tener control incluso sobre estas unidades funcionales analógicas.



**Figura 1. Diagrama de bloques de un transceptor SDR.**

Como se puede observar de la Figura 1, los bloques básicos de la etapa de potencia de un Front-End de RF (Radio Frecuencia), tales como el Amplificador de Potencia (PA: Power Amplifier)

en el transmisor y el Amplificador de Bajo Ruido (LNA: Low Noise Amplifier) en el receptor [1], siguen vigentes para un SDR. Del mismo modo, el filtro pasa-banda que se coloca antes de la antena tanto en receptor como transmisor también estará presente en estos dispositivos. Quizás lo que es necesario mencionar es que las ganancias de ambos amplificadores deben ser configuradas y controladas mediante la unidad de Procesamiento en Banda Base.

Otro bloque importante en el esquema presentado por la Figura 1, es el reemplazo del VCO (Voltage Controlled Oscillator u Oscilador Controlado por Voltaje) por el tradicional Oscilador Local utilizado con numerosidad en implementaciones pasadas. Esto se da para tener un mayor control sobre la frecuencia de la señal portadora que es básicamente el componente principal del Front-End de RF. Así se puede variar las bandas de transmisión y recepción del sistema SDR.

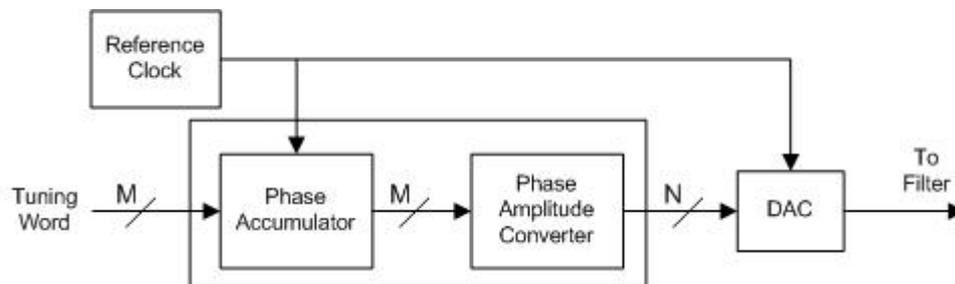
Finalmente, dos elementos importantes de un transceptor SDR son los convertidores análogos-digitales (ADC: Analog to Digital Converter) para el receptor, y convertidores digitales-análogos (DAC: Digital to Analog Converter) para el transmisor. Estos dos elementos son los que permiten que todo el procesamiento de la señal sea complemente digital en la etapa de banda base. Esto, a su vez, garantiza la versatilidad de todo el sistema pues el uso de sistemas digitales es lo que da provee el nivel de flexibilidad y reconfigurabilidad a todo sistema de comunicaciones. Así, el procesamiento digital en banda base se encargará de ejecutar los algoritmos de modulación y codificación requeridos por el usuario, e interpretar las necesidades del mismo enviadas mediante el software de manejo del SDR.

### 1.3. Técnicas de Síntesis Digital Directa en SDRs.

Se define como Síntesis Digital Directa como aquel método de control digital capaz de generar múltiples frecuencias a partir de una señal de reloj. Es decir, esta técnica nos permite crear arbitrariamente distintos tipos de formas de onda a partir de una sola referencia (señal de reloj), y controlar digitalmente los parámetros de la forma de onda (amplitud, frecuencia, fase, etc.) [2]. Este control completo sobre la onda a generar presenta una gran ventaja sobre cualquier método de generación de onda analógica pues es mucho más preciso y otorga mayor facilidad en el control sobre la onda generada. Es por estas razones que el uso de esta técnica se está haciendo más popular en los sistemas de comunicaciones, y sobre todo en aquellos que requieren un gran nivel de control, velocidad y reconfigurabilidad como son los SDRs.

De la teoría de Síntesis Digital Directa se llegó a la implementación de un sistema capaz de generar una señal a partir de una sola frecuencia de reloj. Debido a que es en esencia un oscilador, se le denominó Oscilador Controlado Numéricamente (NCO: Numerically Controlled Oscillator).

A pesar de que implementaciones actuales de NCOs, también denominado DDSs (Digital Direct Synthethizers o Sintetizadores Digitales Directos) presenta ciertas variaciones entre ellas, todas presentan los bloques básicos para la generación sincronizada y controlada de la señal requerida, sin importar la particularidad de la misma.



**Figura 2. Esquema básico de un DDS. [3]**

Todo NCO consiste básicamente en un circuito que recibe como entradas la señal de reloj y una señal de sintonización (Tuning Word), a partir de las cuales determina la frecuencia de la onda a generar [3], como se puede apreciar en:

$$f_o = \frac{M \times f_{clk}}{2^n} \quad (1)$$

donde  $M$  es la señal de sintonización,  $f_{clk}$  es la frecuencia de la señal de reloj, y  $2^n$  es la cantidad de puntos que representa la rueda de fase (todos los ángulos existentes: de  $0^\circ$  a  $360^\circ$ ). Así, todo DDS consta de las siguientes unidades funcionales, ilustradas también en la Figura 2:

- **Acumulador de Fase:** Consiste en un registro de datos de un determinado ancho de palabra, los cuales guardan la información de la fase de la señal. La secuencia de recorrido del registro completo es una sola: ascendente; sin embargo, el incremento de cada salto es establecido por la señal de sintonización.
- **Convertidor Fase-Amplitud:** Consiste en una tabla de datos, usualmente implementada sobre una Memoria de Solo-Lectura (ROM: Read-Only Memory), la cual contiene los datos de

amplitud correspondientes a valor de fase de la onda a generar. De esa manera, la señal que cumple el papel de dirección de memoria es la salida del Acumulador de Fase.

- **Convertidor Análogo-Digital:** Circuito que se encarga de convertir la secuencia de datos digitales enviados por el Convertidor Fase-Amplitud en la onda analógica deseada.
- **Filtro Pasa-bajos:** Circuito que se encarga de eliminar la presencia de espurias en la señal generada.

#### 1.4. Implementaciones.

Implementaciones de hardware reconfigurable para SDRs se ha venido realizando desde hace casi 7 años. Los complejos algoritmos de procesamiento de señal se presentan como las principales dificultades para los diseñadores. Las herramientas a usar para el diseño de este tipo de dispositivos son numerosas y variadas, y muchas veces el hardware final contiene una mezcla de todos ellos: FPGAs (Field Programmable Gate Arrays), DSPs (Digital Signal Processors o Procesadores Digitales de Señales), Procesadores, ASICs y SoCs (Systems on Chip o Sistemas sobre Chip). Dependiendo del tipo de transceptor requerido es que se debe seleccionar entre estas herramientas.

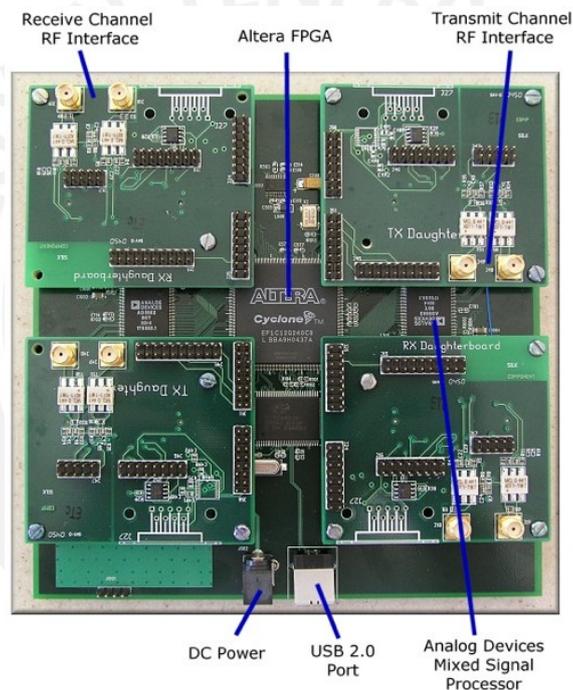
Los desarrollos iniciales apuntaban básicamente a conseguir un dispositivo capaz de lograr modular y demodular en cuadratura [4]. Así también, se buscaba que pudiera amoldarse a distintas constelaciones – como QPSK, QAM-16 y otros más – para darle así mayor versatilidad al sistema [5]. Sin embargo estos circuitos reconfigurables aún no cumplían con las expectativas de la tecnología SDR. A pesar de esto, debido a que ya existían desde hace largo tiempo mejores técnicas de modulación que usaban la modulación en cuadratura y las constelaciones como partes de un algoritmo más complejo, estas primeras implementaciones marcaban un buen punto de partida para mayores desarrollos en el área.

##### 1.4.1. USRP y GNU Radio.

El primer dispositivo SDR que logró alcanzar un gran nivel de versatilidad fue la USRP (Universal Software Radio Peripheral), la cual fue implementada en los Estados Unidos por la compañía Ettus Research [6]. Esta tarjeta incluía también el software aplicativo GNU Radio que lograba convertir la PC en una consola de procesamiento de las señales que recibía la USRP, así

como también permitía manejar el hardware mediante sentencias. Obviamente, el software GNU Radio se ejecuta principalmente sobre SO (Sistema Operativo) GNU/Linux; no obstante, también es compatible para MAC OS X.

Dentro de la USRP, se encuentran varios circuitos integrados, entre ellos convertidores análogos-digitales y digitales-análogos, filtros, amplificadores, y procesadores. No obstante, una de las unidades funcionales más importante de esta novedosa solución es el FPGA (Field Programmable Gate Array) Cyclone II de la compañía Altera. Este dispositivo de lógica programable es el encargado de realizar los pesados algoritmos de procesamiento digital de señales y sirve como nexo entre las etapas de banda base y banda intermedia. El uso de este FPGA es una de las características innovadoras de esta implementación.



**Figura 3. Tarjeta USRP. [6]**

El nivel de flexibilidad otorgado por la USRP es más que considerable. Sin embargo, para poder moverse entre las distintas bandas de frecuencia, necesita distintas tarjetas hijas correspondientes a los Front-Ends de RF específicos [6], como se observa en la Figura 3. A pesar de esto, la USRP se considera aún como uno de los primeros grandes logros en la tecnología SDR.

#### 1.4.2. KUAR: Kansas University Agile Radio.

A partir de la USRP y del aplicativo GNU Radio se han derivado numerosas implementaciones, ya sea tomando como base la USRP para la realización de sistemas SDR, o siguiendo el ejemplo de la compañía Ettus Research de usar FPGAs para el procesamiento de señal requerido [6]. A pesar de ello, las implementaciones más actuales superan los límites alcanzados por esta tarjeta. En la Universidad de Kansas, en el año 2007, se implementó otro prototipo de dispositivo SDR denominado KUAR (Kansas University Agile Radio). Este dispositivo, compatible también con el software GNU Radio, supera considerablemente a la USRP, pues puede funcionar en un vasto rango de frecuencias al igual que la implementación de la compañía Ettus Research sin necesidad de acoplar distintos Front-Ends de RF; es decir, solo hay un transmisor y un receptor en toda la tarjeta, y ambos funcionan para todo el rango de frecuencia del sistema [7]. Así también, el KUAR otorga mayor portabilidad pues no necesita de una de una PC para poder configurarlo. Este dispositivo posee un procesador encargado de reconocer las instrucciones y de mandar la información necesaria al resto del sistema. Como ha sido mencionado antes, el aplicativo GNU Radio puede ejecutarse sobre este dispositivo; esto es posible gracias al antes mencionado procesador, el cual es capaz de entender los comandos del Shell de SO GNU/Linux.

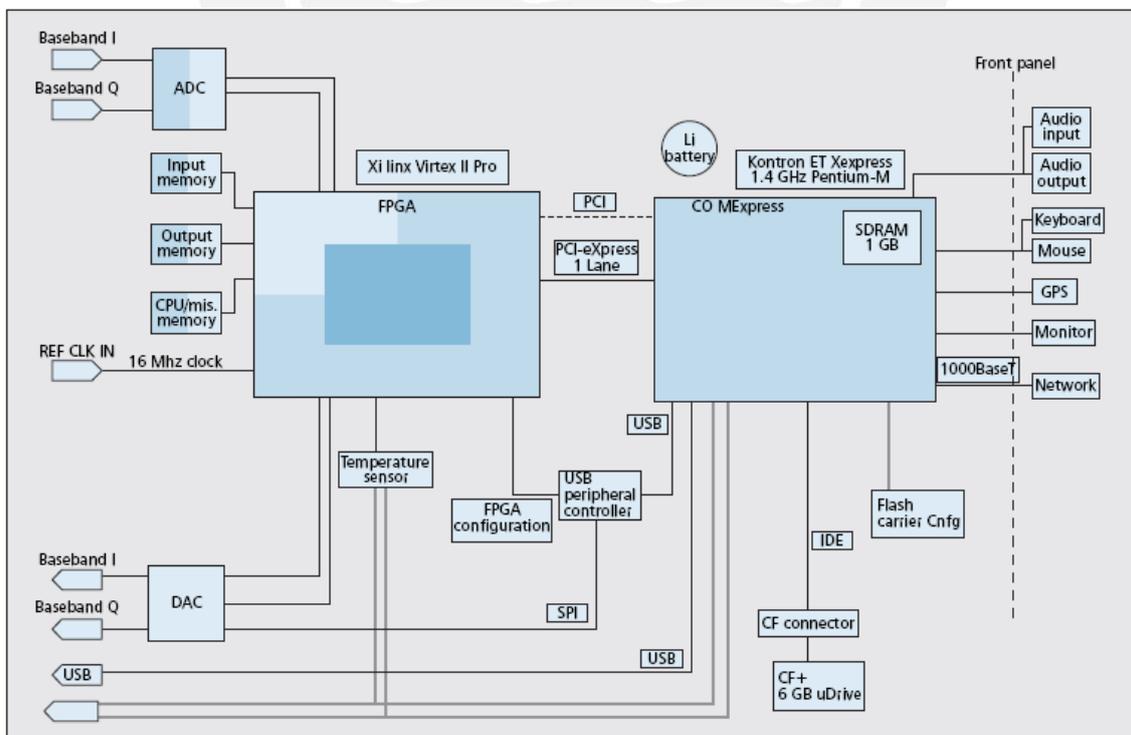


Figura 4. Tarjeta Digital del dispositivo KUAR. [7]

Implementaciones como el KUAR evidencian la complejidad de diseño que conllevan los sistemas SDR. La exquisita mezcla de distintas tecnologías contenida en este dispositivo – el cual se observa en la Figura 4 – es una muestra clara de cómo se complementan una con la otra. Para su implementación, se utilizó un procesador de control Kontron ET Express de 1.4 GHz, una tarjeta de desarrollo de Xilinx en donde se encontraba el FPGA Virtex II-Pro, y ASICs tanto para los ADCs (Analog to Digital Converters o Convertidores Análogo-Digital) y DACs (Digital to Analog Converters o Convertidores Digital-Análogo) como para los circuitos del Front-End de RF [7]. El procesador de control sirve como plataforma de configuración, y sobre él se pueden ejecutar sistemas operativos, aplicativos, y todo lo que esto conlleva: llamada a librerías, manejo de drivers, comunicación con interfaces, y más operaciones de esa índole. Por otro lado, el FPGA solamente se ocupa de realizar los algoritmos de modulación en función de la información otorgada por el procesador de control a través del bus PCI Express. El FPGA sirve como interfaz de procesamiento IF (Intermediate Frequency) o banda intermedia, pues éste está conectado tanto a un ADC, el cual le entrega las señales del receptor para ser procesadas, como a un DAC, a través del cual envía la señal para ser trasladada a alta frecuencia por el transmisor [7]. La velocidad y eficacia del procesador para ejecutar instrucciones secuenciales lo hacen idóneo para el rol de plataforma de software y unidad de control, mientras que el usar al FPGA para los algoritmos de procesamiento de señal y traslado de señal a banda intermedia resulta muy beneficioso debido al gran nivel de paralelismo y baja latencia que éste presenta.

## CAPÍTULO 2: TRANSMISORES OFDM

### 2.1. Modulación OFDM.

Dado que la tecnología SDR apunta solo a comunicaciones inalámbricas, no todas las técnicas de modulación deben ser analizadas. Sin embargo, algoritmos como OFDM (Orthogonal Frequency Division Multiplexing o Multiplexación Ortogonal por División de Frecuencias) o CDMA (Code Division Multiple Access o Múltiple Acceso por División de Código) si serían una valiosa herramienta para esta tecnología.

La técnica de modulación OFDM tiene muchas definiciones, pues ha sido un estándar que ya tiene gran tiempo de existencia, y ha sido adoptado por distintas tecnologías de telecomunicaciones. Sin embargo, una de las más precisas y adecuadas es la siguiente: “Técnica de modulación multi-portadora en la cual determinados símbolos de datos modulan una colección paralela de sub-portadoras regularmente espaciadas. Estas sub-portadoras deben tener una separación mínima de frecuencia para así mantener la ortogonalidad de sus correspondientes formas de onda en el dominio del tiempo. Sin embargo, sí existe traslape entre los espectros correspondientes a cada sub-portadora” [8]. En otras palabras, el esquema de modulación OFDM implica dividir la trama principal en varias sub-portadoras, las cuales serán mapeadas a constelaciones ya preestablecidas; así, el resultado final es la suma de la parte real (coseno) y parte imaginaria (seno) de todas las sub-portadoras. He ahí el porque es una modulación ortogonal. De esta técnica de modulación han aparecido algunas implementaciones que siguen los mismos principios básicos, pero que presentan ciertas particularidades. A continuación se muestra una clasificación de esta técnica de modulación:

- **OFDM:** Esquema tradicional de modulación ortogonal como aquel descrito en la definición.
- **COFDM:** Mejora del algoritmo de modulación en cuanto a robustez frente al ruido en base a técnicas de codificación y entrelazado (interleaving).
- **NC-OFDM:** Este esquema difiere del tradicional únicamente en el posicionamiento de las portadoras. No hay necesidad que se encuentren adyacentes y traslapándose una a la otra. He ahí el nombre NC (Non Contiguous o no contiguas) [7].

La modulación OFDM ha cobrado gran apogeo para la realización de sistemas de comunicaciones inalámbricos. Debido a que permite un uso más eficaz del espectro con respecto a las modulaciones FDM y FDMA, y otorga además una mayor resistencia contra el

ruido del canal, este algoritmo se ha convertido en el favorito para numerosos y variados servicios inalámbricos: WiFi, Wimax, TDT (Terrestrial Digital Television o Televisión Digital Terrestre), Televisión para dispositivos móviles y celulares (DVB-H) y muchos más. Es así que el lograr adaptar este esquema de modulación a la tecnología SDR abriría una gran cantidad de oportunidades para obtener un mejor control sobre el servicio; lo que se derivaría en un aumento de la calidad del mismo. Quizás, la principal aplicación se daría para los nodos de las redes Wimax, pues esta tecnología ya cuenta con esquemas de modulación adaptativa. La tecnología Wimax permite dos tipos de transmisiones: LOS (Line of Sight o con Línea de Vista) y NLOS (Non Line of Sight o sin Línea de Vista), por lo que necesita de esquemas de modulación adaptativa para contrarrestar las condiciones precarias del segundo tipo de transmisión.

Debido a que el principio básico de la modulación OFDM es el paso de las señales del dominio del tiempo hacia la frecuencia (obtener la parte real e imaginaria de cada sub-portadora), los algoritmos de la Transformada de Fourier en dominio discreto han sido la herramienta principal de la gran mayoría de diseñadores.

#### 2.1.1. Transformada de Fourier: DFT y FFT.

La Transformada de Fourier (llamada así por su descubridor, el matemático francés Jean-Baptiste Fourier) es un procedimiento matemático que se emplea para cambiar una función del dominio del tiempo al dominio de la frecuencia con el fin de obtener en frecuencia información valiosa que no es fácilmente apreciada en el tiempo.

La Transformada de Fourier se aplica sobre funciones continuas, es decir, las funciones que son definidas en todos los valores del tiempo  $t$ . Estas funciones, por ejemplo, podrían representar señales analógicas de voltaje que continuamente varían y que son producidas por un micrófono u otro tipo de transductor.

Sin embargo, el procesamiento digital de señales involucra señales discretas (señales que son muestreadas en determinados intervalos de tiempo) más que señales continuas. Una forma modificada de la Transformada de Fourier es la Transformada Discreta de Fourier o DFT (Discrete Fourier Transform) que es usada en señales discretas. Aquí, el espectro de una señal se refiere al modo en que la energía en la señal es distribuida sobre sus varios componentes de frecuencia.

Cuando la DFT es aplicada a una señal discreta, el resultado es un juego de coeficientes seno y coseno. Cuando las ondas seno y coseno de frecuencias apropiadas son multiplicadas por estos

coeficientes y luego sumadas, la señal original es reconstruida. Las ondas seno y coseno son componentes de frecuencia de la señal original, en el sentido de que la señal puede ser reconstruida a partir de estos componentes. Los coeficientes determinados por la DFT representan las amplitudes de dichas componentes.

En la práctica, el procedimiento por el cual los coeficientes seno y coseno son calculados requieren mucho cómputo. Para determinar cada coeficiente, cada uno de los valores muestreados de la señal debe ser multiplicado por su correspondiente valor en la onda seno o coseno de su respectiva frecuencia. Luego, estos productos deben ser sumados y el resultado dividido entre el número de muestras involucradas para dar el valor del coeficiente.

