UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO DE UN RECEPTOR DIGITAL DE SEÑALES DE RADIOFRECUENCIA MEDIANTE UN FPGA.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO ELECTRÓNICO

PRESENTA:

GABRIEL AUGUSTO ZEBADÚA GARCÍA.

DIRECTOR DE TESIS:

DR. PABLO ROBERTO PÉREZ ALCÁZAR.

MÉXICO D.F, 2012.

A mis padres…

“L'homme n'est rien d'autre que ce qu'il se fait"
El hombre no es otra cosa que lo que hace de sí mismo.

Jean Paul Sartre.

Agradecimientos

En primer lugar, deseo expresar mi mas profundo agradecimiento a mi director de tesis, el Dr. Pablo Perez Alcazar , por considerarlo una de las personas que más marcarón mi formación como ingeniero. El profesionalismo, la pasión por su trabajo y la excelencia con la que se desempeña son de las cosas mas valiosas que logre encontrar en la facultad. Le doy las gracias, doctor, por mostrarme con su ejemplo lo que puede esperarse de un profesional y por abrirme las puertas a campos de conocimiento nuevos. Sin su dirección, este trabajo de tesis no hubiera sido posible.

Agradezco a mis sinodales por los aportes que realizaron a este trabajo , en particular al Dr, Rogelio Alcantara Silva por sus consejos y aportes que mejoraron de manera importante el contenido de este trabajo de tesis.

Mencion especial merece la colaboración de la profesora Norma Elva Chavez Rodriguez quien tuvo a bien ofrecer la tarjeta de desarrollo que permitió llevar a cabo este trabajo.

Doy gracias a mi *alma mater* por ofrecerme las condiciones para lograr una transformación profunda durante mis estudios de licenciatura y a todos los profesores que de alguna manera cambiaron mi forma de pensar y fueron importantes en mi formación, sobre todo por el profesionalismo con el que se desempeñan: Susana Aburto, Fernando Sanchez Rodriguez, Juan Velázquez , Gabriel A. Jaramillo, Patricia Hong, Victor Pinilla, Rafael Iriarte., Carlos Rivera, Gerardo Espinosa, Moises Rueda, Roberto Macias, Lauro Cruz y Laura Oropeza.

A mi padre, por haberme ofrecido todo cuanto necesite durante mi carrera, por haber sabido hacerme llegar las palabras que yo necesitaba en cada momento y por mostrarme con su ejemplo, el camino a buscar la excelencia y el profesionalismo, mi mas profundo amor, respeto y admiración para él.

A mi madre por su ejemplo de esfuerzo y de perseverancia, por los valores con los que me formó y por creer siempre en que podia enfrentar grandes retos y superarlos.

A Belena, mi novia y mejor amiga, por ser la persona de quien mas recibi muestras de apoyo, y aliento. Gracias por darme la motivación para desarrollar esta tesis y para dar lo mejor de mi en ella, por tu entrega y esfuerzo en cada cosa que haces. Eres de lo más valioso que hay en mi vida.

A mis hermanos, Julio por ser un ejemplo para mi, de arrojo, de valentía y de la imporancia de creer uno mismo, y A Ilse, por su carácter critico y su aliento a buscar siempre hacer bien las cosas. A pesar de nuestras diferencias, es un orgullo para mi que sean mis hermanos, y han hecho de mi una mejor persona.

A mis compañeros en los primeros semestres con quienes pude compartir alegrias y tragos amargos durante mi estancia en la facultad y de quienes aprendi muchas cosas, Arturo, Ben, Alberto E, Alberto H, Humberto , Rafael, Victor. Muchas gracias amigos.

A quienes conoci en el camino y quienes lograron contagiarme con su entusiasmo , y su constante busqueda de la excelencia: Luis Melchor, Vicente Hernandez. , Martin Rodiguez, Luis Rodiguez, Alejandro Vaca, Ismael Castillo, Javier Hernandez, Fernando Leguizamo, Ricardo Granados, Hugo Montenegro y Enrique Velázquez.

A Fernando Solis, por sus consejos y su guia a través de mi estancia en al facultad.

A Leonardo Garcés y Marco Serna, por su interes en este proyecto, sus muestras de apoyo y por la gran cantidad de comentarios que dieron sobre esta tesis, mismos mejoraron, sin duda, el resultado final de la misma, ambos amigos entrañables y excelentes seres humanos.

A todos aquellas personas que no menciono y con las que me encontre a lo largo de la carrera quienes de alguna manera aportaron algo en mi camino.

Finalmente, al lector, por interesarse en este trabajo, esperando que el tiempo que se ha inveritdo en esta tesis pueda serle de utilidad.

JURADO ASIGNADO.

PRESIDENTE. Dr. Francisco Javier García Ugalde.
VOCAL. Dr. Pablo Roberto Pérez Alcazar.
SECRETARIO. Dr. Rogelio Alcántara Silva.
1er SUPLENTE. M.I. Ricardo Mota Marzano.
2do SUPLENTE. M.I. Norma Elva Chavez Rodriguez.

I| Indice general

RESUMEN.

Esta tesis presenta y evalúa diversas tecnicas y herramientas para la implementación de un receptor digital de señales. Se destaca en ella su importancia y las ventajas que ofrece respecto a un esquema analógico de recepción. Se hace enfasis en las ventajas del uso de dispositivos con arquitectura configurable como los FPGA en la implementación, asi como en el uso de muestreo paso-banda, que permite muestrear a frecuencias menores a la de Nyquist. Sobre esta tecnica se presentan una serie de consideraciónes para su uso en receptores digitales asi como algunos criterios que permiten elegir la frecuencia de muestreo paso-banda. Además, se propone un analisis que conduce a la predicción de la ubicación en frecuencia de la información de interes despues de emplear este tipo de muestreo, este analisis se acompaña de programas en matlab que permiten efectuar esta tarea de manera automatica. Cabe destacar que la bibliografía no ofrece hasta ahora ninguna herramienta similar, por lo que este analisis constituye un avance importante en la aplicación de muestreo paso-banda, la cual no se limita a los receptores digitales de señales.