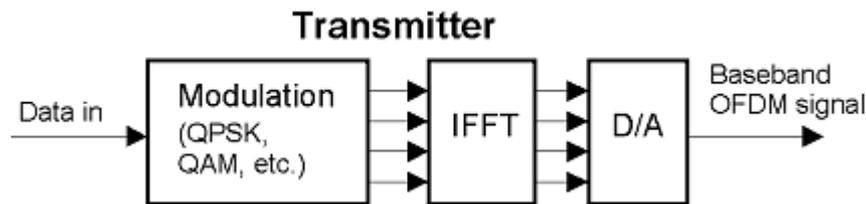
Si la señal consiste en un número de muestras  $N$ , la DFT requiere el cálculo de  $N$  coeficientes seno y  $N$  coeficientes coseno. Para determinar cada coeficiente,  $N$  productos de las muestras de la señal y de la respectiva onda seno o coseno deben ser evaluadas y sumadas. El número total de pasos en el cómputo de la DFT es  $N^2$ , cada paso requiere de la evaluación de la función seno y coseno junto con la multiplicación (y no incluye el cálculo de los  $N$  productos para encontrar cada coeficiente). Por ello, la DFT tiene un procedimiento numérico muy extenso. [9]

Aunque la información que proporciona la DFT sobre los componentes de frecuencia de una señal sea sumamente valiosa, el gran esfuerzo computacional involucrado hasta los sesentas hacía que la DFT sea una aplicación poco usada. Dos avances importantes cambiaron la situación completamente. El primero fue el desarrollo del computador digital con su capacidad de realizar cálculos numéricos de forma rápida y exacta. El segundo fue el planteamiento de Cooley y Tukey (originalmente descubierto por Gauss [10]), el cual presenta un algoritmo numérico que permite evaluar la DFT con una reducción significativa en la cantidad de cálculos requeridos. Este algoritmo, llamado Transformada Rápida de Fourier, o FFT, permite que la DFT de una señal muestreada sea obtenida de manera rápida y eficiente.

## 2.2. Etapas de un transmisor OFDM.

Un transmisor cualquiera siempre se divide en 3 etapas: la etapa de banda base, la etapa de banda intermedia, y finalmente la etapa de alta frecuencia o RF. Sin embargo, las dos últimas etapas son prácticamente iguales en todo transmisor, por lo que solo este trabajo solamente se enfocará en la primera de ellas: la etapa de banda base o específicamente la unidad funcional conocida como modulador.

Como se explicó anteriormente, la modulación OFDM divide la trama principal en varias sub-portadoras, cada una de las cuales será mapeada a una constelación o esquema de modulación digital [8]. A la salida de este bloque se obtiene la información de cada sub-portadora en el dominio de la frecuencia, por lo que se debe pasar esa información al dominio del tiempo. En consecuencia, la operación que se debe realizar con cada sub-canal es la IFFT (Inverse FFT o FFT Inversa) para así obtener la información requerida, lo que se puede apreciar en la figura 3.



**Figura 5. Diagrama de Bloques de un Transmisor OFDM.**

Luego de haber realizado el cambio de dominio de las señales, se debe realizar una transformación más. Es claro que todo lo realizado se ejecuta sobre sistemas digitales, pues esto facilita el control y la variación de los parámetros de modulación. Sin embargo, la señal requerida es una señal analógica por lo que se debe realizar una conversión digital-análoga. Como es normal en toda conversión de este tipo, la simple operación induce una cierta cantidad de ruido a la señal generada. Así, siempre se busca contar con convertidores de alta resolución para aminorar esta presencia de espurias. No obstante, esto también influye de manera negativa, pues cuan mayor es la resolución del convertidor, mayor será el costo del dispositivo final.

### 2.3. Técnicas de implementación de moduladores OFDM.

#### 2.3.1. Uso de DSP (Digital Signal Processor).

Transceptores OFDM para la tecnología SDR han sido el principal objetivo de los diseñadores. Sin embargo, para su implementación, la mayoría se ha basado en el flujo de diseño tradicional de transceptores OFDM de hardware específico; es decir, en la utilización de DSPs para la ejecución del algoritmo de la DFT (Discrete Fourier Transform o Transformada Discreta de Fourier) o FFT (Fast Fourier Transform o Transformada Rápida de Fourier) y sus respectivas inversas; y su respectivo Front-End de RF [11][12]. El uso común de DSPs se debe a que las tasas de transferencia de las señales a modular solía ser muy pequeño – no superaba los 10

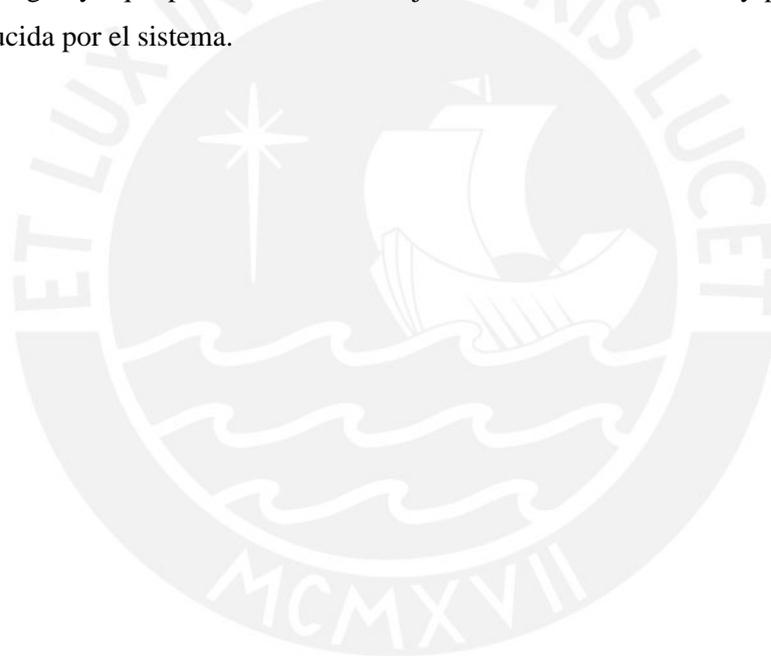
Mbps – y estos sistemas tenían una excelente performance a estas bajas velocidades. En consecuencia, la primera limitación que se hace notar en el uso de dispositivos que ejecutan instrucciones secuenciales es que inducen una latencia considerable si se les encarga la realización de complejos algoritmos de procesamiento de señal con tasas de transferencia altas. Sin embargo, esto no ha impedido a muchos diseñadores a usar DSPs en la realización de SDRs, como es el caso del dispositivo implementado en el Georgia Institute of Technology, en el año 2004. Esta “testbed” o plataforma de prueba contiene 2 sets de transmisores y receptores y está pensada para funcionar bajo el estándar IEEE 802.11a. Cabe resaltar, que a pesar de que para este dispositivo se usó un DSP que funcionaba a una alta frecuencia – de hasta 200 Mhz – no se logró alcanzar grandes velocidades. El sistema funciona con un máximo ratio de bit de 30 Mbps para la señal de entrada [13]. Esto es un resultado aceptable, mas no el óptimo, pues la proyección de los SDRs es un dispositivo que logré procesar señales con ratios de bit compatibles con las tecnologías que ya existen para comunicaciones alambradas: 10 Mbps, 100 Mbps y 1000 Mbps.

### 2.3.2. Uso de DUC (Digital Up-Converter).

Es usual que para poder unir las etapas de RF y de banda base de sistemas como los transmisores OFDM se usen DUC (Digital Up Converter) como interfaz de IF [13]. Esto se debe a que los DSPs no pueden alcanzar tasas de transferencia elevadas, lo que delimita también el rango de frecuencias que se pueden generar con la adición de un DAC. El mandar señales ubicadas completamente en banda base hacia la etapa de RF puede complicar el diseño del sistema en gran medida. Los DUCs presentan una excelente alternativa de solución para este inconveniente. No obstante, también podrían ser utilizados en reemplazo de la etapa de IFFT de los moduladores OFDM, pues una señal OFDM no es más que una suma de senos y cosenos. Ya que estos contienen en su interior DDSs (Direct Digital Synthesizers o Sintetizadores Digitales Directos), se logra conseguir una perfecta cuadratura entre las señales ortogonales [14]. De esa manera se ahorra procesamiento, hardware y tiempo. Sin embargo, para su implementación y correcto funcionamiento se requeriría un dispositivo que pueda funcionar en los rangos de frecuencia apropiados. Es así que los FPGAs se presentan como una buena alternativa para la implementación de estos sistemas.

Los FPGAs brindan un gran nivel de paralelismo y rápida reconfiguración, lo cual los convierte en herramientas apropiadas para el diseño de SDRs. Es por eso que la mayoría de las

implementaciones actuales en esta tecnología los toman como las herramientas de facto. El rol principal que se les da es el de un DSP dedicado, con unidades funcionales específicas, que agilizan la ejecución de los complicados algoritmos de procesamiento de señal [15][7]. Así también, como fue mencionado anteriormente, sirven también para trasladar la señal en banda base a banda intermedia. En consecuencia, son comunes también las implementaciones de DUCs sobre FPGAs, con fines de reducción de costos. Cabe resaltar sin embargo, que a pesar de que tanto los bloques de DSP como los del DUC se suelen implementar sobre el mismo FPGA, no ha habido muchos trabajos que fusionen ambas unidades funcionales para obtener un DSP robusto y que logre moverse en frecuencias más elevadas. Del mismo modo, tampoco existen muchas implementaciones que traten de eliminar el bloque de IFFT de los moduladores OFDM y reemplazarlos con un DUC y para lograr mayor rapidez de procesamiento. El elaborar módulos que integren ambas unidades funcionales con eficiencia significará un gran avance para esta tecnología, ya que permitirá utilizar mejor los recursos existentes, y permitirá reducir la latencia inducida por el sistema.



### CAPÍTULO 3: PLANTEAMIENTOS PARA LA IMPLEMENTACIÓN DE EL MODULADOR OFDM RECONFIGURABLE

#### 3.1. Modelo Teórico

La tecnología SDR busca difundir el diseño de radios reconfigurables mediante una interfaz gráfica para así tener un completo y remoto control del dispositivo a un mínimo costo y esfuerzo. El diseño de estos mencionados radios involucra el uso de hardware altamente reconfigurable y veloz. En consecuencia, el análisis o el enfoque de diseño de cualquier SDR difieren del flujo de la metodología de diseño de un dispositivo que funcionará como hardware dedicado. Es así que, el hardware a diseñar debe contar con 3 características principales: velocidad, reconfigurabilidad y alta calidad de señal.

Como se ha mencionado antes, la técnica de modulación OFDM es uno de los principales objetivos para esta tecnología. Debido a que OFDM constituye una modulación ortogonal, su resultado estará en función de señales sinusoidales (senos y cosenos), por lo que la meta fundamental en el diseño de moduladores OFDM reconfigurables es el poder generar este tipo de señales en base a unidades funcionales digitales. Esto debido a que el hecho de utilizar bloques de circuitos digitales otorga un mayor control sobre todo el sistema.

Teniendo en cuenta todos estos objetivos, el panorama del flujo de diseño de un modulador OFDM reconfigurable para un futuro SDR se torna mucho más claro. En primer lugar, es necesario tener en cuenta las necesidades del usuario del dispositivo. En otras palabras, en función al requerimiento del usuario se deben establecer las características preliminares del SDR; por mencionar algunas, tenemos el valor de la radio frecuencia, el ancho de banda a utilizar, y otras más. Sin embargo, todas estas estipulaciones se deben consultar y validar con las entidades normativas y reguladoras del uso del espectro, pues al fin y al cabo, el espectro de frecuencias es un recurso del cual dispone la sociedad. Es así que se debe tramitar el uso de las frecuencias y del ancho de banda con las entidades correspondientes antes de iniciar el diseño del SDR. Así también, es necesario tomar en consideración el ambiente en el cual funcionará el modulador solicitado. Es decir, se debe evaluar las condiciones del canal de transmisión para así escoger las características apropiadas del SDR. Obviamente, el SDR debe ser capaz de cambiar estas características en función de los cambios del canal de transmisión; no obstante, es necesario analizar las condiciones iniciales y la proyección de las posibles variaciones de estas

para poder dar al SDR las características necesarias para poder funcionar sin problemas en el medio requerido.

Por otro lado, otro punto importante en el diseño del modulador OFDM es la tecnología sobre la cual se va a implementar. Esto depende también en gran parte sobre las necesidades del usuario (tamaño del dispositivo, consumo de potencia y otras de ese estilo). Del mismo modo, un punto importante a evaluar es la gama de herramientas que se encuentra a disposición del equipo de diseñadores, pues ella también puede constituir un limitante en el proceso de implementación. Sin embargo, cabe resaltar que la principal meta de la tecnología SDR es funcionar sobre dispositivos móviles, por lo que portabilidad y bajo consumo de área son factores claves para su desarrollo. En consecuencia, el diseño de circuitos integrados (ICs) y el uso de FPGAs son las alternativas más viables a disposición. El uso de ICs o FPGAs involucra un grupo diverso de herramientas de diseño asistido por computadora (CAD: Computer-Aided Design), las cuales permiten un mejor y más rápido diseño del circuito, así como también proveen herramientas de simulación que comprueban el buen funcionamiento del mismo.

El diseño del circuito de lógica digital es una de las partes principales de cualquier modulador reconfigurable, y por ende también de cualquier SDR. En consecuencia, es necesario el uso de HDLs para agilizar el flujo del diseño de esta etapa. Sea el FPGA la plataforma final o no, el uso de lenguajes como el VHDL son esenciales para la obtención de sistemas eficaces y robustos. Además, el uso de FPGAs conlleva muchas prestaciones que lo hace idóneo para SDRs, siendo las principales la portabilidad y un gran nivel de paralelismo.

Uniendo los dos factores mencionados (herramientas disponibles y las características del sistema) se puede proceder a plasmar el modulador OFDM deseado, lo cual se reduce a dos procesos principales: la selección de la plataforma de implementación y traducir las características requeridas a instrucciones que el sistema pueda interpretar mediante la configuración de sus unidades funcionales, tales como los DDSs. El resultado es un sistema eficaz y veloz, el cual entrega una señal de alta calidad (bajo ruido).

Finalmente, es importante resaltar que estamos lidiando con una tecnología que aún está en sus primeras etapas de desarrollo, lo cual nos proporciona una buena proyección a futuro y nos asegura la existencia de oportunidades para ahondar en la investigación y mayores opciones para el desarrollo de proyectos orientados a circuitos para comunicaciones (por ejemplo, la elaboración de un SOC), tomando como base lo expuesto en el presente trabajo.

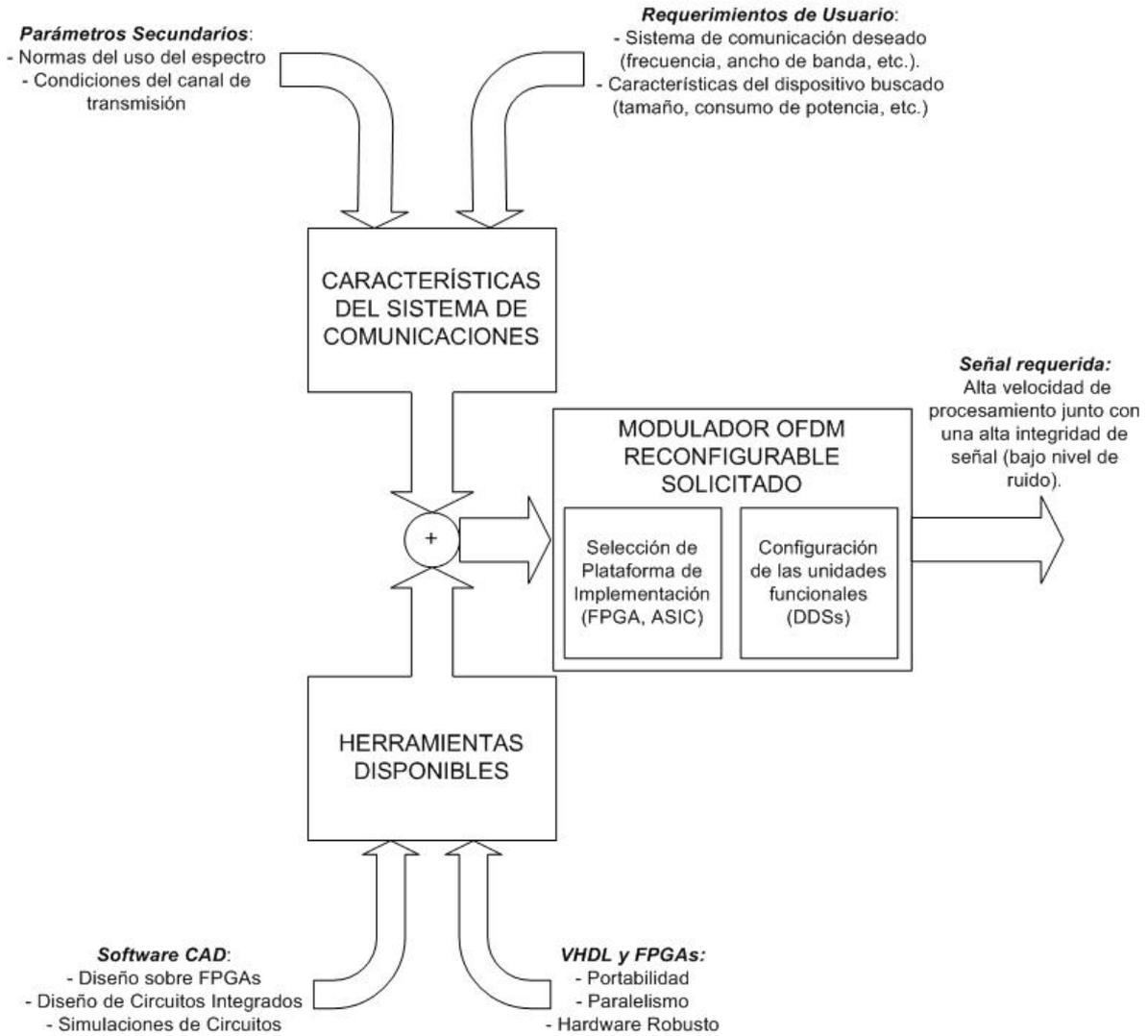


Figura 6. Representación Gráfica del Modelo Teórico.

### 3.2. Definiciones Operativas

El presente diseño de un modulador OFDM reconfigurable en un FPGA debe ser evaluado para medir su desempeño y verificar su funcionamiento. Así, los criterios más importantes son los referidos a la eficacia del sistema y a la arquitectura planteada.

#### ➤ Rango de Frecuencias en Banda Intermedia (Rango IF)

Indica los límites, tanto inferior como superior, del espectro de frecuencias que alcanza este modulador. Este es un indicador importante pues da una idea del alcance y versatilidad del

sistema para moverse en el espectro de frecuencias. Así, se puede evaluar la reconfigurabilidad del sistema, la cual es un factor clave para sistemas SDR. Un buen Rango IF sería desde 0 hasta 50 MHz.

➤ **Rango Dinámico Libre de Espurias (SFDR: Spurious Free Dynamic Range)**

Se define como la relación entre la potencia de la componente espectral deseada con respecto al nivel más alto de las componentes armónicas no deseadas. Este indicador es muy similar al SNR (Relación Señal a Ruido), pero constituye un mejor indicador de la pureza de la señal generada. Se expresa en dBc (decibelios con relación a la portadora) o en dB. Un valor apropiado de SFDR sería de 60 dBc.

➤ **Recursos utilizados**

Corresponde a la cantidad de compuertas, flip-flops, multiplicadores, LUTs (look-up tables o tablas de búsqueda), memorias, elementos lógicos (recursos lógicos) utilizados por el diseño. Es una forma aproximada de medir el área ocupada en el FPGA. Según este parámetro escogeremos al FPGA. En el diseño se procurará que sea el mínimo posible para reducir el área utilizada.

➤ **Cantidad de etapas del sistema**

Indica el número de etapas de segmentación utilizadas en el modulador OFDM planteado, así como el número de etapas de carga (load stage) de datos que tendrá toda la arquitectura. Es de importancia porque es un indicador de la complejidad de la arquitectura, así como también del tiempo de procesamiento promedio que ocupará para modular una señal, expresado en función de la frecuencia de la señal de reloj (ciclos de reloj).

➤ **Frecuencia de trabajo**

Es la frecuencia a la cual va a trabajar el sistema diseñado. Es un indicador de la rapidez del sistema. A mayor frecuencia, se tendrá mayor velocidad en el procesamiento. Sin embargo, se debe tener en cuenta también las limitaciones existentes en los dispositivos así como en el diseño mismo.

➤ **Tiempo de respuesta del sistema**

Indica el tiempo que se demora el sistema en cumplir con la labor asignada y otorgar un valor válido a la salida. Debe considerarse, además del tiempo calculado con la cantidad de

etapas y la frecuencia de reloj, un promedio de la latencia que el circuito induce. Se debe procurar que sea el mínimo posible. También su posible valor dependerá de la aplicación del sistema.

➤ **Portabilidad del diseño**

Indica si el diseño realizado con VHDL es transparente o no a los diferentes FPGA's. Esto quiere decir que un código será portable si puede ser sintetizado en cualquier FPGA. Significa que es independiente del dispositivo, pues no ha utilizado recursos propios del FPGA, los cuales pueden variar entre las diferentes familias y fabricantes. De esta forma se logra que el sistema pueda ser implementado en cualquier FPGA que cumpla con los requerimientos mínimos del mismo sistema.

### 3.3. Hipótesis de la Investigación.

#### 3.3.1. Hipótesis Principal.

Dado que actualmente las telecomunicaciones siguen la tendencia de la tecnología SDR, la cual de conseguir un hardware genérico que permita variar sus parámetros y amoldar sus características para que sea posible usarlo en cualquier medio de comunicación; entonces, el diseño de un modulador OFDM reconfigurable sobre un FPGA permite el profundizar en la investigación de esta tecnología y constituye una alternativa viable para el diseño de los transmisores en los cuales está presente esta técnica de modulación.

#### 3.3.2. Hipótesis Secundarias.

- 1) A pesar que la tecnología SDR están aún en desarrollo y quizá a una distancia considerable del momento en que ya pueda desplegarse sobre un sistema de comunicaciones real, el ahondar y persistir en la investigación y esfuerzos en pro de su desarrollo logrará que esa distancia se acorte y se pueda brindar una mejor calidad de servicio a la sociedad.
- 2) La técnica de modulación OFDM seguirá presente en la mayoría de sistemas de comunicaciones inalámbricas, por lo que la implementación de bloques configurables OFDM será una de las metas principales para lograr unificar distintos servicios en un solo

dispositivo; he ahí la importancia de la implementación del modulador OFDM reconfigurable.

- 3) El uso de un FPGA para la implementación del modulador OFDM reconfigurable presenta mayores beneficios con respecto al uso de un DSP. Gracias a sus características, el FPGA permite alcanzar mayores velocidades de procesamiento – gracias al paralelismo que otorga el dispositivo – y mayores rangos de frecuencia.
- 4) El FPGA otorga la posibilidad de expandir el diseño sobre sí mismo o integrar todo un sistema en un ASIC como diseño posterior. En otras palabras, otorga un gran nivel de escalabilidad.

### 3.4. Objetivos de la Investigación.

#### 3.4.1. Objetivo General.

Diseñar e implementar un modulador OFDM reconfigurable sobre un FPGA y comprobar su funcionamiento mediante una etapa de pruebas de integridad de señal (SFDR), versatilidad en cuanto al uso del espectro (Rango IF), y velocidad de procesamiento (latencia).

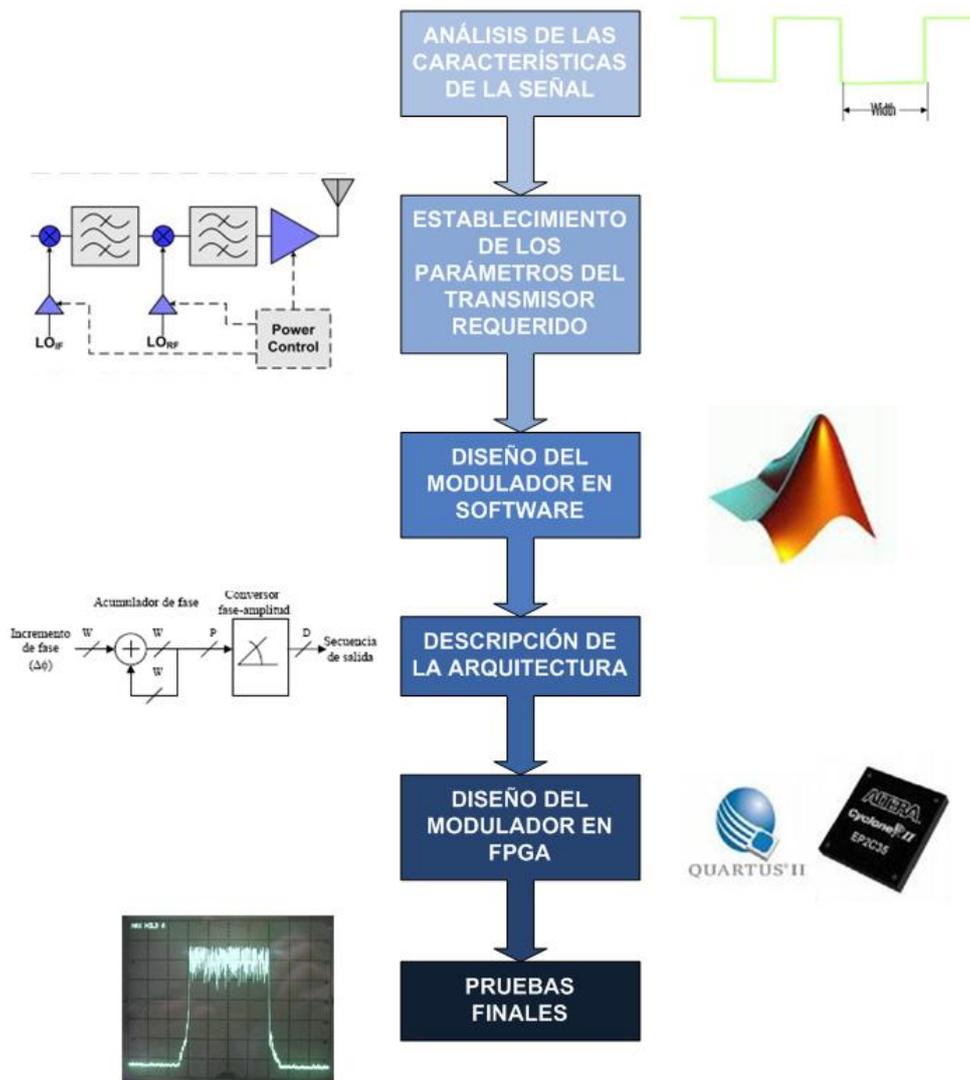
#### 3.4.2. Objetivos Específicos.

- 1) Diseñar el modulador OFDM reconfigurable en alguna herramienta de software como lo es el entorno MATLAB, para así lograr una mayor comprensión de su funcionamiento y poder comprobar su correcto funcionamiento.
- 2) Verificar el buen funcionamiento del dispositivo mediante los rangos especificados para cada factor de evaluación.
- 3) Diseñar las principales unidades funcionales del sistema (DDS, Mapeador de constelaciones digitales, Convertidor Serial/Paralelo) a nivel de hardware y plasmarlas sobre el FPGA como etapa previa al modulador OFDM, y realizar la correspondiente etapa de verificación.

- 4) Diseñar el modulador OFDM reconfigurable e implementarlo sobre el FPGA, y evaluar los resultados obtenidos en base a los parámetros establecidos anteriormente.
- 5) Realizar el diseño siempre tratando de conseguir un uso eficiente de los recursos del FPGA y bloques de hardware modulares y portables.

### 3.5. Metodología de la Investigación.

El diseño de la arquitectura se desarrollará siguiendo el esquema presentado a continuación, el cual ilustra la secuencia de pasos a seguir para la obtención del sistema.



**Figura 7. Metodología de Diseño.**

La primera etapa del flujo de diseño es analizar las características de la señal de entrada, pues en base a ella se procederá diseñar el modulador buscado. Es así que se debe tener en cuenta los niveles de voltaje de la señal, ancho de pulso y la tasa de transmisión (throughput). De este modo se tiene un indicador de la velocidad de procesamiento requerida por la arquitectura a diseñar.

La siguiente fase en el diseño del modulador es identificar los parámetros del transmisor requerido, siendo los principales la frecuencia de transmisión o rango de frecuencias para el caso de un sistema SDR, ancho de banda, cantidad de portadoras, rango dinámico de la señal modulada e inclusive la potencia de transmisión.

Una vez establecidos todos los parámetros de la señal de entrada y del modulador se tiene todo lo necesario para proceder a la etapa de diseño. Debido a que se trata de un modulador OFDM reconfigurable, la arquitectura se reduce a la implementación de un determinado número de moduladores pasa-banda, por lo que será necesario manejar distintas constelaciones digitales: QPSK, QAM-16 y QAM-64 por mencionar algunas. Por conocimiento general, la implementación de un modulador OFDM cualquiera se resume en un módulo que ejecute la IFFT, lo cual sucede principalmente debido a que la modulación OFDM consiste en colocar a las sub-portadoras en distintos lugares del espectro de frecuencias. Es decir, el procesamiento de la señal se da únicamente en la frecuencia, por lo que luego de procesarla se debe retornar al dominio del tiempo. Analizando el proceso de manera matemática y en el dominio discreto se resume de manera genérica en el algoritmo de la DFT, la cual se define de la siguiente manera:

$$X_p[k] = \sum_{n=0}^{N-1} x_p[n] e^{-j(2\pi/N)kn} \quad (2)$$

Donde  $N$  es la cantidad de bits que definen a la DFT y  $x_p$  es la representación discreta de la señal en el dominio del tiempo [16]. Partiendo de esta ecuación, se puede definir a la DFT inversa o IDFT como:

$$x_p = \frac{1}{N} \sum_{k=0}^{N-1} X_p[k] e^{j(2\pi/N)kn} \quad (3)$$

A partir de esta ecuación podemos definir una que nos permita describir el grupo de portadoras ortogonales. Cabe mencionar también que los miembros de un grupo de portadoras ortogonales

son linealmente independientes [16]. De esta manera, a partir de una secuencia de datos  $d_n$  ( $d_0, d_1, d_2, \dots, d_{N-1}$ ), en la cual cada dato  $d_n = a_n + jb_n$ , se define:

$$D_k = \sum_{n=0}^{N-1} d_n e^{-j(2\pi nk/N)} = \sum_{n=0}^{N-1} d_n e^{-j2\pi f_n t_n} \quad (4)$$

donde la frecuencia en el dominio discreto ( $f_n$ ) se define como  $f_n = n/(NDT)$  [16]. Del mismo modo, se define el tiempo en el dominio discreto como  $t_k = kDt$ , donde  $D$  es un símbolo aleatorio de la secuencia de datos  $d_n$ . De este modo, si solamente se toma la parte real de la variable  $D$ , se obtiene:

$$Y_k = \text{Re}\{D_k\} = \sum_{n=0}^{N-1} [a_n \cos(2\pi f_n t_k) + b_n \sin(2\pi f_n t_k)] \quad (5)$$

Por lo tanto, se puede definir de manera genérica a una señal OFDM como se indica en [16]:

$$y(t) = \sum_{n=0}^{N-1} [a_n \cos(2\pi f_n t_k) + b_n \sin(2\pi f_n t_k)] \Rightarrow 0 \leq t \leq N\Delta t \quad (6)$$

En esta última ecuación se puede observar que cualquier señal OFDM se puede describir mediante una suma de senos y cosenos, lo cual es clave para el planteamiento de la arquitectura del modulador. Así, lo que se propone en este trabajo de tesis es el uso de módulos DDS en reemplazo del bloque IFFT para la generación de las señales senos y cosenos que componen a la señal OFDM.

Teniendo en cuenta la cantidad de sub-portadoras con las que se quiere trabajar, y los antes establecidos parámetros de la señal de entrada y del transmisor, se puede dimensionar correctamente los bloques principales de la arquitectura: los DDSs, los mapeadores de constelaciones digitales y la interfaz serial-paralelo que recibe la señal de entrada. En adición a esto, se puede establecer la frecuencia de trabajo que gobernará todo el sistema. Del mismo modo, analizando todas las posibles tareas que pudieran ser requeridas del modulador OFDM, se procede a dimensionar el ancho de palabra de la instrucción y en consecuencia, el comportamiento de la Unidad de Control que manejará a todo el sistema.

Luego de finiquitar la descripción de la arquitectura del modulador, se procede a su implementación en el FPGA. La arquitectura planteada se describe mediante código VHDL (Very High Speed Integrated Circuit Design Hardware Description Language) y una herramienta CAD que automatiza y simplifica tanto la síntesis del circuito, como su posterior implementación en el dispositivo. Para abordar de manera exitosa esta labor es necesario tomar en cuenta los objetivos planteados en cuanto al diseño; de este modo se puede evaluar la eficacia del sistema en función de la portabilidad de su diseño o su optimización en cuanto a cantidad de hardware usado, o del mismo modo, si se necesita que se ajuste solamente a ciertos requerimientos particulares o si tiene proyectado un propósito de uso genérico. Sea cual sea el caso, es más eficiente diseñar cada una de las unidades funcionales por separado y llevar a cabo su verificación de manera individual antes de proceder a la unificación de todo el sistema. Adicionalmente, otro aspecto importante que no se debe dejar de lado es la obtención de un balance entre la cantidad de recursos utilizados y la frecuencia de trabajo escogida para el sistema.

Cabe resaltar que en primera instancia se implementó el modulador OFDM reconfigurable en software con el propósito de verificar su funcionamiento y poder realizar ajustes a la arquitectura planteada. Sin embargo, la implementación en hardware no fue un fiel seguimiento a los esquemas planteados en el software, pues debieron realizarse algunas modificaciones para poder obtener optimizaciones en cuanto a la rapidez de procesamiento y la cantidad de hardware usado.

### 3.6. Análisis del Sistema y Determinación de Requerimientos.

El modulador OFDM a diseñar será implementado para procesar señales digitales – tramas de bits – de hasta 100 Mbps pues es necesario que sea compatible con las tecnologías de comunicación actuales (Ethernet 100BaseT). Es decir, independientemente de lo que se esté transmitiendo (video, datos, etc.) el dispositivo está pensado para trabajar en una red de conmutación de paquetes, por lo que la proyección final es que sea accesible por red para su configuración.

La señal de entrada, como se mencionó anteriormente, será una trama de bits con una tasa de transferencia de 100 Mbps, y variará entre 0 y 5 voltios, que corresponden a '0' y '1' lógico respectivamente. Esto se cumplirá para toda señal que se requiera modular, independientemente

de la información guardada en la señal (audio, vídeo, dato, etc.). Además de la señal a modularse, el sistema necesita otras señales de entrada para que se reconfigure según lo requerido por el usuario. Así, otra importante señal de entrada es la Instrucción, la cual ingresa a la Unidad de Control y determina las características del modulador. Esta Instrucción consistirá en una palabra binaria que indicará únicamente la cantidad de portadoras a usar por el modulador, que repercute directamente con el ancho de banda ocupado por la señal modulada. Del mismo modo, esta Instrucción determinará la constelación digital a usarse para modular cada sub-portadora.

Otro parámetro importante que debe ser configurable es la frecuencia de las portadoras requeridas, por lo que debe existir una señal de entrada que provea de esta información a cada sub-canal. Debido a que este dispositivo es un prototipo de prueba, no es requerimiento contar con una gran cantidad de portadoras; en consecuencia, se escogió que el modulador OFDM a implementar podría usar como máximo 8 sub-portadoras para modular la señal de entrada. Es decir, se necesitarán 8 palabras digitales que se encarguen de sintonizar cada sub-canal del modulador.

Finalmente, lo que se tiene a la salida es la sucesión de palabras digitales, las cuales luego de pasar por una etapa de conversión del dominio digital al analógico, formarán la señal modulada. Esto implica el uso de un DAC, por lo que se debe definir el ancho de palabra de salida del dispositivo para así escoger apropiadamente el convertidor. Para dispositivos de comunicaciones se debe tener en cuenta la necesidad de obtener una señal con la menor cantidad de espurias, por lo que en estos casos se necesitan DACs de alta resolución. Para la implementación presentada en este trabajo se fijó el ancho de palabra de salida a 10 bits, por lo que a partir de eso se buscó el DAC apropiado. Cabe mencionar que se necesita un convertidor que trabaje a altas frecuencias pues se está trabajando con tramas de bits de altas tasas de transferencia. Es así que se escogió un DAC que reciba de entrada la señal de reloj con la que realizará las conversiones, para así sincronizar todo el dispositivo con el convertidor externo.

### 3.7. Consideraciones para el diseño.

El dispositivo implementado está pensado como una unidad funcional de un transmisor SDR, por lo que es necesario hacer hincapié en el hecho de que la única función del mismo es ejecutar el procesamiento digital requerido a la señal de entrada para realizar la modulación y llevar la

señal al rango de IF. La etapa de banda base, codificación de señal, y la etapa de RF no serán implementadas ni discutidas en el presente trabajo de tesis, pues se asume que la señal – sea datos, vídeo u otro – ya ha sido codificada y la etapa de RF se considera un trabajo posterior de investigación.

La etapa previa a la implementación es la simulación de la arquitectura mediante el software Matlab R2007 de Mathworks. De esta simulación también se obtiene la sucesión de datos digitales que a usar para formar las ondas sinusoidales dependiendo del ancho de palabra y el tamaño de la memoria a utilizar. Es así que si alguno de estos dos últimos parámetros mencionados anteriormente variara, se tendría que ejecutar la simulación nuevamente para obtener nuevamente el nuevo grupo de palabras digitales requeridas. Una vez simulada la arquitectura, se procederá a describir los módulos (archivos de extensión .vhd) dentro de la misma que se encargarán de guardar en memoria aquellas palabras digitales obtenidas de la simulación en software.

La simulación en software se seguirá realizando hasta obtener una señal con la mínima cantidad de ruido posible, siendo el indicador a analizar la Relación Señal a Ruido (SNR: Signal to Noise Ratio) o incluso el Rango Dinámico libre de Espurias (SDFR: Spurious Free Dynamic Range) pues es el factor más usado en sistemas de comunicaciones para evaluar la integridad de la señal. Ambos indicadores evalúan la relación entre la potencia de la señal deseada con respecto a la potencia de componentes ruidosas, pero el SFDR es más usado pues no toma en cuenta la potencia media de los espurias presentes, si no el máximo valor de potencia entregado por alguna componente ruidosa dentro de la señal entregada. Así, el valor mínimo valor de SDFR que se requiere es de 60 dBc, para de ese modo garantizar la calidad de la señal entregada.

El diseño en hardware se realizará a través de la descripción de la arquitectura planteada haciendo uso del lenguaje de descripción de hardware VHDL con la herramienta CAD Quartus II 8.0 de la compañía Altera. El dispositivo de lógica programable sobre el cual se plasmará el circuito será el FPGA Cyclone II distribuida por la compañía antes mencionada. Uno de los objetivos principales con respecto a la implementación en hardware es obtener una descripción modular y genérica, es decir, que no depende del FPGA usado para la implementación. Esto implica evitar siempre que se pueda el uso de bloques y circuitos embebidos dentro del FPGA utilizado pues son estos los que no se pueden trasladar de un dispositivo de lógica programable a otro. Sin embargo, en el análisis de la arquitectura se observó que en cierta etapa del modulador iba a ser necesaria la ejecución de multiplicaciones complicadas, operación que

siempre genera cierto retardo cuando no se optimiza su implementación. Fue este el motivo que obligó el uso de los bloques multiplicadores contenidos dentro del FPGA para así obtener la menor latencia posible de este complicado bloque. Esto puede aparentar en un inicio una disminución en el nivel de portabilidad del sistema implementado; sin embargo, cabe resaltar que la mayoría de FPGAs poseen bloques de multiplicadores embebidos para dar mayor eficacia al sistema y facilitar el trabajo del diseñador, por lo que esta falencia se puede contrarrestar infiriendo en el diseño los bloques de multiplicadores dependiendo del FPGA con el que se trabajó, acción fácil de realizar con la herramienta CAD Quartus II 8.0.

Finalmente, es necesario mencionar que siempre se debe sopesar la cantidad de recursos disponibles con la rapidez y eficiencia que se requiere del sistema a implementar. En otras palabras, se puede implementar un dispositivo muy veloz pero esto implicaría el uso de mayor cantidad de recursos del FPGA. Otro aspecto de este balance que es necesario tener en cuenta es la integridad de la señal entregada por el sistema en contraste con la cantidad de recursos necesarios y la latencia inducida por la adición de bloques memoria. Debido a estos motivos se debe tener criterio para poder lograr un compromiso adecuado en cuanto al uso de hardware con respecto a la rapidez del sistema y la calidad de señal que se quiere alcanzar, dependiendo del sistema de comunicación sobre el cual funcionará el dispositivo SDR.

## CAPÍTULO 4: DISEÑO DE LA ARQUITECTURA DEL MODULADOR OFDM RECONFIGURABLE EN EL FPGA CYCLONE II DE ALTERA

El presente capítulo pretende mostrar el proceso de diseño del modulador OFDM reconfigurable, el cual involucra la determinación de sus parámetros – tales como la cantidad de portadoras, la cantidad de bits de resolución de la señal entregada, etc. – y la arquitectura planteada para la implementación en hardware sobre el FPGA.

Según la metodología planteada, el siguiente paso a seguir es la implementación del modulador en software, principalmente con fines de verificación de funcionamiento. De esta manera, se obtiene también un mejor entendimiento del funcionamiento del sistema y se logra comprobar el correcto performance del mismo.

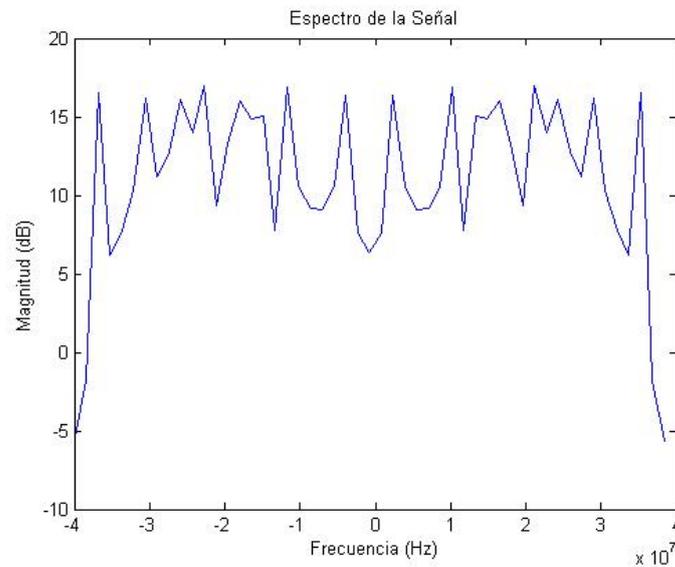
Posteriormente, luego de haber comprobado el funcionamiento del sistema en software, se procede a diseñar la arquitectura en función a bloques digitales y unidades funcionales modulares que sean capaces de interactuar de manera apropiada.

### 4.1. Diseño del modulador OFDM reconfigurable en software.

Para el diseño del modulador en software se utilizó el entorno de programación Matlab R2007 de la compañía Mathworks. El programa realizado no utilizó ninguna función de algún toolbox especial pues simplemente se planteó un algoritmo que siguiera la ecuación expresada en (5). Por ende, la implementación se resumió a la generación de ondas sinusoidales con distintas amplitudes dependiendo de la señal de entrada que se utilizara para realizar la prueba. De este modo, se pudo analizar la señal obtenida tanto en el dominio del tiempo como en el de la frecuencia.