En lo que respecta a los bloques que componen el receptor se propone un esquema simple que hace uso de filtros Hogenauer y filtros FIR, considerando las características y limitantes de una tarjeta de desarrollo disponible, la Spartan 3 de Xilinx. Se buscó que el esquema fuése lo mas general posible, de tal manera que permite ser ajustado a las necesidades de cada usuario.

Las pruebas planteadas para validar el desempeño de dicho esquema, son con señales paso-banda con frecuencia central en 20 [Mhz] y ancho de banda de 50[Khz] y con 200[Mhz] y anchos de banda de 50 y 700[Khz]. Las señales son simuladas con ayuda de matlab para generar la salida que entregaría el convertidor analógico digital y con ellas se alimenta el FPGA. El muestreo se efectua a 4.2[Mhz] en la señal de 20[Mhz] y a 42[Mhz] para las señales de 200[Mhz], lo que permite el procesamiento con el FPGA, capaz de operar hasta los 50[Mhz].

Los resultados obtenidos demuestran validez del análisis y del esquema planteado. A pesar de ser un esquema con filtros computacionalmente económicos, ofrece tolerancia media al ruido (permitiendo manejar SNR de hasta 10[dB]).

Capítulo 1

Introducción.

1.1 Antecedentes.

1.1.1 Motivación.

En los últimos años, el número de sistemas electrónicos que utilizan un elemento de transmisión y recepción de señales eléctricas en el rango de radiofrecuencia ha aumentado, pudiéndose mencionar como ejemplo, los sistemas de obtención de imágenes y de espectroscopia por Resonancia Magnética Nuclear [Wright 97], empleados en medicina y biología para extraer información estructural o química de una muestra en forma no invasiva, o algunos dispositivos MEMS usados como sensores (de tamaño micrométrico) en diversas áreas de la ciencia y la ingeniería. Lo anterior muestra que la aplicación de este esquema, va mucho más allá de los sistemas convencionales de telecomunicaciones.

En el caso de los dispositivos MEMS, existe un interés creciente en la incorporación de antenas en su interior, con el objetivo de trasladar la parte de procesamiento de la señal al exterior del dispositivo, y reducir con ello el tamaño y consumo de potencia del mismo [Varadan 01].

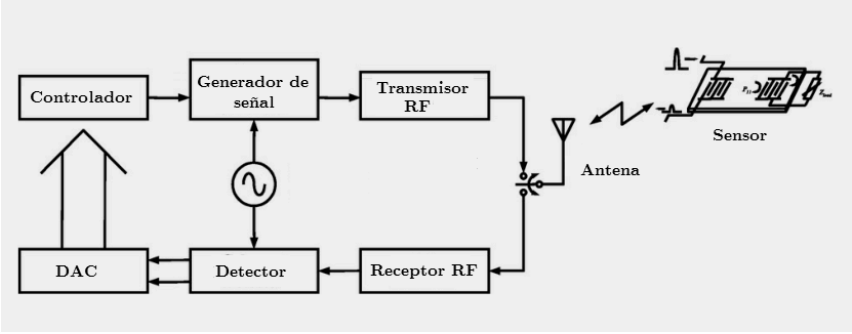

Fig. 1.1A Recepción de señales en dispositivos MEMS [Reindl 01]

La figura 1.1A presenta un esquema genérico en donde se ilustra la interacción entre el sistema de procesamiento, la antena y el dispositivo MEMS.

El desarrollo de este enfoque se ha visto motivado debido a la variedad de magnitudes físicas que pueden ser medidas a través de estos dispositivos. Entre ellas destacan: temperatura, presión, momento, aceleración o humedad [Reindl 01]. En la figura 1.1B se muestra un acelerómetro MEMS que cuenta con una antena para transmitir señales.

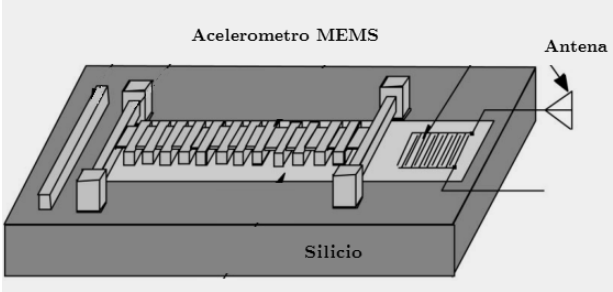

Fig. 1.1B Acelerómetro MEMS con sistema de transmisión de señales [Varadan 01].

En los dispositivos de transmisión inalámbrica, es importante hacer uso de señales de alta frecuencia, lo que permite reducir el tamaño de los componentes físicos que las manejan[1]. En muchos casos, las señales que se transmiten tienen ancho de banda reducido y son de tipo paso-banda[2], este trabajo se orienta al manejo de señales con dichas caracterisicas.

---

[1] Las antenas empleadas para la transmisión de señales, requieren tener una longitud de al menos $\lambda/4$, siendo λ la longitud de onda de la señal a transmitir. La cual está dada como: $\lambda=v/f$, siendo v su velocidad de propagación y f su frecuencia. De dicha expresión puede observarse que un aumento en la frecuencia reduce el tamaño requerido para la antena, lo que ha motivado el manejo de señales cada vez con mayor frecuencia.