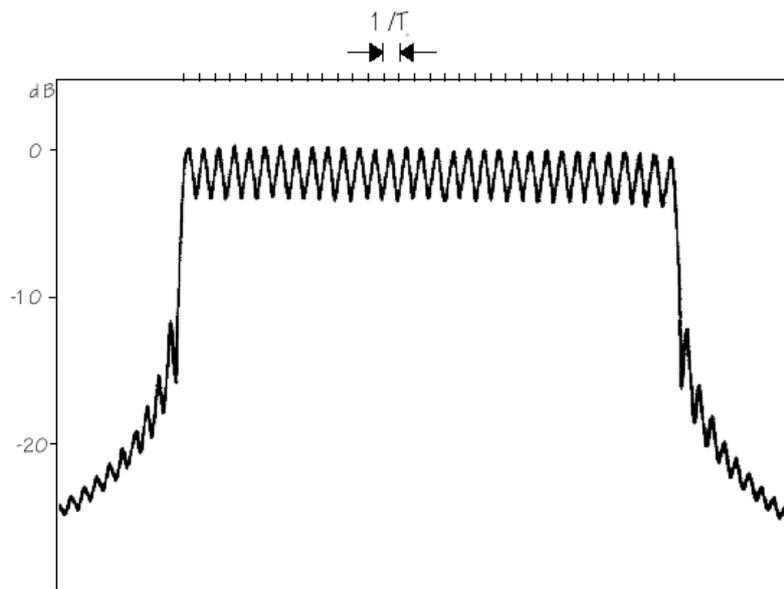
Para la simulación se escogió utilizar un modulador de 8 sub-portadoras (al igual que el que se planea implementar en hardware) y se trabajó en ambos ambientes del entorno de Matlab: mediante el uso de bloques en el Simulink, y mediante el uso de comandos en un script. Esto ocurrió debido a que ambos ambientes facilitaban ciertas operaciones a la señal. Así, el Simulink se utilizó para la generación de ondas sinusoidales de frecuencia, fase y amplitud variable y para la sumatoria de las mismas. Por el contrario, el procesamiento en frecuencia de

la señal y la visualización de la misma se realizó en un script. Los resultados obtenidos son los ilustrados en la Figura 8.



**Figura 8. Resultado de Simulación en Software.**

Se puede apreciar que los picos de cada sub-portadora no son perfectos, lo cual ocurre por el hecho de trabajar con señales sinusoidales con desfases y amplitudes variables. Por el contrario, de manera teórica y en un plano completamente ideal, una señal OFDM tendría todos los picos correspondientes a cada sub-portadora sin deformaciones, como se puede apreciar en la Figura 9.



**Figura 9. Espectro ideal de una señal OFDM con 32 sub-portadoras [15].**

Se puede apreciar que la señal obtenida en la simulación presenta cierta cantidad de ruido; sin embargo, se puede considerar como nula pues lo más importante es el posicionamiento en frecuencia de cada sub-portadora, mas no su amplitud ni su comportamiento en el tiempo. Por lo tanto, la simulación en software demostró que la generación de señales OFDM mediante la sumatoria de las apropiadas señales sinusoidales es una alternativa viable y eficiente.

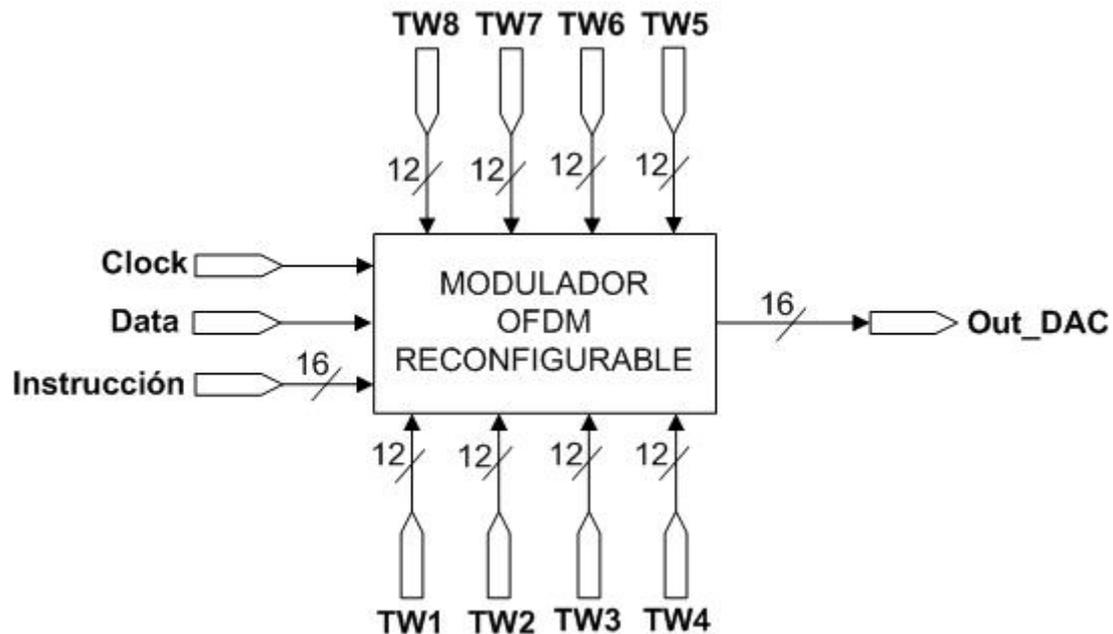
#### 4.2. Diseño del modulador OFDM reconfigurable sobre el FPGA Cyclone II de Altera.

Una vez que ya se tiene claro el funcionamiento del modulador OFDM reconfigurable, se procede a implementar la arquitectura planteada. La manera más eficiente de implementar cualquier sistema complejo es dividirlo en unidades funcionales, para así poder hacer un diseño modular y poder realizar el análisis de fallas y errores de manera más sistemática. Así, se debe tener en claro los principales bloques del sistema, diferenciados claramente por las funciones que realizan:

- El DDS, el cual se encarga de generar las señales seno y coseno con frecuencia variable.
- El Mapeador I&Q o Mapeador de Constelaciones Digitales, el cual se encarga de seleccionar, en función a la constelación digital escogida (QPSK, QAM-16 o QAM-64) las amplitudes de cada componente sinusoidal.
- El Convertidor Serial/Paralelo, el cual se encarga de generar las sub-portadoras en función de la trama de entrada.
- La Unidad de Control, la cual se encarga de manejar las demás unidades funcionales y establecer ciertos parámetros que condicionan el comportamiento de cada una de ellas.

Otro factor importante que se debe tomar en cuenta antes de iniciar el diseño de cualquier unidad funcional, es analizar las entradas y salidas de todo el sistema en general para tener una visión general de las señales que se necesitan para ejecutar ciertos procesos, y adicionalmente, para no perder perspectiva de cuáles son las señales de salida de todo el sistema, las cuales al final representan todo el objetivo de la implementación. Así, se analiza al sistema completo como una caja negra se tendría lo mostrado en la Figura 10. Como se puede apreciar la salida que entrega el sistema – la señal Out\_DAC – no es realmente la señal OFDM requerida; se necesita de una etapa de conversión del dominio digital al analógico. Sin embargo, al analizar la

salida del modulador planteado se tiene ya una idea de la selección del Convertidor Digital-Análogo (16 bits de resolución).



**Figura 10. Entradas y Salidas del Sistema**

Como se había mencionado antes, la señal de entrada o Data sería una trama serial de 100 Mbps. Se puede notar claramente que la señal de entrada posee una alta tasa de transferencia, por lo que la señal de reloj o Clock debe ser de una alta frecuencia para así permitir al sistema funcionar sin problemas. En función de la tasa de transferencia de Data, la mínima frecuencia de trabajo del sistema sería 200 MHz, valor que fue escogido luego de evaluar las características de los posibles DACs que estarían a disposición.

Por otro lado, se puede observar un grupo de entradas de la misma longitud de palabra (12 bits): las señales TWX ( $X = 1,2,3...8$ ). Cada una de estas señales cumple la función de sintonizar la frecuencia de cada sub-portadora del modulador. Finalmente, la señal de entrada Instrucción es aquella que le indica al sistema qué sub-canales van a estar activos, y además contiene la información sobre las constelaciones digitales a usar por cada sub-portadora.

Luego de analizar el funcionamiento general del sistema, se procede a diseñar la arquitectura de las principales unidades funcionales del sistema, para que al término de esa etapa, el diseño del Modulador OFDM Reconfigurable se resume simplemente a la unión e interacción de bloques preexistentes.

4.2.1. Descripción y diseño de la arquitectura del Sintetizador Digital Directo (DDS).

El Sintetizador Digital Directo es la unidad funcional principal de todo el modulador, pues en base a él se hace posible la generación de señales analógicas y la manipulación de la frecuencia y fase de las mismas. En consecuencia, se debe procurar diseñarla de manera que sea lo más parametrizable para que así sea posible el aumentar su resolución en frecuencia o amplitud, dependiendo de la cantidad de recursos disponibles.

Siguiendo la teoría expuesta anteriormente sobre las técnicas de Síntesis Digital Directa, se debe implementar los mostrados en la Figura 2: el Acumulador de Fase y el Convertidor Fase-Amplitud. Sin embargo, para este caso particular no se siguió el esquema genérico ilustrado en la figura antes mencionada, pues para este circuito se necesita generar tanto senos como cosenos, cosa que se puede lograr aprovechando un solo bloque DDS. A continuación, en la Figura 11, se muestra la arquitectura planteada para cualquier DDS a usar en todo el modulador:

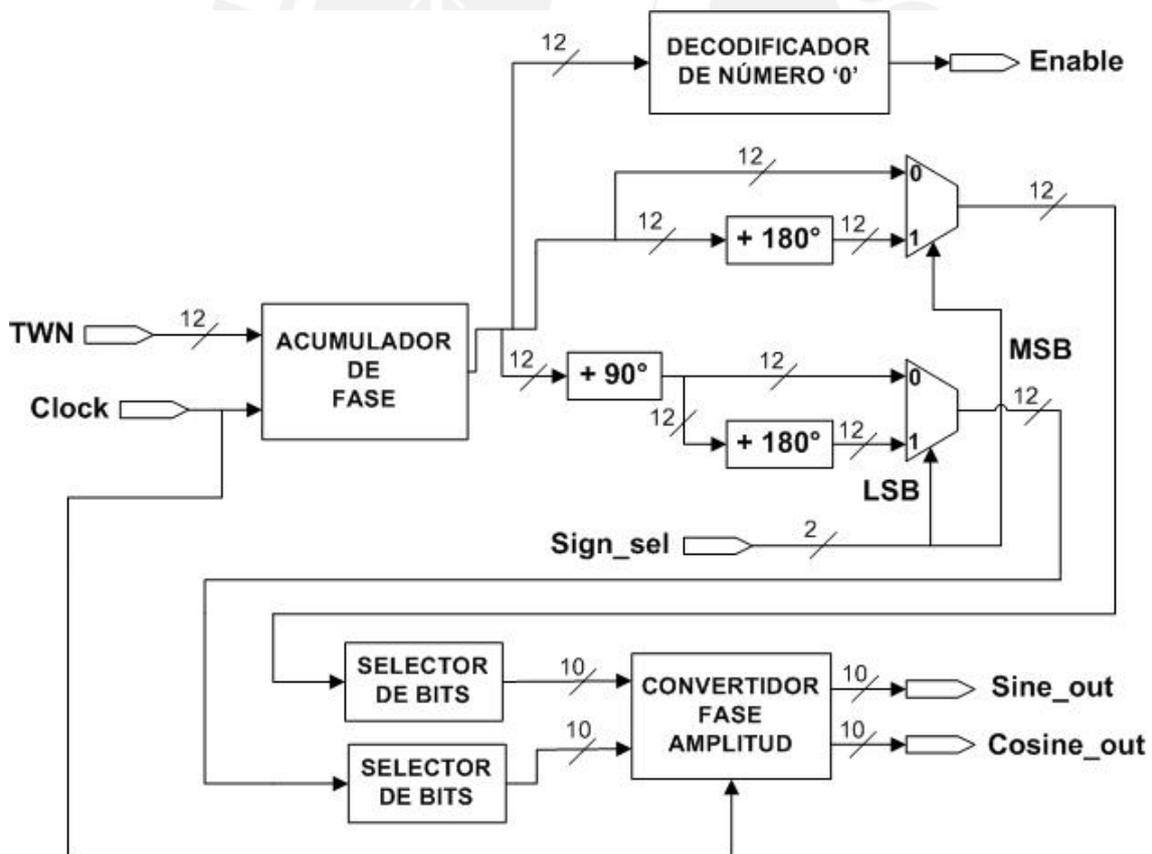


Figura 11. Diagrama de Bloques del DDS.

El Acumulador de Fase consiste básicamente en una rueda de fase que contiene una determinada cantidad de puntos que representan todo el período de una señal sinusoidal, es decir, representan los 360 grados de un círculo. Así, la señal TWN le indica al Acumulador de Fase qué tan rápido debe recorrer la rueda de fase, y de esa manera se puede variar la frecuencia de las señales a generar. La salida del Acumulador representa la fase de la señal sinusoidal, en particular de la función seno. Como es sabido, las funciones seno y coseno son ortogonales, es decir, se puede generar una a partir de la otra induciendo un desfase de 90 grados. Esto facilita aún más la tarea de diseño pues en función de la cantidad de puntos contenidos en la rueda de fase se puede calcular la cantidad de puntos en la rueda que representa algún desfase. Es así que de un solo Acumulador de Fase ya se tienen ambas señales sinusoidales.

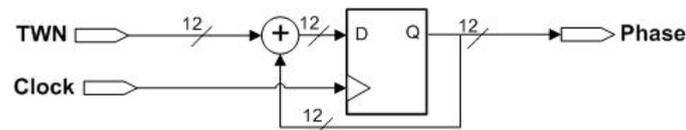
Se debe recordar que la mayoría de constelaciones digitales contiene coeficientes tanto positivos como negativos, por lo que en algún momento será necesario realizar una inversión de onda. Esto podría aparentar ser un problema pues ejecutar multiplicaciones entre enteros positivos mediante compuertas lógicas ya presenta cierta dificultad, y más aún si se trabaja con enteros positivos y negativos. No obstante, se puede evitar esa engorrosa operación aprovechando las prestaciones que da el uso de un DDS. Si se interpreta una inversión de onda como un desfase de 180 grados, se puede aprovechar nuevamente la presencia de la rueda de fase. Es por eso que ambas señales sinusoidales son bifurcadas, para así tener tanto la onda invertida y la onda original; luego, mediante un multiplexor se puede escoger cual de las dos ondas pasa a la siguiente etapa del circuito dependiendo de la constelación escogida y la data de entrada, información contenida en la señal Sign\_sel que proviene de otra importante unidad funcional: el Mapeador I&Q. Así mismo, hacia el Mapeador se envía una señal de habilitación o Enable a través del bloque Decodificador de Número '0', el cual evalúa cuando se ha completado la formación de una onda sinusoidal.

La etapa final del DDS es el Convertidor Fase-Amplitud que consiste básicamente en una LUT (Look-up Table) y contiene la secuencia de datos necesaria para formar las ondas seno y coseno. En otras palabras, es en esencia una memoria, por lo que se debe dimensionar de manera correcta pues el uso de memorias de gran tamaño induce latencia al circuito. He ahí la razón de ser del bloque Selector de Bit, el cual únicamente se encarga de tomar los 10 bits más significativos de la señal proveniente del multiplexor para así ajustar la señal a la dimensión de la memoria contenida en el Convertidor Fase-Amplitud. Finalmente, a la salida del mencionado bloque se obtiene las secuencias de datos que representan a las ondas seno y coseno en las señales de salida Sine\_Out y Cosine\_Out respectivamente.

#### 4.2.1.1. Acumulador de Fase.

El Acumulador de Fase, como se mencionó anteriormente, consta fundamentalmente de una rueda de fase, la cual está conformada por una determinada cantidad de puntos. El Acumulador se encarga de recorrer los puntos dentro de la rueda de fase mediante incrementos constantes, los cuales son establecidos a través de su principal señal de entrada: la señal TWN o Palabra de Sintonización (Tuning Word). Así, cuanto mayor sea este parámetro, más rápido terminará de recorrer la rueda de fase, lo que ocasionará que el DDS genere una señal de alta frecuencia. Por el contrario, a medida que el valor de TWN disminuye, también decaerá de manera proporcional la frecuencia de la señal generada.

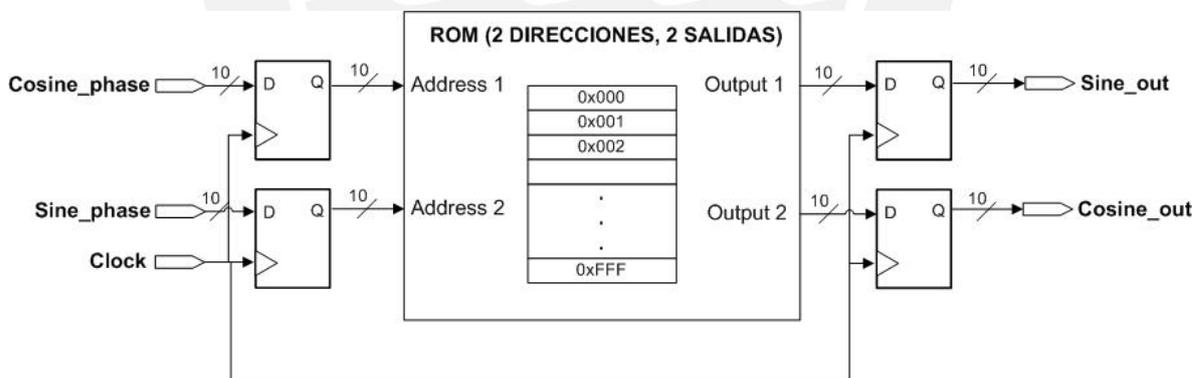
Existen diversas maneras de implementar un Acumulador de Fase, siendo una de las principales a través de un contador. Básicamente, el contador se pondría en marcha y el incremento estaría establecido por la señal TWN. Cuando se llegue al límite de la cuenta – establecida por la cantidad de puntos en la rueda de fase – se reinicia el contador. Esta alternativa es bastante efectiva, pero posee la desventaja de consumir una cantidad considerable de recursos. Otra manera de implementarlo, y quizás la que mejor mantiene en balance entre la cantidad de recursos usados y la efectividad en el funcionamiento es utilizando un registro basado en un flip-flop D con un lazo de realimentación [17]. Así, a la entrada del flip-flop se tiene la suma de la señal TWN con el valor de la salida del registro retroalimentada, como se ilustra en la Figura 12. La principal ventaja de esta arquitectura es que no es necesario colocar ningún circuito que verifique si es que ya se llegó al final de cuenta para forzar al reinicio de la misma, pues en caso se sobrepase la cuenta (overflow) simplemente se pierde el bit de acarreo. De este modo se asegura que el valor de la cuenta nunca sobrepase el rango establecido por la rueda de fase. La salida obtenida (señal Phase) es una representación de la fase de la onda sinusoidal que desea generar. Cuando esta señal Phase adquiere el valor de 0, es cuando la salida del DDS Enable se activa (toma el valor de '1'), pues esto asegura que se ha recorrido toda la rueda de fase.



**Figura 12. Esquema del Acumulador de Fase.**

#### 4.2.1.2. Convertidor Fase-Amplitud.

Este bloque constituye la última etapa del circuito del DDS. Como se ha mencionado anteriormente, es en esencia una memoria que contiene los valores que ordenados de manera apropiada forman la onda sinusoidal. Ya que los datos guardados en la memoria jamás serán modificados y su modo de uso será de solo lectura, la implementación de una ROM se considera lo más adecuado. Sin embargo, se observa a simple vista de la Figura 12, que no es una ROM convencional, pues consta de dos señales de dirección de memoria y de dos salidas, las cuales son correspondientes a las ondas seno y coseno que se necesita generar. El requerir dos señales en un inicio implicó la idea de implementar un DDS para cada una. Esto garantizaba mayor integridad de señal, mayor paralelismo, lo que luego se derivaba en menos latencia inducida. Sin embargo, implicaba el uso del doble de recursos, lo cual era bastante ineficiente. Debido a esto, cuando se descubrió la facilidad que otorgaba el acumulador de fase para generar desfases y así obtener ambas señales de un solo bloque, se decidió optimizar el uso de hardware e implementar una memoria de doble entrada y doble salida para el Convertidor Fase-Amplitud. El principal problema que podía presentar es el caso es el acceso simultánea a una sola dirección de memoria, con lo que ocurriría un conflicto en el funcionamiento de la ROM. No obstante, se logró comprobar que esta particular situación jamás iba a suceder, pues por el mismo desfase entre las señales, las direcciones de memoria nunca iban a coincidir en el mismo valor.



**Figura 12. Esquema del Convertidor Fase-Amplitud**

Los datos a ingresar dentro la ROM deben ser los apropiados para generar una señal sinusoidal. Ya que los desfases se han generado en base a una onda seno, solo se ingresarán en la memoria los valores ordenados adecuadamente para formar esa señal. Así, la ecuación a usar para llenar los valores dentro la memoria está expresada (6) [18]. Se puede observar que los números a ingresar varían con respecto a las longitudes de palabra escogidas tanto para la señal de

dirección de memoria (señales Address 1 y Address 2) como la de cada registro contenido en la ROM.

$$x[i] = \frac{1}{8} \times \left( 2^{D-1} + 2^{D-1} \sin\left(\frac{2\pi i}{2^m}\right) - 1 \right) \quad (6)$$

Si se realiza la correspondencia de las variables de la ecuación con el circuito a implementar para este sistema, se obtiene lo siguiente:  $D = 10$  (longitud de palabra de cada registro);  $m = 10$  (longitud de palabra de la señal dirección de memoria);  $x[i]$  no es otra cosa que el valor a guardar en el registro actual; y finalmente  $i$ , que viene a ser la posición de memoria actual. Utilizando estos datos es posible calcular el tamaño que ocupará la ROM a implementar. Así, el tamaño que la ROM ocupará será  $2^{10} * 10 = 10240$  bits [17].

Aprovechando el conocimiento a priori de la propiedad periódica de toda señal sinusoidal, se puede intuir que no es necesario colocar todos los datos calculados mediante la ecuación (6), puesto que esta cadena de datos también posee esta propiedad. Por este motivo, es posible escoger únicamente la cuarta parte de esta cadena de datos y aún así formar la onda seno. A esta propiedad se le conoce como Simetría de Cuarto de Onda [19]. De este modo, se logra reducir al 50 % el tamaño inicial ocupado por la ROM.