[2] Estos conceptos se discuten en el en capitulo 2.

A lo largo de más de 75 años, la electrónica analógica ha brindado herramientas para realizar la recepción de señales, haciendo uso exclusivamente de componentes analógicos en las tareas involucradas: amplificación, demodulación, filtrado, generación de señales y cualquier otro proceso requerido sin importar si la señal de interés fuese de tipo analógico o digital [Nagurney09]. Por lo anterior resulta reelevante presentar una breve discusión sobre sus componentes, para resaltar algunas características y desventajas asociadas con este enfoque.

1.1.2 Receptores analógicos.

Un receptor analógico de señales (RAS) está constituido, de manera general, por los componentes mostrados en la figura 1.1C, los cuales permiten realizar la tarea de recepcion. Dicho receptor está conformado por una antena, que permite recibir las señales, conectada a un amplificador sintonizado, el cual amplifica solo algunas componentes de la señal.

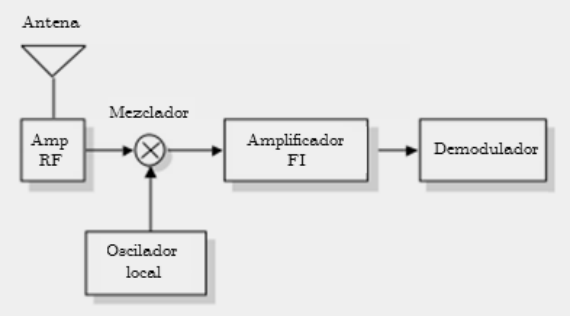

Fig. 1.1C Estructura de un receptor analógico

La utlidad de este bloque esta asociada a dos aspectos importantes que se deben destacar:

1) Durante su propagación, las señales eléctricas adquieren ruido como parte del proceso de transmisión y el amplificador permite minimizar su efecto en la señal, al ofrecer ganancia de manera selectiva unicamente a las componentes en frecuencia que resulten de interes.

2) La antena siempre recibe una gran cantidad de señales eléctricas que inciden en ella, por lo que es necesario separar aquella que resulte de interés del resto, antes de ser procesada.

Despues del amplificador sintonizado, se emplea un mezclador de señales, compuesto por un oscilador, que genera una señal senoidal de frecuencia intermedia, y un multiplicador. Su objetivo es obtener el producto de la señal generada localmente y la salida del amplificador sintonizado. Con lo anterior se logra centrar el espectro de la señal recibida en el rango de frecuencia intermedia [Briham88]. Veasé la figura 1.1D.

Fig. 1.1D Proceso de traslación de una señal de RF a frecuencia Intermedia.

Al desplazar el espectro a frecuencia intermedia, es posible hacer uso de componentes electrónicos operando a una frecuencia menor, y como resultado, obtener un aumento en la precisión que puede obtenerse al operar a una frecuencia mas baja y una reducción en el costo por las mismas razones. A *priori* puede pensarse que un traslado del espectro directamente a banda base representaría una opción más viable, no obstante, *la selectividad de un filtro analógico depende en proporción directa de la frecuencia de resonancia*, por lo que el traslado a banda base se traduce en un deterioro en la capacidad para separar la señal. Así pues, el rango de frecuencia intermedia ofrece economía y rendimiento de manera simultanea y por ende, es empleado en la mayoría de estos sistemas.

Después, a la salida del mezclador, se utiliza un filtro de frecuencia intermedia y banda de transición angosta, que permite eliminar la mayor parte de las componentes no deseadas del espectro de la señal. Finalmente, se realiza el proceso de demodulación analógica, según el tipo de modulación que haya sido empleado para transmitirla.

Una desventaja importante de estos receptores, aparece al hablar de flexibilidad, si bien es cierto que permiten ajustar los valores de sus componentes para modificar el tipo de señal que pueden procesar, es importante mencionar que esto debe hacerse manualmente, lo cual, aunado a las tolerancias presentes en componenten analógicos, ofrece una fuente importante de imprecisión, dando lugar a diferencias significativas entre las características deseadas para el receptor y las que se obtienen al reconfigurarlo.

Además de ello se pueden mencionar dos inconvenientes importantes adicionales:

1) El proceso de diseño suele ser complejo y sujeto a muchas fuentes de imprecisión, lo que hace que no pueda ser repetido fácilmente.

2) Durante su operación, exhiben variaciones que generalmente no están bajo el control del diseñador, tales como las condiciones ambientales o el envejecimiento y desgaste natural, los cuales producen variaciones en los valores de los componentes y conducen a errores al procesar la señal.

Por lo anterior, se ha hecho necesario encontrar enfoques diferentes que permitan resolver estos inconvenientes. Una alternativa, que se ha ido consolidando con el paso de los años y ha motivado este trabajo de tesis, debido a la gran aceptación que ha encontrado, es presentada a continuación.

1.1.3 Receptores digitales.

Este enfoque, surgió como conscuencia del desarrollo de la tecnología de los circuitos integrados, ya que gradualmente se introdujeron componentes digitales en un esquema de recepción analogica. En un principio solo permitian la generación de señales, codificación o decodificación de información en forma digital, hasta que hace poco más de dos décadas, se propuso por primera vez la implementación de un receptor de señales digital (RDS)[Harris04].

Con los componentes digitales necesarios, estos receptores podrían tomar la señal analogica directamente a la salida del amplificador sintonizado para realizar la conversión analógica-digital, y después efectuar las operaciones de demodulación y procesamiento de los datos a través de algún algoritmo almacenado en un dispositivo de procesamiento digital. Esta alternativa, ofrece un proceso más simple, económico y versátil, ademas de un mayor control sobre el proceso de diseño, una mayor estabilidad durante su operación y la capacidad de repetir el diseño de manera muy simple, únicamente cargando el *software* en un nuevo dispositivo.