Finalmente, es necesario resaltar los flip-flops D ubicados a la entrada y salida de la memoria. El propósito de ambos es el mismo: disminuir al máximo la presencia de señales indeseadas en las salidas de la memoria. Cualquier señal errónea que ingrese a la ROM causará la selección de un registro equivocado dentro de la memoria, lo que finalmente ocasionará la presencia de una señal errónea en la salida del DDS. Si esto llegara a ocurrir, la falla se manifestaría a manera de sobre-impulsos en dentro de la señal, ya sean éstos negativos o positivos. La principal consecuencia sería un daño considerable a la pureza espectral de la señal. Por ende, se debe tomar todas las medidas posibles para prevenir esta falla. No obstante, cabe mencionar que la colocación de estos flip-flops ocasiona que la arquitectura presentada para el Convertidor Fase-Amplitud adquiera la característica de segmentación (pipeline). Esto da un mayor orden a la ejecución de las tareas de cada parte del circuito y hace predecible al sistema, pues se reduce al máximo la presencia de irregularidades en la señal de salida.

4.2.2. Descripción y diseño de la arquitectura del Mapeador de Constelaciones (Mapeador I&Q).

El Mapeador I&Q es una unidad funcional fundamental de todo el modulador. Este bloque es el encargado de escoger los coeficientes de las ondas sinusoidales en función de la constelación escogida. Como se mencionó anteriormente, se está trabajando con tres constelaciones (QPSK, QAM-16 y QAM-64), las cuales pueden variar dependiendo del diseñador del sistema. Debido se establecieron todas las constelaciones antes de proceder a la etapa de diseño. Así, las gráficas de dos ejes que definen a las constelaciones están ilustradas en las Figuras 13, 14 y 15, las cuales corresponden a las constelaciones QPSK, QAM-16, y QAM-64 respectivamente [20].

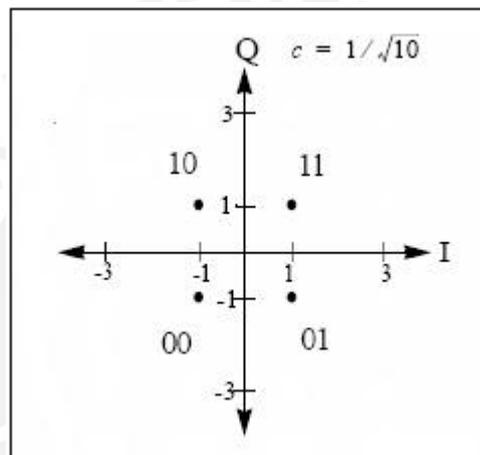


Figura 13. Constelación QPSK.

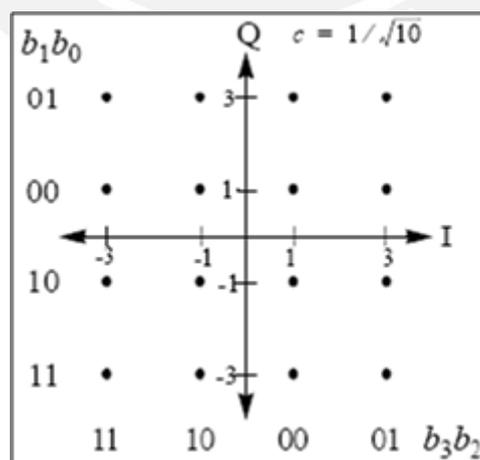
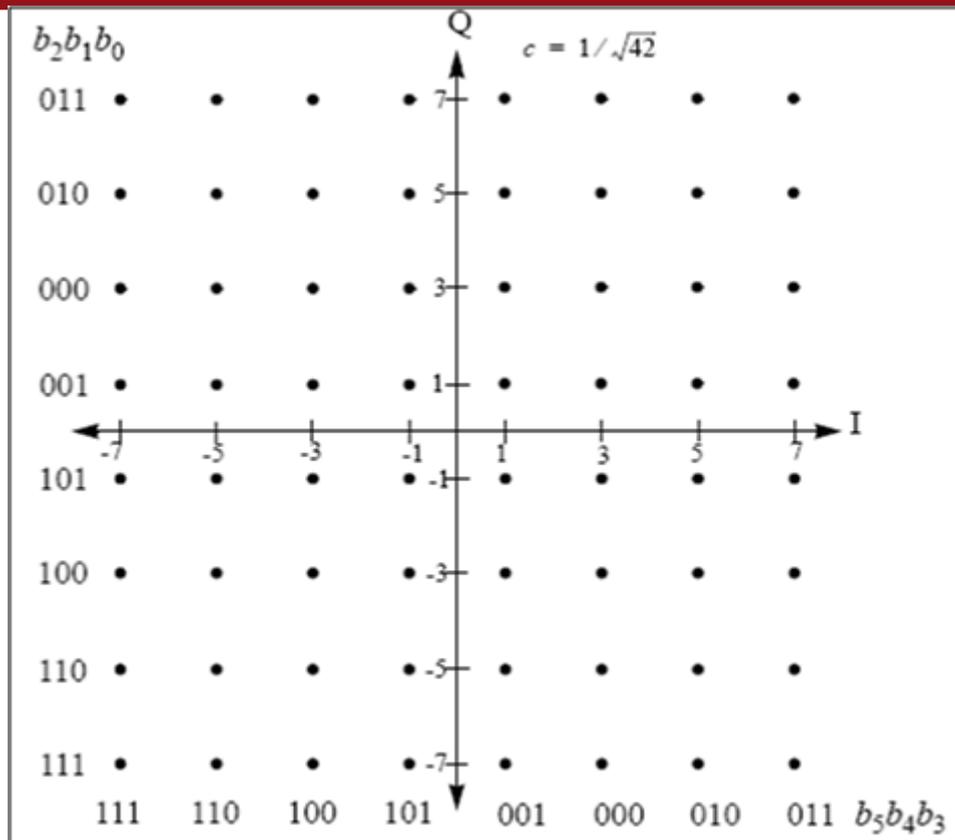


Figura 14. Constelación QAM-16.



**Figura 15. Constelación QAM-64.**

Como se puede apreciar, los coeficientes que se obtienen de cualquiera de las 3 constelaciones, pueden ser tanto positivos como negativos. Esto generó una problemática, pues si se diera el caso de trabajar con números negativos se tendría que trabajar con notaciones especiales para expresar los números obtenidos (complemento a 2). Sin embargo, ya que estos coeficientes serán los que se multipliquen por las ondas sinusoidales obtenidas de los DDSs, se prefirió aprovechar su facilidad para generar desfases y así simular el efecto de un coeficiente negativo (inversión de onda) en vez de trabajar con enteros positivos y negativos. De este modo, se simplifica las operaciones aritméticas, lo que se deriva en una mayor rapidez del sistema.

Analizando las constelaciones, se puede notar que dependiendo del caso, se necesitan manejar señales de 6 (QAM-64), 4 (QAM-16), y 2 (QPSK) bits. Asumiendo que se va a trabajar con un solo registro de entrada que almacena la trama de datos que ingresa al Mapeador I&Q, se escogió darle una longitud de palabra de 12 bits a dicho registro. Esta longitud se escogió debido a que se necesitaba un solo registro que pueda ser recorrido de 2 en 2 bits (6 pasadas), de 4 en 4 (3 pasadas), o de 6 en 6 (2 pasadas), dependiendo de la constelación seleccionada. Este

único registro de 12 bits es la señal clave de toda esta unidad funcional. Por ende, su inicialización y su actualización son procesos de vital importancia para el correcto funcionamiento del Mapeador. En consecuencia, la arquitectura del mismo se planteó básicamente en los procesos que involucran el uso de este registro. El esquema propuesto para el Mapeador I&Q se muestra en la Figura 16.

Partiendo del hecho que se está trabajando con 8 sub-portadoras y sabiendo a priori la tasa de transferencia de la trama de entrada principal (100 Mbps), se puede hallar fácilmente la tasa de transferencia de cada sub-canal:  $100 / 8 = 12.5$  Mbps, y así poder realizar la lectura de cada bit de la misma con el uso de la señal de reloj o Clock. Como se puede apreciar en la Figura 16, la sub-portadora (señal Sub-carrier) entra como una trama serial al Mapeador, con la cual se escribirá en el registro de 12 bits mencionado anteriormente. Esta escritura se realizará en el bloque denominado Cargador de Datos del Registro.

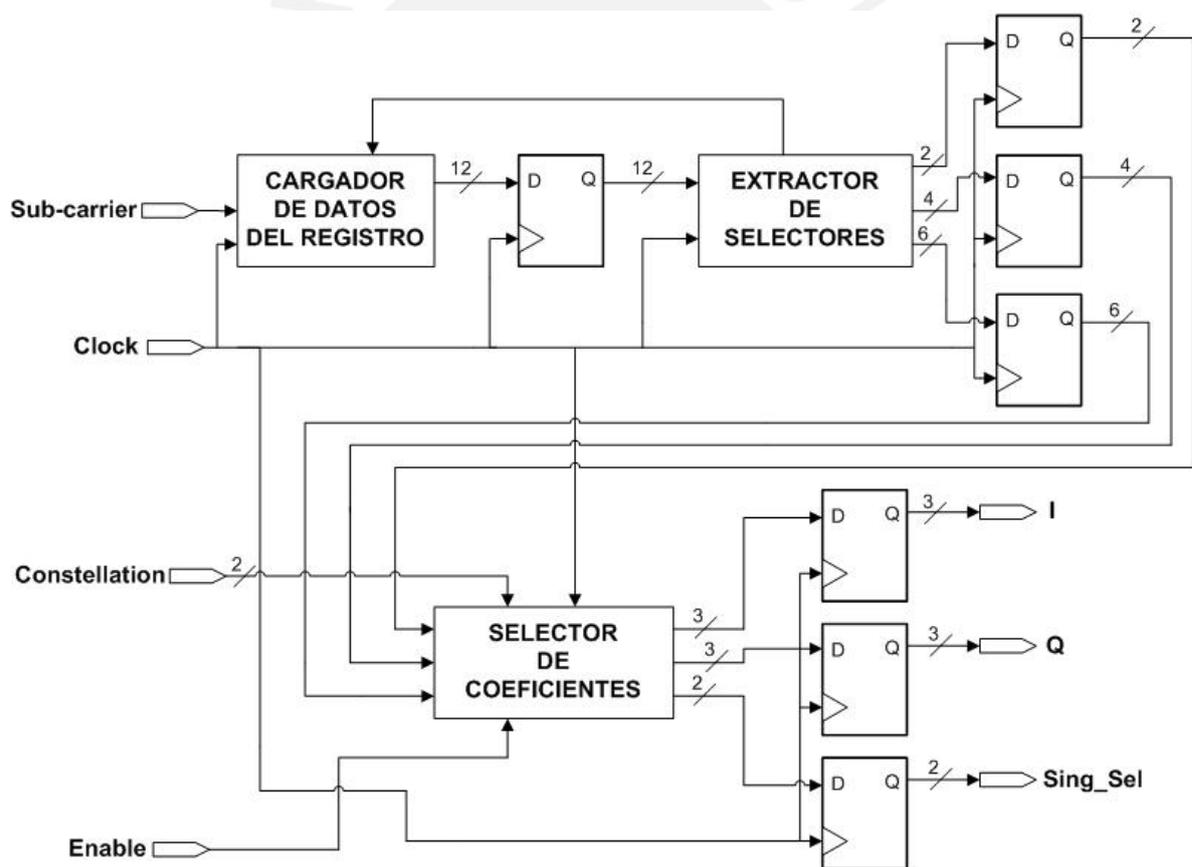


Figura 16. Esquema del Mapeador I&Q.

Sabiendo que la señal de reloj o Clock posee una frecuencia de 200 MHz (período de 5 ns), se calcula que la duración de un bit de información de una sub-portadora equivale a 16 flancos de subida de la señal Clock. Así, en función de la señal Clock se puede muestrear y leer la señal Sub-carrier, procesos que ocurren en el bloque Cargador de Datos al Registro.

Como se estableció anteriormente, la señal se almacenaría en un registro de 12 bits, el cual se convierte en la salida del bloque Cargador del Mapeador. Cabe resaltar, que la escritura de este registro toma una considerable cantidad de tiempo: se necesitan 16 flancos de subida (períodos de reloj) de la señal Clock por cada bit, lo que significa que en total se necesitarán  $16 * 12 = 192$  ciclos de reloj; expresado en unidades de tiempo, se necesitarán  $192 * 5\text{ns} = 960$  ns. En otras palabras, se necesita aproximadamente 1 ms para llenar el registro, lo cual deja una gran ventana de tiempo para ejecutar los procesos de las demás unidades funcionales. No obstante, el tener una gran ventana de tiempo para ejecutar los procesos tiene sus desventajas, siendo la principal la presencia de tiempos muertos en la señal de salida del circuito, lo que ocasionaría su discontinuidad en la misma y básicamente una deformación del espectro deseado. Para contrarrestar este efecto, se deben utilizar memorias internas para almacenar la data hasta el punto de poder tener un flujo continuo de datos en las siguientes etapas. Esto ocasiona el uso de más recursos y requiere la generación de una lógica de control para el manejo de esas memorias internas, pero se debe realizar para poder obtener la señal deseada. A estas etapas que generan una disminución en la velocidad del flujo de datos de un sistema se les conoce como “cuello de botella” (bottleneck), y la entrada del Mapeador I&Q es el principal cuello de botella de todo el sistema.

La salida del Cargador de Datos pasa por una etapa de carga de datos (flip-flop D) para evitar que la siguiente etapa reciba señales erróneas a la entrada. Los demás flip-flops D del esquema del Mapeador I&Q cumplen la misma función; es decir, tenemos una arquitectura segmentada. Luego del Cargador, sigue el bloque denominado Extractor de Selectores. Este subsistema está encargado de leer el registro de 12 bits que le llega de entrada y obtener de él los selectores de todas las constelaciones. Por ende, constará de 3 salidas: una señal de 2 bits por la constelación QPSK, una señal de 4 bits por la constelación QAM-16, y una señal de 6 bits por la constelación QAM-64. Con fines de agilizar la interacción con la etapa anterior, el Extractor de Selectores envía una señal de habilitación Cargador de Datos indicándole que ya terminó de recorrer todo el registro, por lo que se debe actualizar el valor del mismo. Así, se mantiene un flujo continuo de datos.

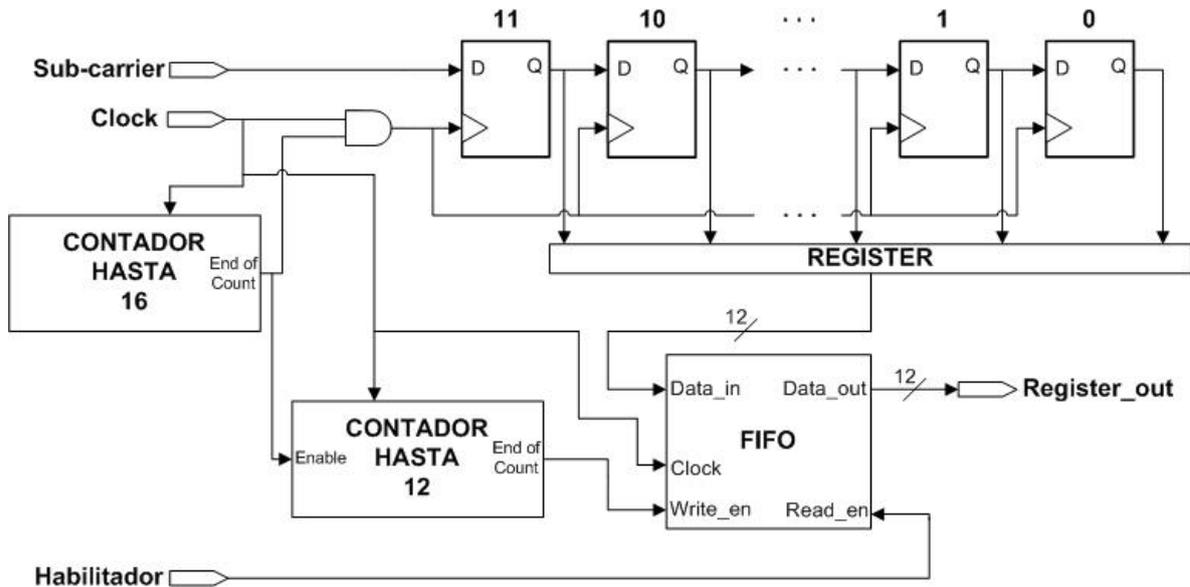
Después de pasar la etapa de carga, los 3 selectores ingresan al bloque denominado Selector de constelaciones. En este subsistema, dependiendo de la señal Constellation, la cual contiene la información de la constelación digital escogida, se procederá a seleccionar los coeficientes de las ondas sinusoidales: las componentes en fase (I) y en cuadratura (Q), junto con los signos que acompañan a cada una de ellos (señal Sign\_sel de 2 bits). En esta etapa se trabajó en su mayoría con memorias, tanto para el almacenamiento de las distintas constelaciones, como para contrarrestar los efectos del cuello de botella que se genera en la unidad funcional siguiente: el DDS. Ya que para generar las ondas sinusoidales se utiliza una considerable cantidad de ciclos de reloj, es necesario el uso de memorias internas para almacenar los valores de las señales I, Q y Sign\_sel para no perder la continuidad de la señal que se requiere generar. Así, como se indicó anteriormente, la señal Enable proveniente del DDS le indica al Selector de Constelaciones que ya puede dejar pasar el siguiente set de datos que representan los coeficientes de las ondas sinusoidales.

#### 4.2.2.1. Cargador de Datos del Registro.

Este bloque viene a ser la etapa inicial del Mapeador, y además, como se estableció anteriormente, el principal cuello de botella de todo el modulador. La sub-portadora ingresa como una trama serial al Cargador de Datos, para luego convertirse en la entrada de un convertidor serial/paralelo de 12 bits. Este convertidor, implementado de manera simple mediante 12 flip-flops D en cadena, no recibe como señal de reloj la señal Clock (reloj de todo el sistema), si no la salida de una compuerta lógica AND. Debido a que cada bit de la sub-portadora ocupa un período equivalente a 16 ciclos de la señal Clock, se necesita usar un bloque que lleve esta cuenta para así realizar la lectura de la señal en el momento preciso. En consecuencia, la salida del Contador Hasta 16 (señal End of Count), junto con la señal Clock ingresan a la compuerta AND, y la salida de la misma se convierte en la señal de reloj del convertidor serie/paralelo de 12 bits, como se puede apreciar en la Figura 17.

Los 12 bits en paralelo ingresan al registro llamado Register, el cual luego se convierte en la entrada de la memoria FIFO (First-In, First Out) utilizada como buffer de almacenamiento para contrarrestar el efecto de cuello de botella. Esta memoria FIFO es en esencia una simple memoria RAM, con su dirección de memoria tanto para lectura como para escritura; lo único particular es la manera en que va aumentando el valor de estas direcciones de memoria. Así, la dirección de memoria de escritura varía de manera ascendente con un incremento constante de 1

espacio de memoria a medida que más información necesita ser almacenada. La dirección de memoria de lectura varía de manera similar, a medida que los requerimientos del uso de la información almacenada vayan aumentando. Cuando ambas direcciones alcancen el límite permitido, simplemente se reinician.



**Figura 17. Esquema del Cargador de Datos al Registro.**

Las señales Write\_en y Read\_en funcionan como señales de habilitación de escritura y lectura de los registros de la FIFO respectivamente, por lo que son señales que influyen de gran manera en el uso de la misma. A la señal Write\_en se le direcciona la salida del Contador Hasta 12, la cual fundamentalmente cumple el trabajo de asegurar que se completó la lectura del nuevo set de 12 bits de la sub-portadora. Adicionalmente, la señal Enable o habilitadora de este contador tiene direccionada la señal End of Count del bloque Contador Hasta 16 (señal que asegura la lectura de 1 solo bit de la señal Sub-carrier). Así, únicamente cuando se culmine la lectura de un bit de la señal Sub-carrier, se incrementará la cuenta del bloque Contador Hasta 12; de esta manera se logra asegurar que se guarde la información correcta en los registros de la FIFO.

Finalmente, la señal Read\_en de la FIFO tiene direccionada la señal de entrada Habilitador, la cual proviene del bloque ubicado a continuación del Cargador de Datos: el Extractor de Selectores. Esta señal tiene la simple tarea de habilitar la lectura de los registros contenidos en la FIFO, por lo que solo se debe activar (tener el valor de '1' lógico) cuando el bloque Extractor de Selectores esté preparado para recibir el nuevo set de 12 bits.

4.2.2.2. Extractor de Selectores.

Este bloque, como su nombre lo indica, tiene la tarea de extraer los bits de selección de las constelaciones digitales disponibles. La principal entrada de este subsistema es la señal de 12 bits proveniente del registro del Cargador de Datos del Registro. Esta palabra de 12 bits se divide en 3 señales iguales (todas de igual longitud), las cuales entran a distintos convertidores serial/paralelo (flip-flops D en serie), cada uno de una cierta longitud de palabra dependiendo de la constelación digital a la que esté relacionado: 1 de 2 bits para QPSK, 1 de 4 bits para QAM16, y 1 de 6 bits para QAM64. Luego, las salidas de los convertidores serial/paralelo son almacenadas en registros (REG\_QPSK, REG\_QAM16, y REG\_QAM64) para su posterior uso.