Asimismo, permite la reconfiguración del sistema para poder procesar cualquier otra señal de RF, sin importar el tipo de modulación empleado o el formato en que se encuentren los datos y, con ello, ofrecer un sistema de recepción universal, capaz de permitir la interoperabilidad entre sistemas con rango de frecuencias y modulaciones distintas [Nagurney09][Mcloskey84].

Por otra parte, este planteamiento permite la posibilidad de realizar algunas funciones que no están al alcance de un receptor analógico. La más importante es que, al poder reconfigurar o actualizar los RDS por medio de software, se tiene la posibilidad de hacerlo de manera remota.

A pesar de las ventajas presentadas, realizar un sistema digital, que fuese capaz de operar en tiempo real, hasta hace un par de décadas resultaba muy demandante en costo, tamaño del dispositivo y fuera del alcance de los componentes digitales, en lo que a velocidad de operación se refiere. Actualmente, los componentes digitales han tenido un desarrollo importante y ofrecen características más favorables, permitiendo que la implementación de un sistema de recepción digital resulte una opción más atractiva que en ese entonces.

Si bien la tecnología actual en convertidores analógicos a digitales (CAD) no permite realizar muestreos a todas las tasas requeridas para señales de RF, es posible, en el caso de señales de RF limitadas en banda y un espectro de magnitud de tipo paso-banda, hacer uso de técnicas de submuestreo o muestreo cuadratura, que permiten reducir la frecuencia de muestreo se reduce por debajo de la frecuencia Nyquist, permitiendo así reducir significativamente la tasa de muestreo a la que el CAD debe operar [Mcloskey84]. Este hecho presenta al receptor digital de señales como una alternativa conveniente, en este trabajo se pretende explorar dichas tecnicas para hacer uso de las mismas en caso de que resulten favorables.

Además, día con día se desarrollan CADs con velocidades mayores, con mejor resolución y mayor precisión, lo cual ofrece un panorama alentador para la implementación de RDS. El desarrollo en CADs se debe a que se trata del componente más importante de un sistema de procesamiento digital, ya que el alcance del mismo siempre estará limitado a la velocidad de operación y al ancho de banda ofrecidos por los mismos.

Para la implementación de un receptor digital de señales (RDS), es posible hacer uso de diferentes enfoques, ya sea trabajando con computadoras, microcontroladores de propósito general o de procesamiento digital de señales.

Cualquiera de ellos permite realizar las operaciones elementales, aritméticas y lógicas, necesarias para el procesamiento.

Sin embargo, contienen una cantidad limitada de unidades de procesamiento, casi siempre concentradas y controladas por un mismo elemento, lo que obliga a que cada operación tenga que esperar su turno para ser identificada y enviada al bloque capaz de ejecutarla.

Teniendo en cuenta lo anterior, se ha diseñado hardware especializado, conocido como FPGAs (*Arreglos de lógica programable en campo)*. Se trata de dispositivos hechos a base de compuertas lógicas, integradas en un solo encapsulado, las cuales pueden ser "alambradas" a través de herramientas de síntesis (descripción de hardware), de acuerdo a las necesidades del diseñador, para cumplir con alguna función en particular, permitiendo desarrollar, entre otras funciones, las operaciones necesarias en procesamiento digital de señales. Estos arreglos de compuertas resultan convenientes en PDS, ya que permiten adoptar un enfoque diferente: en lugar de tener una sola unidad capaz de realizar un gran número de operaciones, se crean pequeños bloques operarando simultáneamente, consiguiendo con ello emplear el paralelismo y reducir el tiempo de espera de las operaciones involucradas en el procesamiento. Lo anterior se traduce en un incremento en la cantidad de datos que pueden procesarse por unidad de tiempo. Esto es vital en aplicaciones de alta frecuencia, puesto que estas señales obligan a emplear sistemas capaces de operar a grandes velocidades.

1.2| Objetivos.

Como se ha señalado antes, el incremento en el campo de aplicación de los sistemas inalámbricos de trasmisión de señales, el aumento en las velocidades de los componentes digitales y CADs, así como el nuevo hardware que ha sido desarrollado (FPGAs), hacen ver atractiva la idea de realizar un sistema de recepción de señales utilizando técnicas digitales.

Por esto, el objetivo del presente trabajo es desarrollar los elementos fundamentales que componen un receptor digital de señales. Para esto se han propuesto los siguientes objetivos intermedios:

1.- Análisis de los aspectos teóricos relacionados con el proceso de conversión analógica digital y los requerimientos para la implementación del sistema.

Se presentan elementos para guiar en la elección del tipo de muestreo, señalando las fuentes de imprecisión involucradas en el proceso de CAD, como son: variaciones en la velocidad de muestreo (jitter) y error de cuantización inherentes al proceso.

2.-Discusión de las técnicas de procesamiento digital de señales disponibles para trabajar con señales de alta frecuencia, involucrando el procesamiento de tasa múltiple y algunas de las estructuras y técnicas de diseño convenientes para receptores digitales, que están disponibles en la literatura.

3.- Seleccionar , tomando en cuenta la discusión presentada, las herramientas, técnicas o estructuras más convenientes para la implementación del receptor digital en un dispositivo FPGA.

4.- Implementación de los bloques básicos de un receptor digital generico para operar directamente en RF, mediante un dispositivo FPGA,

describiendo el hardware con lenguaje VHDL, destacando las limitantes presentes y los alcances logrados.

1.3 Estructura de la tesis.

En este capítulo de introducción se ha tratado de situar el trabajo de tesis, la importancia que el sistema de recepción de señales tiene y algunas de las nuevas áreas que pueden verse beneficiadas con la implementación de este tipo de diseños.

Se ha presentado el esquema general de recepción analógica de señales, y algunos de sus inconvenientes (tolerancias de sus componentes, vulnerabilidad a las condiciones ambientales, etc.), al igual que las ventajas y mejoras asociados con la recepción digital de señales, (estabilidad, menor costo, ajuste o reprogramabilidad en forma remota e inmunidad al medio y al paso del tiempo) con la idea de justificar el interés en los receptores digitales.

En el capítulo 2, se presenta la teoría para el desarrollo de este trabajo, repartida en 4 secciones.

La primera de ellas se orienta a la teoría básica del procesamiento digital de señales, la estructura básica de un sistema de este tipo y a una descripción sobre las ventajas y limitantes ofrecidas con respecto al procesamiento analógico. Además, presenta un resumen de las técnicas de muestreo útiles cuando se requiere procesar señales en el rango de radiofrecuencias.

La segunda sección se centra en presentar la estructura básica de un receptor digital de señales, los bloques que lo conforman y la función que cada uno realiza, así como algunas arquitecturas que han sido propuestas y sus limitantes, asi como las ventajas asociadas al uso de procesamiento multitasa. En su tercera parte, este capítulo ofrece una síntesis de las técnicas y estructuras de *procesamiento digital a tasa múltiple*, las cuales permiten reducir la velocidad de muestreo y flexibilizar las restricciones que deben imponerse sobre el hardware. En la última

sección, se presenta un análisis del hardware que puede emplearse en un RDS, realizando una breve comparación entre los microprocesadores de propósito general, los procesadores digitales de señales y los FPGA, recalcando las ventajas y desventajas de cada uno de ellos con la idea de justificar la orientación de este trabajo hacia los arreglos de compuertas programables.

El tercer capítulo describe, el enfoque a seguir en este trabajo, se parte de una tarjeta de desarrollo fabricada por Xilinx (Spartan 3e), destacando sus características y alcances, permitiendo con ello realizar un diseño de un RDS capaz de operar en el hardware disponible, así como plantear una serie de pruebas con señales de diferentes características que pueden efectuarse en el hardware disponible.

El cuarto capítulo presenta las pruebas realizadas y los resultados obtenidos durante las mismas, mostrando la validez del diseño planteado y resaltando las capacidades del mismo.

Finalmente, el quinto y último capítulo de esta tesis presenta las conclusiones obtenidas, los alcances logrados y las sugerencias que se proponen para continuar sobre la misma linea de trabajo, pretendiendo con ello servir de base para trabajos futuros orientados al uso de FPGAs en aplicaciones de procesamiento digital de señales, haciendo enfasis en las consideraciones practicas que deben hacerse al implementar sistemas de PDS con este tipo de dispositivos.

Capítulo 2

Marco Teórico.

2.1 Procesamiento digital de señales

Una gran cantidad de variables fisicas que se presentan en los diversos fenómenos físicos que ocurren en la naturaleza, al igual que información de diversa índole, pueden ser representados a través de funciones matemáticas a las que se denomina *señales*. Cuando la función que permite representar al fenómeno esta definida para cualquier valor de la variable independiente, se dice que la señal es de tipo analógico. La mayoria de las señales que pueden ser encontradas en la naturaleza son de este tipo.

En la ingeniería electrica, las señales pueden ser respresentadas a traves de una diferencia de potencial que varia en el tiempo, lo cual posibilita hacer uso de sistemas eléctricos y electrónicos analógicos, tales como filtros, mezcladores o analizadores de espectros para trabajar con señales. Dichos sistemas, denominados de procesamiento analógico de señales, permiten realizar dos tareas básicas: cambiar las características de las señales o extraer la información contenida en ellas.

A pesar de que su uso es muy común, no es el único método disponible. El procesamiento digital de señales, (PDS), brinda una alternativa para realizar esta tarea, la cual esta basada en encontrar una representación de las señales analógicas a traves de bits almacenados en componentes digitales y realizar las operaciones necesarias sobre los datos en el dominio digital, buscando con ello ofrecer las ventajas asociadas a estos sistemas. Al hablar de sistemas de procesamiento analógico unicamente se hace referencia a componentes fisicos como capacitores, inductores, o resistencias, mientras que los sistemas de PDS están equipados no solo con hardware, sino también con un programa o *software* que realiza la tarea de procesamiento deseada. Almacenar la tarea de procesamiento de esta forma, permite reconfigurar las operaciones de procesamiento modificando únicamente los datos que se encuentran en localidades de memoria. Esta reconfiguracion es poco viable y dificil de implementar cuanto se trabaja con sistemas analogicos.

Además, el PDS permite eliminar las variaciones durante el diseño y operación de los dispositivos, ya que los componentes que integran un sistema digital de procesamiento no presentan tolerancias en sus valores, como ocurre con sistemas analógicos. Lo anterior permite realizar diseños más precisos de un sistema de procesamiento de señales [Proakis 07][Oppenheim 10].

Como ventaja adicional puede destacarse que un sistema analógico no permite realizar fácilmente operaciones matemáticas complejas. En contraste, los sistemas digitales posibilitan no sólo el desarrollo de cálculos matemáticos elaborados, sino que además hacen posible la implementación de algoritmos de tratamiento de señales que no pueden ser implementados en su contraparte analógica.

Por último, se puede mencionar que existe una ventaja de costo en el procesamiento digital debido a que:

i) El hardware requerido puede ser más barato y de menor tamaño.

ii) La flexibilidad que caracteriza a los sistemas digitales permite reprogramar a un dispositivo para realizar varias tareas, reduciendo la cantidad de componentes que conforman al sistema de procesamiento.