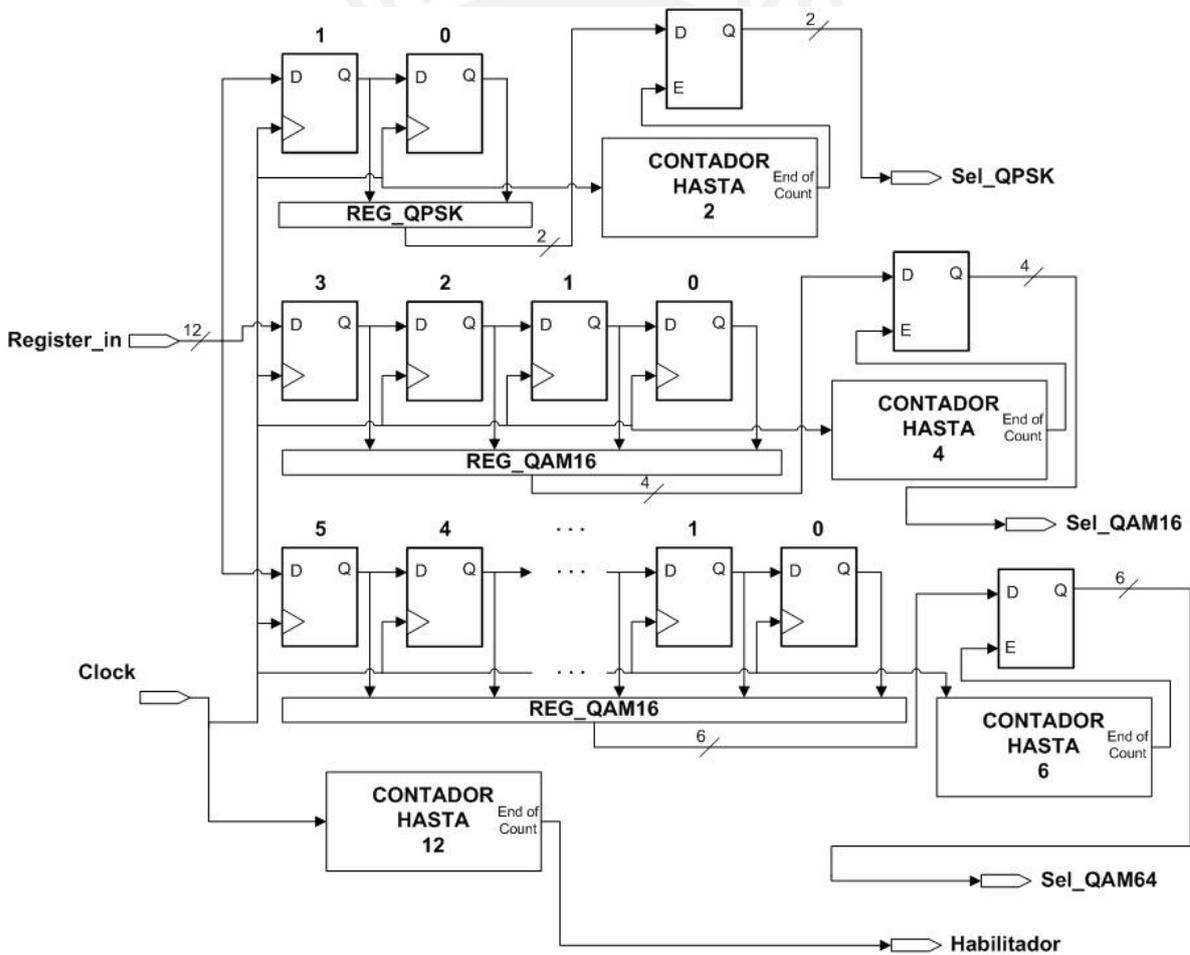


Figura 18. Esquema del Extractor de Selectores.

Cada registro es direccionado a la entrada de un latch, cuya señal de habilitación es el bit de fin de cuenta (End of count) de un contador. El valor de la cuenta depende la constelación a la que corresponda. Así, se necesitaría llevar la cuenta de los números 2, 4 y 6 para las constelaciones QPSK, QAM16 y QAM64 respectivamente. Esto solamente para garantizar que a la salida de cada latch se tenga la data correcta sobre los selectores de constelaciones. Las salidas de los latches se convierten en las salidas del bloque: Sel\_QPSK, Sel\_QAM16, y Sel\_QAM64, las cuales contienen los selectores para la siguiente etapa del Mapeador.

Además de los contadores ya mencionados, se tiene uno adicional que se encarga de verificar el término del recorrido de los 12 bits de la señal Register\_in. Esta cuenta se necesita llevar para indicar al bloque anterior (Cargador de Datos) que el subsistema ya está preparado para recibir un nuevo set de 12 bits para procesar. Así, la salida Habilitador toma el estado de '1' lógico cuando se terminó de recorrer el set actual, y '0' lógico en el caso contrario.

La presencia de los latches en conjunto con los contadores es necesaria para extraer los selectores de manera correcta, evitando el traslape de información en la salida de este bloque. Si se analiza con detalle la arquitectura del Extractor de Selectores, en específico los registros de salida de los convertidores serie/paralelo, se puede notar que en algunos momentos la data contenida en los mismos no será la correcta. Esto ocurre principalmente debido a que los posibles selectores son señales de 2 bits, 4 bits o 6 bits dependiendo de la constelación, y todos ellos están contenidos dentro de la señal Register\_in en grupos de 6, de 3 o de 2 respectivamente. Por ende, grupos de datos traslapados no son útiles para el sistema. He ahí la importancia de los contadores, los cuales únicamente habilitan los latches cuando se tiene la data correcta dentro de los registros.

#### 4.2.2.3. Selector de Constelaciones.

El bloque Selector de Constelaciones es la etapa final del Mapeador I&Q, y consiste básicamente en una serie de memorias que contienen los coeficientes correspondientes a las distintas constelaciones disponibles. Por cada constelación se tiene una memoria para la componente “en fase” o I (In Phase), una memoria para la componente “en cuadratura” o Q (Quadrature), y finalmente una memoria para almacenar la información acerca del signo de las ambas componentes. Así, la componente I corresponde a la onda coseno; y que la componente Q, a la seno. Adicionalmente, la memoria que contiene la información sobre los signos de cada

componente se expresa en una palabra de 2 bits de longitud: el bit más significativo (MSB: Most Significant Bit) corresponde a la componente Q, y el bit menos significativo (LSB: Least Significant Bit), a la componente I.

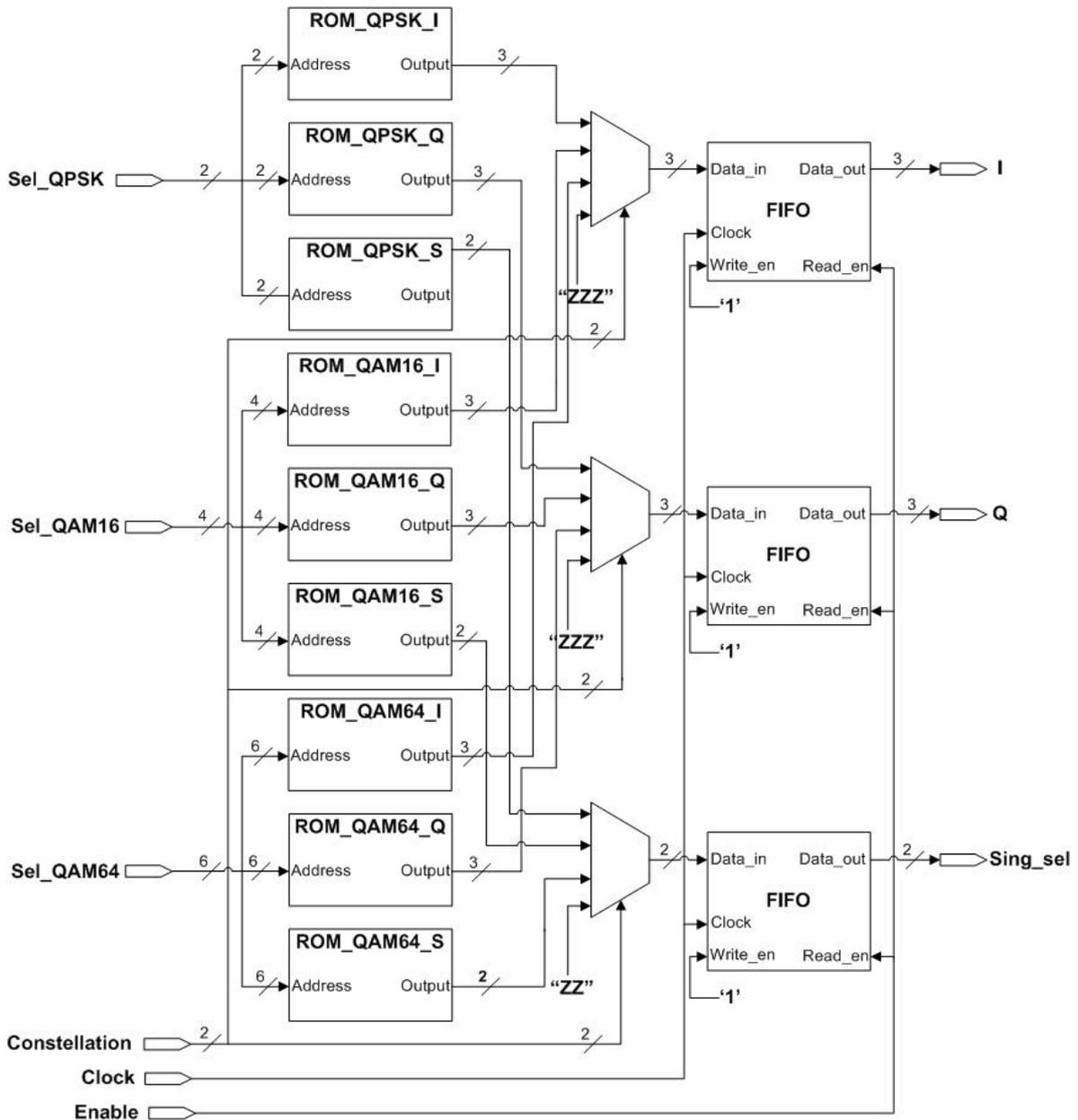


Figura 19. Esquema del Selector de Constelaciones.

Cada selector (Sel\_QPSK, Sel\_QAM16 y Sel\_QAM64) ingresa a 3 distintas memorias actuando en cada uno de ellas como la señal de dirección de memoria, obteniendo así los 3 posibles sets de componentes. La salida de cada memoria ingresa a un multiplexor, dependiendo del tipo de

información que contenga. De este modo, se tienen 3 multiplexores: 1 para escoger entre las posibles componentes I, 1 para la selección de la componente Q, y finalmente 1 para la palabra de 2 bits que indica el signo de las componentes. Se puede apreciar en la Figura 19 que en cada multiplexor la última entrada siempre es una señal de alta impedancia. En el caso que el subcanal se desactive por completo las salidas del Mapeador serán alta impedancia, razón por la cual se coloca alta impedancia en cada multiplexor. Así, el selector de cada multiplexor viene a ser la entrada del modulador Constellation, la cual lleva la información sobre la información escogida para la sub-portadora, como se puede apreciar en la Tabla 1.

**Tabla 1. Información contenida en la Señal Constellation**

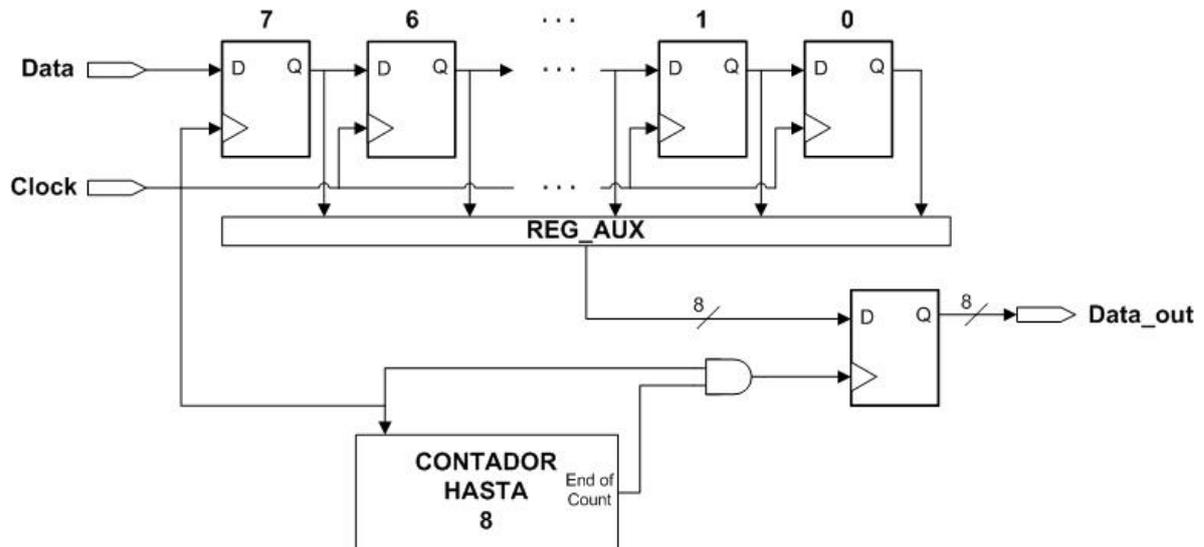
Señal Constellation	
Palabra de 2 bits	Acción
00	Usar QPSK
01	Usar QAM-16
10	Usar QAM-64
11	INHABILITADO

En la arquitectura planteada se puede notar nuevamente la presencia de memorias FIFO. Al igual que un esquema anterior, su propósito es contrarrestar el efecto de cuello de botella. En este caso particular, la señal de Read\_en viene de la señal Enable, la cual es enviada por el DDS indicando que la unidad funcional está preparada para recibir un nuevo set de coeficientes para las ondas seno y coseno. De esta manera, la salida de cada FIFO se convierte en la salida de todo el bloque Selector de Constelaciones.

#### 4.2.3. Descripción y diseño de la arquitectura del Convertidor Serial/Paralelo.

La señal Data – trama digital que se requiere modular – ingresa el Convertidor Serial/Paralelo para ser procesada. Esta señal se direcciona a la entrada de un flip-flop D, el cual forma parte de una serie de flip-flops en cadena colocados precisamente así para lograr dividir la trama serial que llega a la entrada. Ya que se cuenta con 8 sub-portadoras, se necesitan 8 sub-canales, lo que significa que el Convertidor Serial/Paralelo necesita entregar una palabra de 8 bits de longitud a la salida. Sin embargo, antes de direccionar las salidas de los flip-flops a las salidas de todo el bloque, es necesario asegurar que a la salida se tenga la información correcta (evitar traslape de información). En otras palabras, cada set de 8 bits a la salida de los flip-flops no siempre es el

set que se necesita, pues habrán momentos en los cuales se tenga actualizada la salida de un par de flip-flops D, mas no del resto; si en ese momento se envía la data a la salida se estaría dando a las siguientes unidades funcionales información errónea. Las consecuencias de enviar información errónea a la salida de cualquier unidad funcional de este sistema son catastróficas, y se resumen principalmente en la inducción de ruido en la señal de salida. Para tener un nuevo set de 8 bits correctos se necesita 8 ciclos de reloj. He ahí la razón para colocar el bloque Contador Hasta 8 en la arquitectura del Convertidor, como se puede observar en la Figura 20.

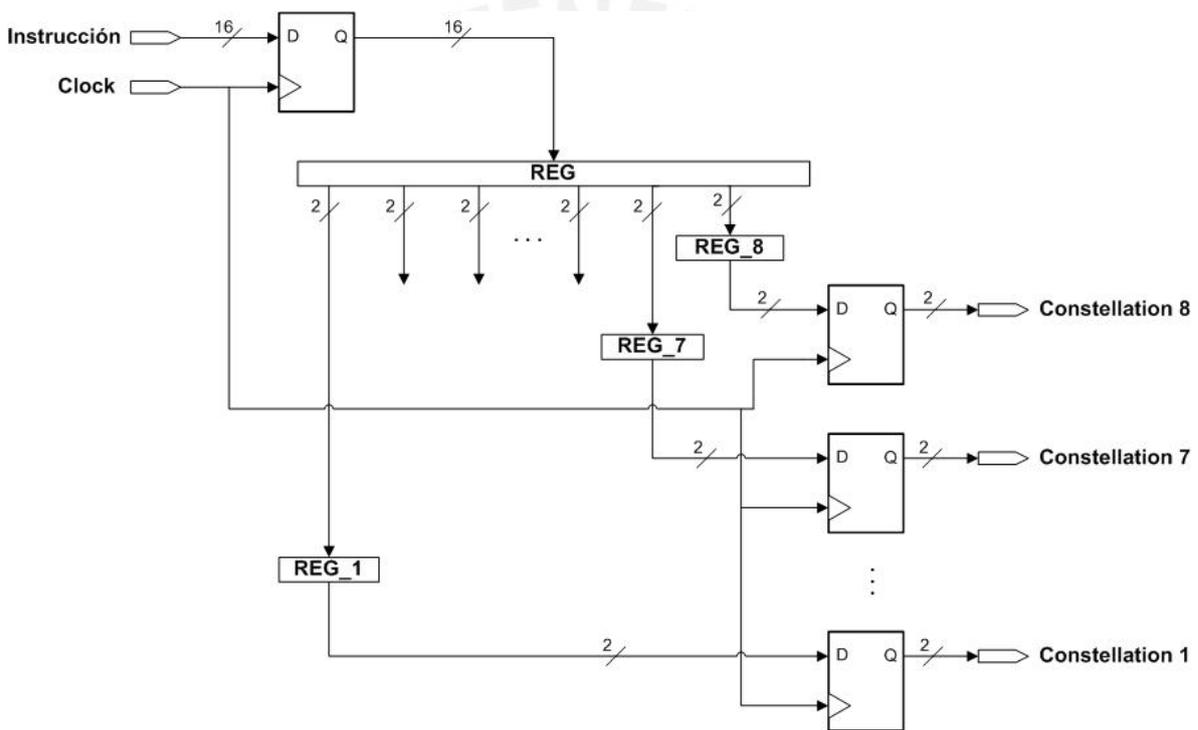


**Figura 20. Esquema del Convertidor Serial/Paralelo.**

Por las razones explicadas anteriormente, la salida de los flip-flops D se direccionan a un registro denominado REG\_AUX, almacenando así la señal antes de mandarla a la salida. Este registro se coloca a la entrada de un flip-flop D, el cual entregará la salida de todo el bloque. La señal de reloj que recibe este flip-flop no es la señal Clock que ingresa al Convertidor, pues ésta ingresa a una compuerta lógica AND, en conjunto con la salida End of Count del Contador Hasta 8. La salida de esta compuerta AND es la que se convierte en la señal de reloj que activa al flip-flop D. Así, se tiene una especie de habilitador par el flip-flop, dejando pasar únicamente la información correcta a la salida del mismo. A esta técnica se le conoce reloj en el idioma inglés como “gated clock”.

#### 4.2.4. Descripción y diseño de la arquitectura de la Unidad de Control.

Esta unidad funcional es quizás la más importante de todas, y para esta arquitectura en particular, la más simple también. Generalmente, para procesadores en específico, la unidad de control viene a ser una compleja máquina de estados pues es aquella encargada de mandar las señales de control a todas las demás unidades funcionales del sistema. Por ende, cuanto más compleja sea la arquitectura, más compleja es la máquina de estados de la unidad de control. Sin embargo, para el caso de este sistema, la Unidad de Control únicamente estaba encargada de manejar a los Mapeadores I&Q, pues los DDSs son controlados por las señales de sintonización TWN.



**Figura 21. Esquema de la Unidad de Control.**

La señal Instrucción ingresa directamente a la Unidad de Control. Esta señal consta de 16 bits de longitud, y contiene la información de las constelaciones a usar por todas las sub-portadoras. Cada grupo de 2 bits contenidos en la señal Instrucción vienen a convertirse en las subsiguientes señales Constellation que ingresan a los Mapeadores I&Q.

Inicialmente, la señal Instrucción pasa por una etapa de carga de datos, para luego ser direccionada a un registro interno llamado REG, como se ilustra en la Figura 21. A partir de la

información almacenada en el registro se procede a obtener todas las palabras de 2 bits que se convertirán luego en las señales Constellation. Los 2 MSBs irán al sub-canal 8, los 2 siguientes al 7, y así sucesivamente hasta llegar al 1. Luego de obtener los grupos de 2 bits del registro REG, se coloca una etapa de carga de datos antes de mandar las señales a la salida de la Unidad de Control.

#### 4.2.5. Descripción y diseño de la arquitectura del Modulador OFDM Reconfigurable.

La arquitectura planteada para el Modulador OFDM Reconfigurable se basa en las unidades funcionales antes mencionadas. Ahora que se conoce sus esquemas y su funcionamiento, la explicación de la arquitectura del sistema general, la cual se muestra en la Figura 22, será más fluida.

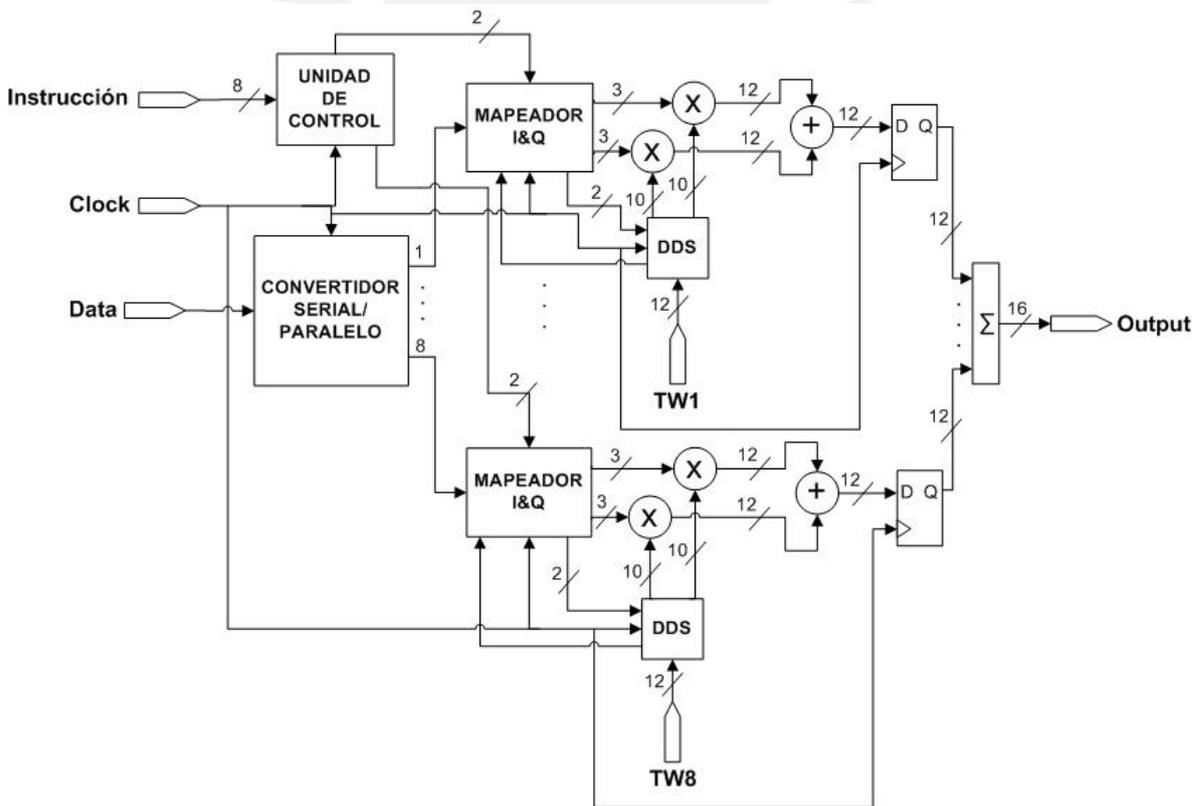


Figura 22. Arquitectura del Modulador OFDM Reconfigurable.

La trama serial Data ingresa al sistema y es direccionada al Convertidor Serial/Paralelo. Ya que esta trama digital posee una tasa de transferencia de 100 Mbps y el Convertidor entrega una palabra de 8 bits, la salida en cada sub-canal tendrá una tasa de transferencia de 12.5 Mbps.

Cada salida del Convertidor Serial/Paralelo ingresa a un Mapeador I&Q, el cual a partir de la trama encuentra los coeficientes apropiados para las ondas seno y coseno que serán generadas luego. Además de tomar en cuenta la trama de entrada, la selección de los coeficientes depende de la constelación digital a usar. La información acerca de la selección de la constelación a usar por cada sub-portadora viene contenida en la señal Instrucción. Esta palabra de 16 bits de longitud ingresa directamente a la Unidad de Control, la cual se encarga de repartir las señales Constellation a cada Mapeador I&Q. Una vez analizada la señal Constellation dentro del Mapeador I&Q, se obtiene los coeficientes de las componentes en fase (I) y en cuadratura (Q) de la sub-portadora. Debido a que algunos coeficientes pueden ser negativos, del Mapeador I&Q también se obtiene la señal Sign\_Sel de 2 bits, la cual contiene la información acerca de qué señales necesitan invertirse para así generar el efecto de un coeficiente negativo.

La señal Sign\_Sel ingresa al bloque DDS, el cual es el encargado de generar las ondas seno (Q) y coseno (I) para la sub-portadora. El MSB de esta señal determina la inversión de onda para la onda seno, y el LSB, para la onda coseno. Sin embargo, la transferencia de los coeficientes I, Q y de la señal Sing\_Sel es administrada mediante una señal Enable o habilitadora que sale del DDS hacia el Mapeador I&Q. Así, solo pasará un nuevo set de datos cuando el DDS haya terminado de generar las ondas sinusoidales y esté listo para nuevos coeficientes. Por otro lado, la frecuencia de estas ondas sinusoidales viene determinada por la otra señal de entrada del DDS: la TWN (N: 1, 2, 3, ...8) o señal de sintonización. Esta señal de 12 bits ingresa al acumulador de fase contenido dentro del DDS y establece la velocidad de recorrido de la rueda de fase, lo que finalmente se traduce en la frecuencia de las ondas generadas. La salida que corresponde a la onda seno ingresa a un multiplicador, en conjunto con la salida Q del Mapeador I&Q, al igual que la salida I y la onda coseno. Ambos productos se suman, para así obtener la sub-portadora requerida.

El resultado de la suma de ambas componentes de cada sub-portadora ingresa a una etapa de carga de datos (flip-flop D), para minimizar al máximo la presencia de ruido en la señal. A continuación, cada sub-portadora ingresa a un sumador para así obtener la señal OFDM requerida en el dominio digital. La salida de este sistema se direcciona directamente a un DAC, el cual se encarga de generar la onda analógica.

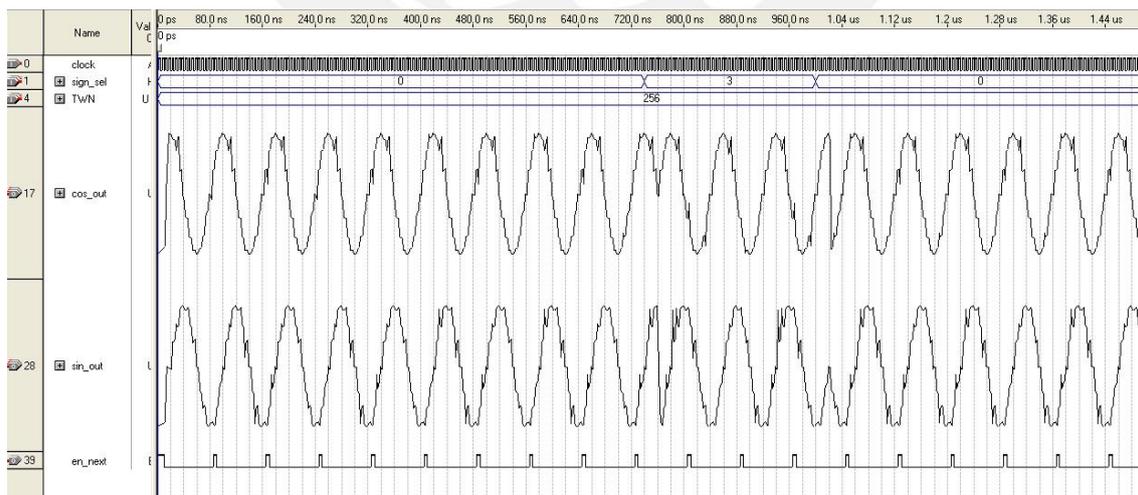
## CAPÍTULO 5: IMPLEMENTACIÓN Y RESULTADOS

El presente capítulo muestra tanto los procesos de verificación del diseño realizado como los resultados obtenidos. Inicialmente, cada unidad funcional descrita en el capítulo anterior es simulada por separado para validar su funcionamiento frente a diversas señales de entrada, siempre tratando de cubrir la mayor cantidad de casos posibles. Una vez realizado esto, se procede a ejecutar la simulación de todo el sistema completo. Finalmente, se procede a la implementación del sistema completo sobre el FPGA y a la visualización del funcionamiento real del circuito. Los resultados de las pruebas realizadas son mostrados en la parte final del capítulo.

### 5.1. Simulaciones.

La simulación de los diversos bloques fue realizada en el software Quartus II 8.0 Web Edition, para así corroborar el funcionamiento de cada uno. Las simulaciones fueron realizadas en el modo *Timing*, el cual toma en cuenta los retardos de las señales dentro del dispositivo.

Las principales simulaciones a realizar son las correspondientes a las unidades funcionales mencionadas en el capítulo anterior: el DDS, el Mapeador I&Q, el Convertidor Serie/Paralelo, y la Unidad de Control. Una vez culminadas las simulaciones de las unidades funcionales, se procederá a la simulación de toda la arquitectura completa.



**Figura 23. Simulación del DDS.**

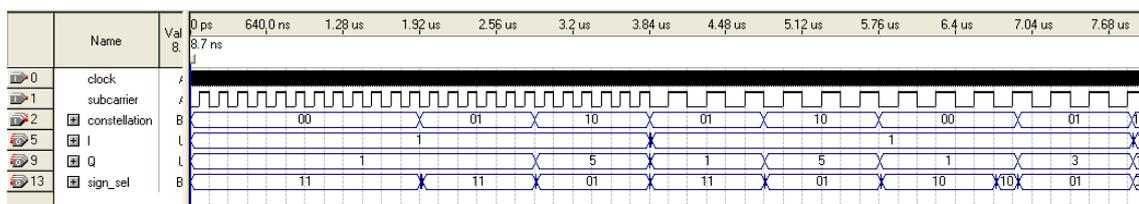
Para la simulación del DDS se generó ondas sinusoidales de 12.5 MHz, y se evaluó ambos casos: tanto de ondas normales como de ondas invertidas. Así, para la frecuencia requerida se escogió el valor de 256 – usando la ecuación (1) – para la señal TWN, y para la señal Sign\_sel se escogió los valores de “00” (0 en notación decimal) y “11” (3 en notación decimal). El primer valor de la señal Sign\_sel corresponde a las ondas no invertidas, y el segundo, a las invertidas.

Las señales generadas cos\_out y sin\_out corresponden a las señales coseno y seno respectivamente. Se puede comprobar en la Figura 23 que se logra una exitosa generación de ondas, habiendo un constante desfase de  $90^\circ$  entre ellas. Se puede apreciar también la presencia de algunas componentes ruidosas, que en algunos tramos degeneran en cierta forma a las ondas sinusoidales. Sin embargo, su efecto en la forma de onda de la señal es mínimo y casi insignificante. Además, como para toda modulación de banda ancha, la información a transmitir se guarda en la frecuencia de la señal generada y no en su amplitud. En consecuencia, una diminuta deformación en la forma de onda no presenta mayores problemas.

En el momento que se cambia el valor de la señal Sign\_sel por 3, se genera la inversión de las ondas sinusoidales. Esta inversión se observa claramente en el cambio brusco que se genera a la mitad del tiempo de simulación, un par de nanosegundos de actualizar el valor de la señal Sign\_sel.

Por último, se percibe que la salida en\_next toma el valor de ‘1’ lógico (5 voltios) cada vez que empieza un período de las ondas sinusoidales. Esto garantiza que se actualice el valor de los coeficientes de las constelaciones obtenidos del Mapeador I&Q cada vez que empiece otro período de las ondas.

En cuanto a la simulación del Mapeador I&Q, se tuvo que evaluar básicamente la correcta correspondencia entre la sub-portadora de entrada, la constelación seleccionada y los coeficientes obtenidos a la salida. Para poder evaluar distintos casos, se colocaron varios valores para la constelación escogida, como se puede apreciar en la Figura 24 (señal constellation).



**Figura 24. Simulación del Mapeador I&Q**

El valor de la sub-portadora escogida (señal sub\_carrier) también se fue variando para observar los cambios en las salidas I, Q, y Sign\_sel. Si se compara los valores de estas señales con respecto a las constelaciones descritas en el capítulo anterior (Figuras 13, 14 y 15) se comprueba el correcto funcionamiento de este bloque, pues los coeficientes obtenidos son los correctos, tanto en magnitud como en signo.

Cabe resaltar también que al observar la Figura 24, los tiempos de latencia o de respuesta del circuito son muy cortos, lo cual es un muy buen resultado. No obstante, es importante recordar que esta unidad funcional necesita muchos ciclos de reloj para poder muestrear y leer la trama serial de entrada. Esto llevó a la necesidad de aumentar los rangos de tiempo de la simulación, para así poder visualizar apropiadamente los resultados. Por ende, los intervalos de tiempo observados en la simulación del Mapeador superan en gran cantidad a los utilizados en las demás unidades funcionales.

Por otro lado, con respecto a la simulación de la Unidad de Control, se pudo simular fácilmente su funcionamiento otorgando diversos valores a la señal Instrucción, o en el caso de la simulación, Instruction (señal de entrada de 16 bits de longitud). Los bits contenidos en esta señal son los que deben repartirse en grupos de 2 bits a las salidas const\_1, const\_2, etc. Estas señales representan la constelación escogida para cada sub-portadora.

A la señal Instruction le fueron otorgados diversos valores completamente aleatorios para su evaluación. Como se puede apreciar claramente en la Figura 25, cada salida toma un grupo de 2 bits de la señal Instruction: los 2 LSBs para const\_1, los 2 subsiguientes para const\_2, y así sucesivamente hasta llegar a los 2 MSBs que corresponden a la señal const\_8. En consecuencia, la simulación demostró el correcto funcionamiento de la Unidad de Control.

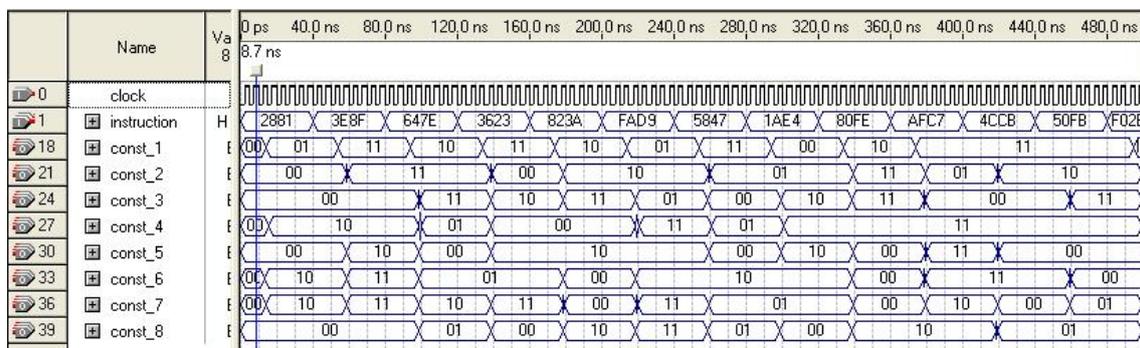
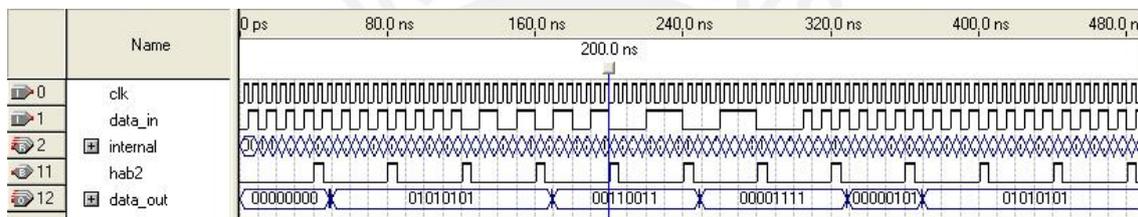


Figura 25. Simulación de la Unidad de Control.

Sin embargo, es necesario hacer hincapié en la presencia de retardos de señal que también se pueden apreciar en la figura de simulación. A pesar de esto, la performance del bloque no se ve afectada.

La simulación del Convertidor Serie/Paralelo, al igual que casi todas las demás, no requiero de un período largo de análisis, pues debido a que funciona en base a la señal Clock de 200 MHz de frecuencia, ejecuta los procesos de manera muy veloz, por lo que el tener una ventana de tiempo en escala de nanosegundos fue suficiente. Esto también se dio en la mayoría de las otras unidades funcionales, siendo la única excepción el Mapeador I&Q.

Se le asignó ciertos valores a la trama señal de entrada de datos (señal data\_in), primando en ellos la intercalación de los 2 posibles estados de la señal ('0' u '1' lógico). El tiempo de intercalación el principal factor de variación de señal, como se puede observar en la Figura 26.



**Figura 26. Simulación del Convertidor Serie/Paralelo.**

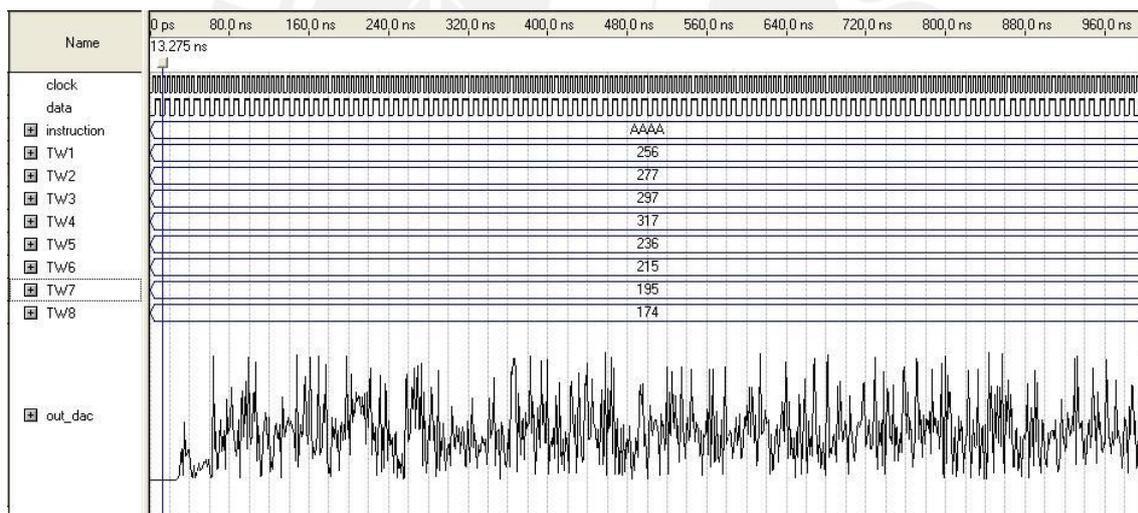
Al observar la gráfica de simulación se puede percibir que los cambios en la señal de salida (data\_out) se dan cada ocho ciclos de reloj, identificados claramente mediante la señal hab2. Esta señal se activa únicamente luego de la ocurrencia de 8 flancos de subida en la señal clk, dejando pasar la palabra de 8 bits a la salida del Convertidor.

Luego de haber logrado simular exitosamente todas las principales unidades funcionales por separado, se procedió a simular todo el modulador OFDM. El código se realizó en función de bloques pre-existentes, lo cuales vienen a ser las unidades funcionales antes mencionadas.

Para la simulación de este complejo sistema, se tuvo que evaluar los valores de entrada a colocar en las variables en función de la señal que se deseaba formar. Así, en primer lugar se seleccionó una señal de reloj de 100 MHz para la trama de entrada (señal data), para así poder evaluar mejor la lectura de la misma. Para la variable Instruction se escogió el valor hexadecimal de 0x AAAA para que todas las constelaciones a utilizar sean las mismas: QAM-64, la cual es la más compleja y la que más latencia induce al sistema. Finalmente, para las

palabras de sintonización (señales TW1, TW2, etc.) los valores fueron escogidos en función de la frecuencia que se quería generar por cada sub-portadora. La selección de las frecuencias fue de manera aleatoria y se simuló tanto el circuito implementado, como el modelo ideal descrito en el software Matlab R2007 para verificar la similitud de las mismas. Lo que motivó a realizar ambas simulaciones por separado y luego comparar resultados es la falta de herramientas de procesamiento de señal en el software Quartus II 8.0 Web Edition. No es usual realizar el análisis en el tiempo de una señal OFDM pues su información se encuentra en su espectro de frecuencias. Sin embargo, el Quartus II 8.0 Web Edition no cuenta con una función de FFT para visualizar apropiadamente el resultado obtenido, por lo que solamente se puede observar la señal obtenida en el tiempo.

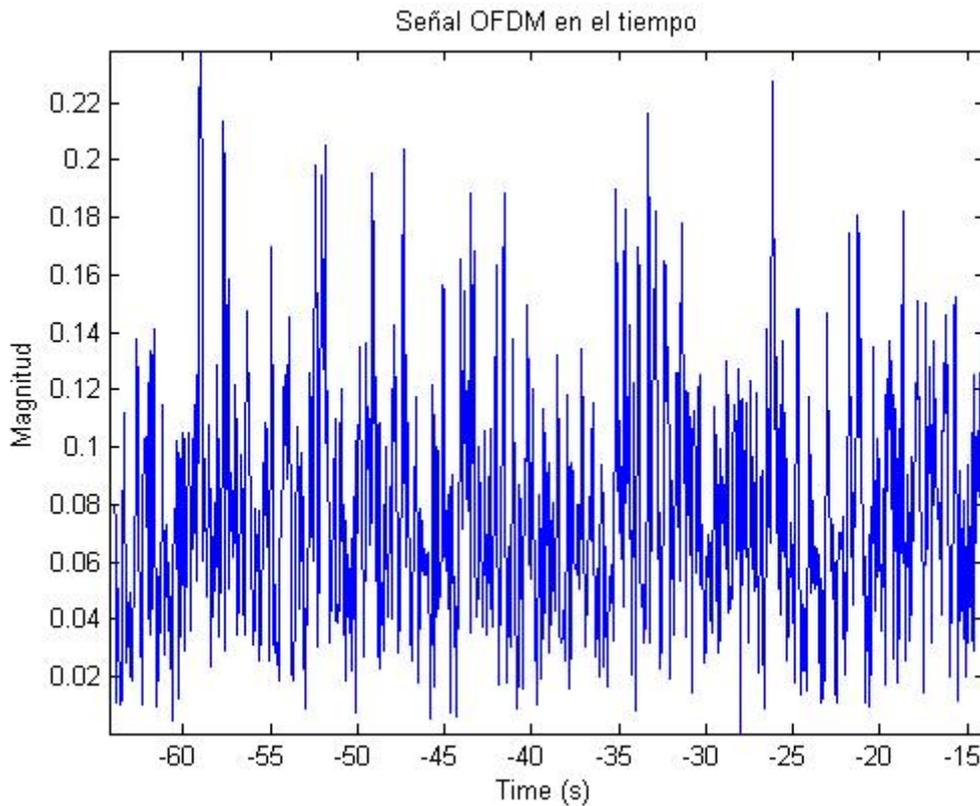
Así, las frecuencias obtenidas para las portadoras fueron las siguientes: 12.5 MHz, 11.5 MHz, 10.5 MHz, 9.5 MHz, 8.5MHz, 13.5 MHz, 14.5 MHz, y 15.5 MHz. Utilizando la ecuación (1), se puede encontrar los datos a ingresar en las variables T<sub>WN</sub>, los cuales fueron: 256, 236, 215, 195, 174, 277, 297 y 317 respectivamente, como se puede apreciar en la Figura 27.