2.1.1 Etapas.

Las señales analógicas no pueden ser tratadas directamente con sistemas de procesamiento digital (computadoras, microprocesadores o algún otro tipo de dispositivo digital capaz de realizar las operaciones básicas necesarias), por lo que es necesario disponer, además del PDS, de una interface entre la señal analógica y el sistema digital. Este componente se denomina convertidor analógico digital (CAD). Como etapa final, algunos sistemas incluyen una interface adicional que permite transformar la señal digital, después de haber sido procesada, en analógica; este bloque recibe el nombre de convertidor digital analógico (DAC).

El esquema de un sistema genérico de PDS es presentado en la figura 2.1A.

A pesar de las ventajas asociadas a procesar señales digitalmente, su campo de aplicación se ha visto limitado por algunos parámetros, tales como la velocidad de muestreo de los CAD y de operacion del hardware de procesamiento de señales; así como, la frecuencia y el ancho de banda de las señales a procesar[1]. Por ello, el PDS depende directamente del estado del arte del hardware de procesamiento [Proakis 07]. Esta limitante hace necesario un analisis de los procesos involucrados en el CAD y de las estrategias que pueden adoptarse durante el proceso para relajar la demanda sobre el mismo. Este análisis que se presenta en la siguiente seccion de este capítulo.

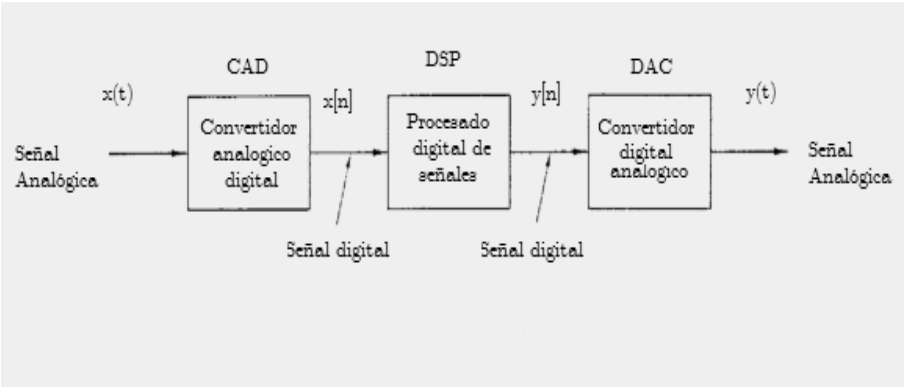

Fig. 2.1A Componentes de un sistema de procesamiento digital de señales.

2.1.2 Conversión analógica digital (CAD).

El objetivo de este proceso es, como se ha mencionado previamente, obtener la representación de una señal analógica que permita ser interpretada por hardware digital, lo cual se consigue al generar una secuencia numérica cuyos elementos expresan la magnitud de la señal en algunos instantes de tiempo [Lyons 10]. Estos valores se denominan muestras, las cuales pueden estar o no uniformemente espaciadas en el tiempo.

[1]En el caso de señales de banda ancha, el CAD debe operar a una velocidad de muestreo elevada.

En aplicaciones prácticas [Proakis 07] y en particular en receptores digitales de señales [Wepman 95], la toma de muestras a intervalos de tiempo regulares (muestreo uniforme) es el que cuenta con mayor aceptación.

Además, es importante destacar que independientemente del tiempo que transcurra entre la toma de muestras consecutivas, muestrear uniformemente una señal, da origen a una secuencia de valores numéricos. No obstante, deben imponerse condiciones a la velocidad de muestreo para garantizar que la información contenida originalmente en una señal pueda ser recuperada.

La figura 2.1B ilustra que el proceso de muestreo es una operación no invertible, ya que una misma secuencia de valores, puede ser producida al muestrear dos señales continuas distintas a una misma frecuencia de muestreo Fm.

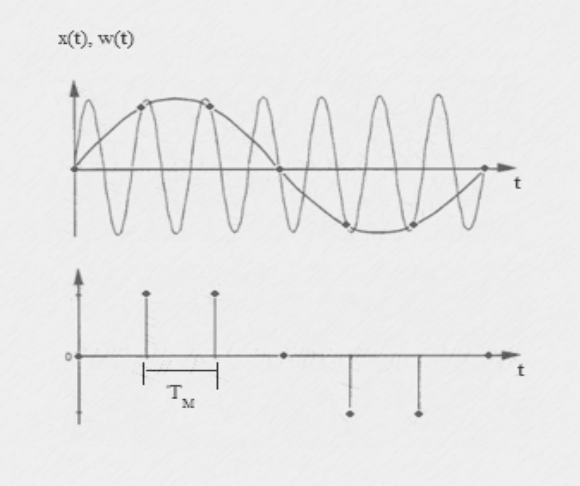

Fig. 2.1B Señales continuas x(t) y w(t) que dan origen a los mismos valores al ser muestreadas a la frecuencia F_M.

En el caso ideal, para obtener esta representación de las señales continuas, se podría hacer uso de un conversor de tiempo continuo a tiempo discreto C/D [Oppenheim 10], el cual permitiría tomar los valores exactos de la amplitud en instantes de tiempo dados. No obstante, los valores de la señal se almacenan en localidades de memoria que brindan precisión finita, por ello, es necesario dividir el rango permitido para la amplitud de la señal analógica en varios niveles, según el número de bits disponibles en dichas localidades, y posteriormente detectar en cuál de los niveles cae la amplitud de cada una de las muestras (proceso de *cuantización*) para finalmente asignarle un valor numérico que corresponda a dicho nivel (*codificación*). Así pues, en la practica, la señal obtenida (señal digital) se caracteriza por ser discreta en el tiempo y discreta en amplitud.