**Figura 27. Simulación del Modulador OFDM Reconfigurable.**

El resultado obtenido en la simulación, parece indicar que el dispositivo está funcionando correctamente. Sin embargo, para poder corroborarlo se debe comparar este resultado con aquel obtenido mediante el modelo ideal. Si se encuentra gran similitud, se puede afirmar que se obtuvo un resultado exitoso.

La gráfica 28 muestra el resultado obtenido de la simulación del modelo ideal, las mismas frecuencias y las mismas constelaciones fueron escogidas para poder dar fe del buen resultado de la simulación del circuito. Se puede apreciar, a grandes rasgos, que las ondas obtenidas en cada simulación sí poseen gran nivel de similitud. Esto finalmente comprueba que la etapa de simulación del sistema ha sido exitosa y se debe proceder a la implementación en hardware.



**Figura 28. Señal OFDM ideal en el dominio del tiempo.**

No obstante, es necesario recalcar que inclusive, solamente analizando los resultados de simulación, ya se puede intuir que habrá una latencia y una presencia significativa de espurias en la señal de salida. Ambos efectos ocurren por el mismo diseño del circuito, y es posible que algún cambio en la arquitectura mejore este aspecto. Sin embargo, generalmente estos efectos ocurren por las celdas básicas usadas por el software Quartus II 8.0 Web Edition para sintetizar el circuito, elementos a los cuales no se tiene acceso. Por ende, un cambio en la arquitectura podría llevar a disminuir estos efectos, pero éste sería muy leve.

## 5.2. Implementación.

La implementación en hardware se llevó a cabo en el FPGA Cyclone II EP2C35F672C6 de la compañía Altera, y la evaluación de resultados fue llevada a cabo mediante el uso de la tarjeta de desarrollo DE2 de la misma compañía.

El código VHDL del modulador OFDM, descrito sobre el software Quartus II 8.0 Web Edition, fue sintetizado para obtener los datos sobre el uso de recursos del FPGA. El resultado otorgado por el software está ilustrado en la Figura 29.

Flow Summary	
Flow Status	Successful - Fri Nov 28 03:50:27 2008
Quartus II Version	8.0 Build 215 05/29/2008 SJ Web Edition
Revision Name	ofdm_mod
Top-level Entity Name	ofdm_mod
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Met timing requirements	No
Total logic elements	2,136 / 33,216 ( 6 % )
Total combinational functions	1,831 / 33,216 ( 6 % )
Dedicated logic registers	1,456 / 33,216 ( 4 % )
Total registers	1456
Total pins	131 / 475 ( 28 % )
Total virtual pins	0
Total memory bits	163,840 / 483,840 ( 34 % )
Embedded Multiplier 9-bit elements	0 / 70 ( 0 % )
Total PLLs	0 / 4 ( 0 % )

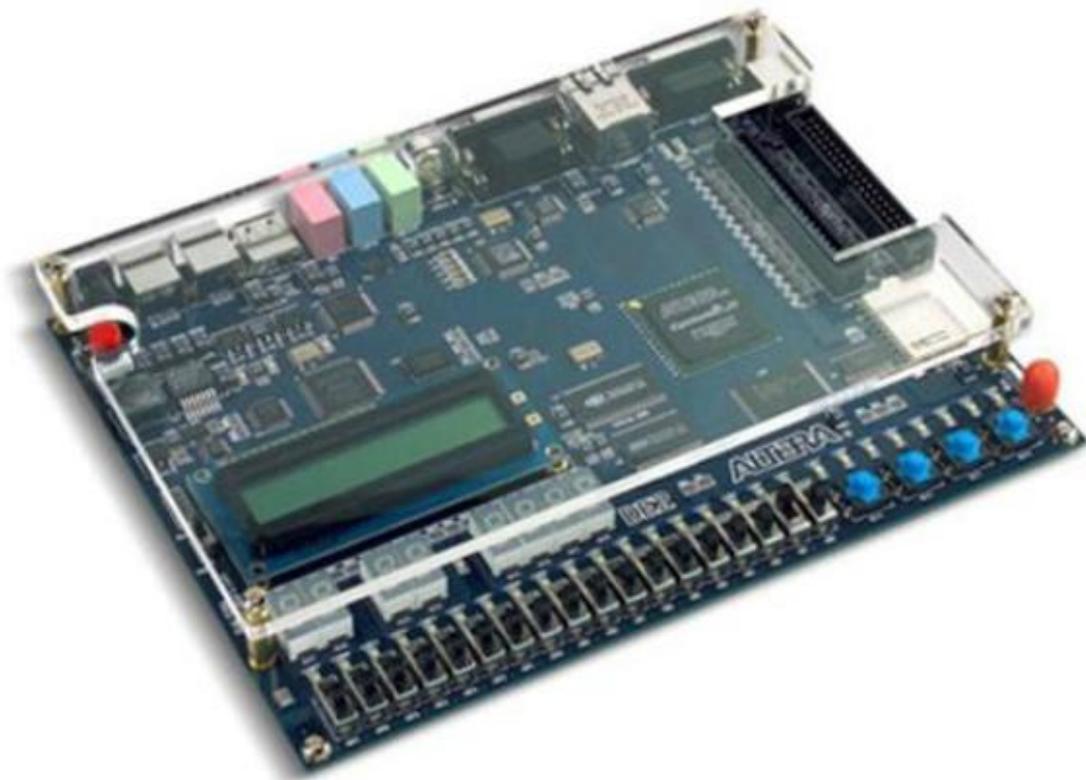
**Figura 29. Resultados de la Implementación.**

Se puede apreciar que se han utilizado un total de 2136 Elementos Lógicos (EL), lo cual implica el uso de únicamente 6 % de toda la cantidad de EL dentro del FPGA. Así también, se usaron un total de 1456 Registros Lógicos Dedicados, lo cual viene a ser solamente el 4% de uso de Registros. Estos 2 resultados iniciales demuestran que el diseño ha sido eficiente y enfocado básicamente en disminuir el consumo de potencia, pues cuanto menor sea la cantidad de recursos utilizados, menor será la potencia consumida por el dispositivo.

En cuanto al uso de memoria dentro del dispositivo, como se pudo intuir desde la etapa de descripción de su arquitectura, se obtuvieron mayores porcentajes de consumo con respecto al resto de parámetros. Las memorias FIFO y ROM usadas en la arquitectura fueron dimensionadas de tamaños considerables, lo que ocasionó que el resultado de la síntesis arroje un uso de 163840 bits de memoria (34% de recursos utilizados).

Finalmente, al evaluar los resultados del análisis *Timing* del Quartus II 8.0 Web Edition, se pudo encontrar que la frecuencia mínima de funcionamiento del sistema es 144.13 MHz. Esta limitación no afecta al diseño pues como se mencionó anteriormente, el reloj del sistema posee una frecuencia de 200 MHz. Del mismo modo, se pudo comprobar que la latencia máxima del sistema, evaluada una vez ya culminada la etapa de toma de datos, es de 19.23 ns.

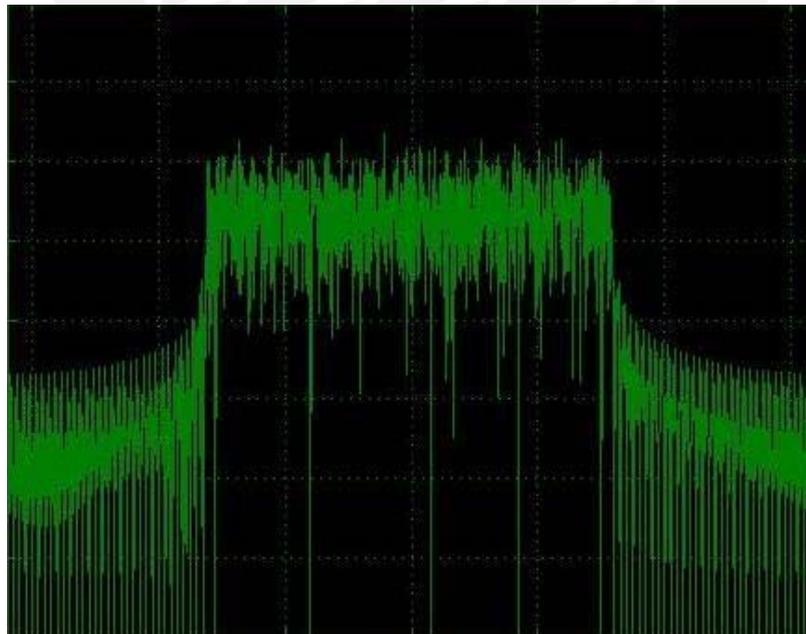
Una vez culminada la etapa de síntesis del circuito, se procede a su implementación física sobre la tarjeta de desarrollo, la cual se puede apreciar en la Figura 30.



**Figura 30. Tarjeta de Desarrollo DE2.**

Como se puede apreciar, esta tarjeta de desarrollo no posee ningún conector coaxial o SMA para poder analizar alguna señal analógica. De hecho, solamente cuenta con un circuito integrado que contiene 3 DACs, y es usado para el puerto VGA. Este circuito integrado es el AD7123 y es usado únicamente para las aplicaciones de vídeo; por ende, los 2 DACs tienen baja resolución (10 bits). Sin embargo, esos convertidores eran la única alternativa para obtener señales analógicas, por lo que se tuvo que adaptar el código para el uso de estos convertidores. Se disminuyó el ancho de palabra de la salida de los DDSs a 7 bits para que así se pueda convertir sin problemas la señal digital de salida. Finalmente, a través de un pin del puerto VGA se puede conectar una punta de osciloscopio y evaluar la señal obtenida. Cabe mencionar también, que la tarjeta de desarrollo DE2 no cuenta con un oscilador interno de 200 MHz, mas si con uno de 50 MHz. No obstante, ya que el FPGA cuenta con PLL embebidos, a partir de la señal de 50 MHz se puede generar una señal de reloj de 200 MHz con lo que ya se podría trabajar sin problemas. Por esto, además todo el circuito implementado, fue necesario el uso de un PLL y el manejo del integrado AD7123.

El resultado de la implementación en hardware se muestra en la Figura 31. Se puede apreciar el espectro de la señal OFDM, con algunas presencias de espurias, que deberían ser eliminadas por las etapas analógicas (filtros, amplificadores) posteriores al circuito implementado aquí. Sin embargo, se puede apreciar en el mismo la presencia de las 8 portadoras, por lo que sí se puede considerar como un buen resultado.



**Figura 31. Espectro de la señal OFMD.**

Se pudo medir en la señal el SFDR, y se obtuvo un valor de 63 dBc, lo cual es relativamente aceptable para un DAC de 10 bits. Sin embargo, cabe mencionar que este resultado se podría mejorar no solamente mediante el uso de DACs de mayor resolución, si no también mediante la implementación de filtros digitales internos que logren dar una mejor forma al espectro.

Finalmente, para poder dar un mejor veredicto a los resultados de la implementación se hizo una comparación de esta implementación con una basada en el uso de algoritmos de FFT para la generación de la señal OFDM. Así también, es necesario mencionar que la implementación basada en la FFT uso un DSP en lugar de un FPGA [13]. Esta implementación fue realizada por el Georgia Institute of Technology en el año 2004 y fue un gran punto de partida para el desarrollo en la tecnología SDR. La Tabla 2 muestra los resultados obtenidos por ambas implementaciones.

**Tabla 2. Comparación de Resultados.**

Parámetros	Basada en FFT-DSP	Basada en DDS-FPGA
Frecuencia de Reloj	200 MHz	200 MHz
Rango IF	DC to 80 MHz	DC to 50 MHz
SFDR	75 dBc	63 dBc
Tasa de Transferencia	30 Mbps	100 Mbps

Se puede apreciar que ambos sistemas funcionan con la misma Frecuencia de Reloj, y sin embargo, la implementación basada en el uso de FFT-DSP posee un mayor rango IF. Del mismo modo, el SFDR alcanzado por la implementación basada en la FFT fue mayor al alcanzado por el trabajo descrito por esta tesis. No obstante, la principal causa de esta deficiencia fue la carencia de un DAC de alta resolución para la implementación, por lo que se pudo mejorar este aspecto cuando se tenga a la mano un mejor hardware. Finalmente, se puede apreciar que esta implementación alcanza una mayor tasa de transferencia para la señal de entrada. Esto es un gran logro pues lo que se quiere es que este dispositivo pueda converger con las tecnologías de comunicación ya existentes, como lo es por ejemplo las redes Ethernet de 100 Mbps. De esta manera, con trabajos posteriores, el dispositivo podría estar ubicado en cualquier lugar, y podría ser manipulado desde un terminal remoto.

## CONCLUSIONES

1. El diseño y la implementación del Modulador OFDM Reconfigurable ha sido presentado. La flexibilidad del dispositivo, la poca cantidad de recursos utilizados y el buen nivel de calidad de señal que entrega el dispositivo comprueba de manera experimental la eficiencia en el diseño de la arquitectura implementada.
2. El uso de FPGAs permite alcanzar mayores niveles de velocidad de procesamiento debido a su gran nivel de paralelismo en su arquitectura interna, y a su capacidad de poder procesar señales tanto en banda base como en banda intermedia (hasta 300 MHz aproximadamente). Estas prestaciones lo hacen un elemento clave para la implementación de sistemas SDR.
3. La arquitectura planteada se presenta como un esquema modular, por lo que sus bloques internos fácilmente pueden ser reutilizados para posteriores diseños. Del mismo modo, esta característica de modularidad permite que el sistema implementado pueda ser utilizado como bloque interno en un sistema mucho más complejo, por lo que agiliza las etapas de diseño. Esta característica lo convierte al circuito en un dispositivo portable.
4. Las ventajas de los FPGAs sobre los DSPs son notables en velocidad, rango de frecuencias y modularidad de diseño. Adicionalmente, la ventaja de usar códigos VHDL sintetizados sobre FPGAs es la posibilidad de poder expandir los horizontes de la implementación, apuntando inclusive al diseño de un circuito integrado. Gracias a herramientas CAD, a partir de códigos VHDL se puede generar el layout completo de un circuito integrado, agilizando notablemente el flujo de diseño; cosa que no se puede lograr con un DSP.
5. La señal OFDM obtenida del dispositivo contiene las componentes de frecuencia requeridas para su correcta detección en el receptor. Sin embargo, la presencia de ruido

en la misma no es tan insignificante, pues se comprobó que el nivel de SFDR es únicamente aceptable, lo que derivará a más estudios para anular esta deficiencia.

6. El circuito descrito permite un gran nivel de reconfiguración y una rápida adaptación a los cambios en los requerimientos en la señal, por lo que conforma también un punto de partido para mayores investigaciones en el área de Acceso Dinámico al Espectro y Radios Cognitivos.



## RECOMENDACIONES

El presente trabajo tuvo como objetivo la implementación de un Modulador OFDM Reconfigurable en hardware. Las etapas previas al modulador, como la codificación de la señal en banda base, no fueron abarcadas en este trabajo. Por ende, para la etapa de implementación se tuvo que describir un circuito adicional que esté mandando constantemente datos a la entrada del modulador, simulando así la etapa previa. Esto fue lo más eficiente pues se necesitaban una considerable cantidad de pines (131), y el manejo de los mismos hubiera sido engorroso. Esto hubiera derivado principalmente en el retraso en la etapa de pruebas. En consecuencia, siempre es aconsejable analizar las alternativas al alcance para la verificación del funcionamiento del circuito, escogiendo siempre las que permiten un buen análisis de resultados y agilicen el proceso de pruebas.

Un posible trabajo posterior a este sería la implementación de este circuito mediante la tecnología VLSI para poder producir un circuito integrado. Esto es más que factible, pues casi todos los circuitos descritos en VHDL son genéricos. Sin embargo, el uso de algún módulo interno de circuitería del FPGA en uso anula todas las posibilidades del diseño VLSI, pues las herramientas CAD actuales solamente son capaces de sintetizar códigos VHDL o Verilog genéricos para la generación de layouts de circuitos integrados. Por tanto, si es que si es un objetivo concreto el diseño VLSI, se debe evitar por completo bloques internos de circuitería del FPGA, para así poder integrar el diseño con las herramientas CAD.

La señal obtenida del modulador presentó una cierta cantidad de ruido que se debe tratar de eliminar o aminorar al máximo. En consecuencia, siempre es recomendable colocar etapas de filtrado y amplificación a la salida de cualquier DAC para poder obtener una mejor calidad de señal. Sin embargo, se mencionó anteriormente que parte de esta falencia se debía a que el hardware disponible no era el idóneo. De manera más específica, se necesita un DAC de alta resolución para poder generar una señal con poca cantidad de espurias, por lo que siempre se debe procurar tener todo el hardware apropiado para poder realizar las pruebas sin problemas.

Por otro lado, también se mencionó la posibilidad de colocar filtros digitales dentro del sistema para contrarrestar este efecto. Esta posibilidad es muy buena y de hecho es la que generalmente se usa en los transmisores actuales. Sin embargo, siempre hay que tomar en cuenta que el uso de filtros siempre induce una mayor latencia al sistema e involucra una mayor cantidad de consumo de recursos. Por tanto, se debe procurar evaluar cuales son los objetivos principales del sistema a implementar para así poder sopesar la posibilidad de colocar filtros digitales, aumentar la resolución del DAC, o colocar mayor cantidad de circuitería analógica de acondicionamiento de señal.



## BIBLIOGRAFÍA

- [1] Buracchini, E., “The software radio concept”, *IEEE Communications Magazine*, Vol. 38, no. 9, pp. 138 – 143, Sept. 2000.
- [2] Walt Kester, and A. Abidi, “Data Conversion Handbook,” Burlington MA: Newnes, 2005, p. 67.
- [3] Vankka, J., “Digital Synthesizers and Transmitters for Software Radio”, Helsinki University of Technology, Springer: 2005, pp. 61 – 65.
- [4] Canyan Zhu; “Research on quadrature transformation structures for SDR receiver”, *Proc. IEEE Int. Conf. Communications, Circuits and Systems*, Vol. 2, pp.
- [5] Xinyu Xu; Bosisio, R.G.; Ke Wu.; “Analysis and implementation of software defined radio receiver platform”, *Proc. IEEE Int. Conf. Asia-Pacific Microwave*, Vol. 5, pp. 4-, Dic. 2005.
- [6] Cass, S., “Tools & Toys: Hardware for your Software Radio”, *IEEE Spectrum Magazine*, Vol. 43, no. 10, pp. 51 – 54, Oct. 2006.
- [7] Minden, G.J.; Evans, J.B.; Searl, L.S.; DePardo, D.; Rajbanshi, R.; Guffey, J.; Qi Chen; Newman, T.R.; Petty, V.R.; Weidling, F.; Peck, M.; Cordill, B.; Datla, D.; Barker, B.; Agah, A.; “An Agile Radio for Wireless Innovation”, *IEEE Communications Magazine*, Vol. 45, no. 5, pp. 113 – 121, Mayo 2007.
- [8] Ye G. Li, Stuber L. G., “Orthogonal Frequency Division Multiplexing for Wireless Communications”, Georgia Institute of Technology, Springer: 2006, pp.8 -10.
- [9] John G. Proakis, Dimitris G. Manolakis; “Digital Signal Processing: Principles, Algorithms and Applications”, Prentice Hall: 1998.
- [10] Heideman, M.; Johson, D; Burrus, S; “Gauss and the History of the Fast Fourier Transform”, *IEEE ASSP Magazine*, Vol. 1, no. 4, pp. 14 – 21, Oct. 1984.

- [11] Yuan-Pei Lin; See-May Phoong; “OFDM transmitters: analog representation and DFT-based implementation”, *IEEE Trans. Signal Processing*, Vol. 51, no. 9, pp. 2450 – 2453, Sept. 2003.
- [12] Helaoui, M.; Boumaiza, S.; Ghazel, A.; Ghannouchi, F.M.; “On the RF/DSP design for efficiency of OFDM transmitters”, *IEEE Trans. Microwave Theory and Techniques*, Vol. 53, no.7, pp. 2355 – 2361, Julio 2005.
- [13] Weidong Xiang; Pratt, T.; Xudong Wang; “A software radio testbed for two-transmitter two-receiver space-time coding OFDM wireless LAN”, *IEEE Communications Magazine*, Vol. 42, no. 6, pp. S20 - S28, Feb. 2001.
- [14] Xilinx Co. (2003, December), *Digital Up-Converter v1.0*, [Online] Available: <http://www.xilinx.com/ipcenter/catalog/logicore/docs/duc.pdf>. [descargado el 20/07/2007].
- [15] Shono, T.; Shirato, Y.; Shiba, H.; Uehara, K.; Araki, K.; Umehira, M.; “IEEE 802.11 wireless LAN implemented on software defined radio with hybrid programmable architecture”, *IEEE Trans. Wireless Communications*, Vol. 4, no. 5, pp. 2299 - 2308, Sept. 2005.
- [16] <http://wireless.per.nl/reference/chaptr05/ofdm/ofdmmath.htm> [visitado el 15/05/2008].
- [17] Robles, O.; Silva, C; “Design and Implementation of a Reconfigurable OFDM Modulator for Software-Defined Radios”, *Proc. IEEE Int. Conf. 4th Southern Programmable Logic*, pp. 241-244, Abril 2008.
- [18] [http://www.memso.bio.sld.cu/habana2003/Articles/T\\_0034.pdf](http://www.memso.bio.sld.cu/habana2003/Articles/T_0034.pdf) [visitado el 02/08/2007].
- [19] Uusikartano, R.; Nittylahti, J.; Renfors, M.; “Area-optimized FPGA implementation of a digital FM modulator”, *Proc. IEEE Int. Conf. Circuits and Systems*, Vol. 4, pp. 360 – 362, 30 de Mayo - 2 de June 1999.
- [20] Wayne Tomassi, “Sistemas de Comunicaciones Electrónicas”, México: Pearson Educación, 2003, p. 476.