El esquema de conversión analógica digital, así como los procesos involucrados, son presentados en la figura 2.1C. A continuación se presenta una breve discusión de cada proceso y de algunos aspectos de interés que tienen que ver con cada uno de ellos.

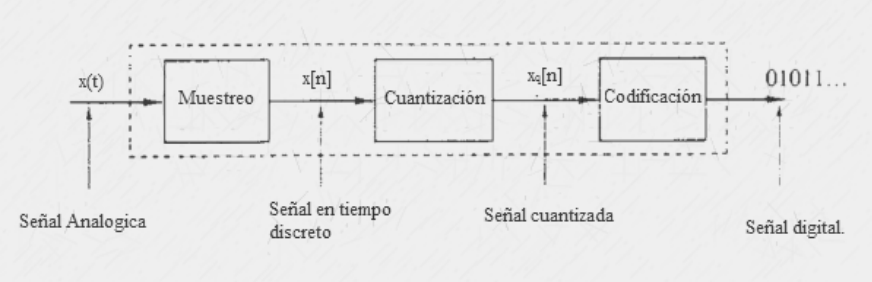

Fig. 2.1C Etapas de la conversión analógica digital.

2.1.3 Muestreo

Como se ha mencionado previamente, en el caso del muestreo uniforme, se pretende obtener una secuencia numerica cuyos valores corresponden a la amplitud que exhibe la señal analógica cada T_M[1] segundos.

Matemáticamente, este proceso es descrito por la relación:

$$x[n] = x_c(nT_m); \quad -\infty < n < \infty \tag{2.1.1}$$

Mismo que puede ser modelado como una multiplicación de señales: la original x(t) y un tren de impulsos s(t), separados T_M segundos.

$$x_M(t) = x(t)s(t) \tag{2.1.2}$$

Donde s(t) está dada como:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_M) \tag{2.1.3}$$

El diagrama a bloques del proceso de muestreo (representado como un interruptor) y el efecto sobre la señal continua se ilustran en la figura 2.1D.

Para comprender mejor este proceso, resulta útil observar lo que ocurre en el dominio de la frecuencia con el espectro de la señal analogica al efectuarlo. Este resultado se consigue tomando en consideración los espectros del tren de impulsos y de la señal original.

[1]El intervalo de tiempo T_M, transcurrido entre cada una de las muestras, se denomina *periodo de muestreo*. Su reciproco $F_M=1/T_M$, conocido como *tasa o frecuencia de muestreo* expresa la cantidad de muestras tomadas, por unidad de tiempo.

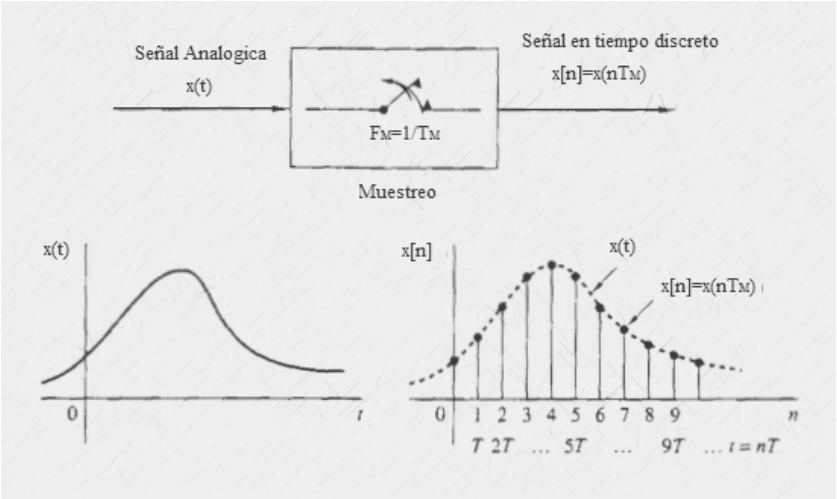

Fig. 2.1D Proceso de muestreo uniforme [Proakis 07].

La multiplicacion de dos señales continuas, produce una señal cuyo espectro, es la convolucion de los espectros de los factores.

$$X_m(F) = X(F) * S(F) \tag{2.1.4}$$

$$X(F) * \left(\frac{1}{T_M} \sum_{n=-\infty}^{\infty} \delta\left(F - \frac{n}{T_M}\right) \right)$$

$$= \frac{1}{T_M} \left(\sum_{n=-\infty}^{\infty} X\left(\frac{n}{T_M}\right) \delta\left(F - \frac{n}{T_M}\right) \right)$$

$$= \frac{1}{T_M} \sum_{n=-\infty}^{\infty} X\left(F - \frac{n}{T_M}\right) = \frac{1}{T_M} \sum_{n=-\infty}^{\infty} X(F - nF_M) \tag{2.1.4a}$$

En la ecuación (2.14a) puede verse que el espectro de la señal muestreada está formado por réplicas del espectro de la señal original centradas en múltiplos enteros de la frecuencia de muestreo (Figura 2.1E). Nótese que este hecho es inherente al proceso de muestreo y no puede ser evitado.

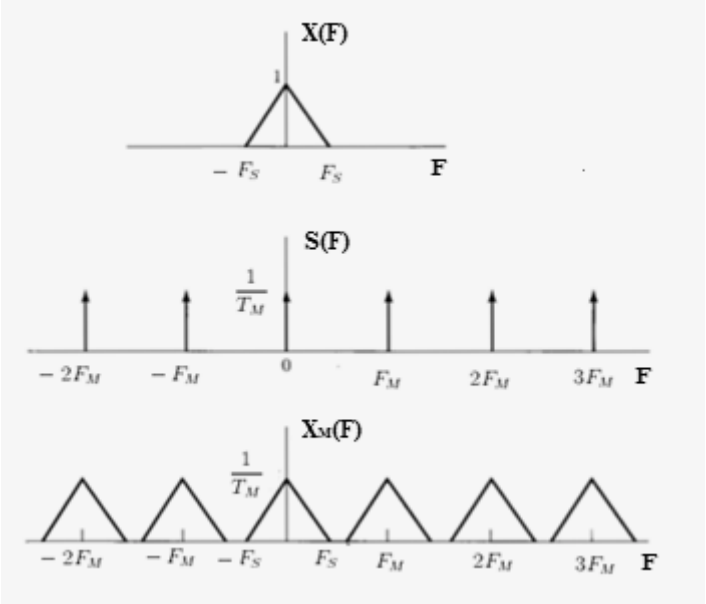

Fig. 2.1E Formación del espectro de una señal muestreada uniformemente [Oppenheim 10].

Como consecuencia de este resultado, el muestreo de una señal analógica requiere que dicha señal esté limitada en banda, lo que significa que su espectro en magnitud no presenta ninguna componente fuera del intervalo [-Fs ,Fs], siendo Fs la frecuencia más grande presente en la señal. En el caso contrario, las réplicas espectrales debidas al proceso de muestreo presentarán solapamiento (Figura 2.1F).Es posible clasificar el tipo de muestreo que se realiza sobre una señal analogica de acuerdo a la velocidad que se emplea con respecto a a la frecuencia máxima presente en la señal, a continuacion se discuten brevemente, algunas alternativas existentes.

2.1.3.1 Muestreo nyquist

El teorema general para el muestreo establece: *una señal puede ser reconstruida exactamente a partir de sus muestras, si estas se obtienen a una tasa de al menos dos veces la componente en frecuencia más grande presente en la señal, Fs.*

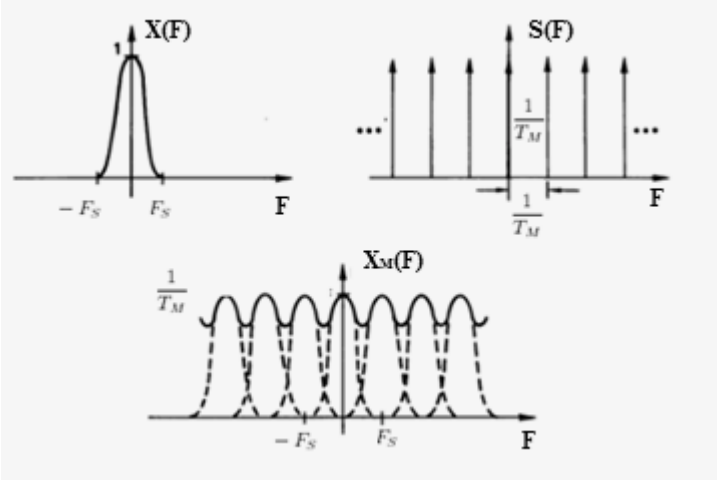

Fig. 2.1F Muestreo a una frecuencia inferior a $2F_S$: se presenta solapamiento espectral.

Esto permite, al menos teóricamente, muestrear una señal a una tasa igual a dos veces la frecuencia máxima de la señal (muestreo Nyquist), sin que las copias espectrales se solapen, véase la figura 2.1G. Si bien una señal limitada en banda pueda ser descrita a través de una expresión matemática, en la realidad las señales comunmente exhiben componentes de una gran cantidad de frecuencias [Wepman 95]. En este caso, un filtro ideal paso banda, colocado antes del proceso de CAD permitiría remover completamente las frecuencias fuera del rango de interés. Al mismo tiempo, conservaría (sin modificar su amplitud) aquellas que resulten útiles lo que permitiria realizar un muestreo Nyquist sin que se presente solapamiento espectral. Véase la figura 2.1H.

Sin embargo, la atenuación que los filtros analogicos ofrecen en la práctica a las frecuencias no deseadas no es infinita, y no se presenta súbitamente, como se plantea en el caso ideal. Lo que ocurre en realidad es un incremento gradual de la atenuación, a partir de la frecuencia de corte hasta la frecuencia de supresión, a lo largo de lo que se denomina banda de transición.

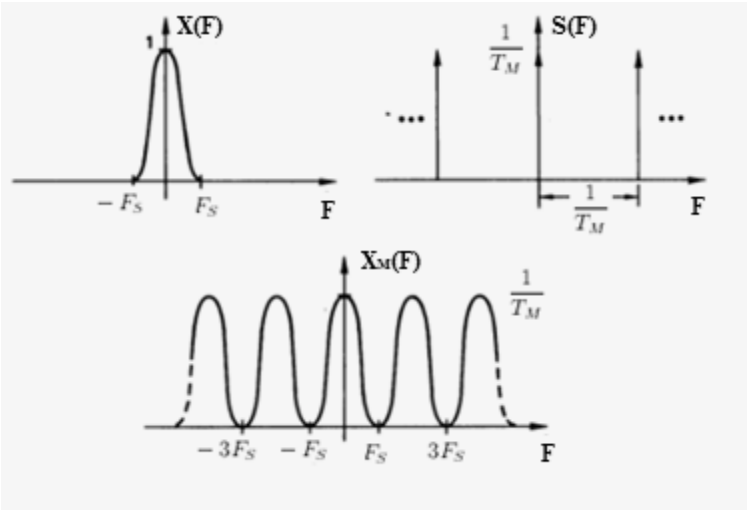

Fig. 2.1G Muestreo Nyquist, no se presenta solapamiento espectral.

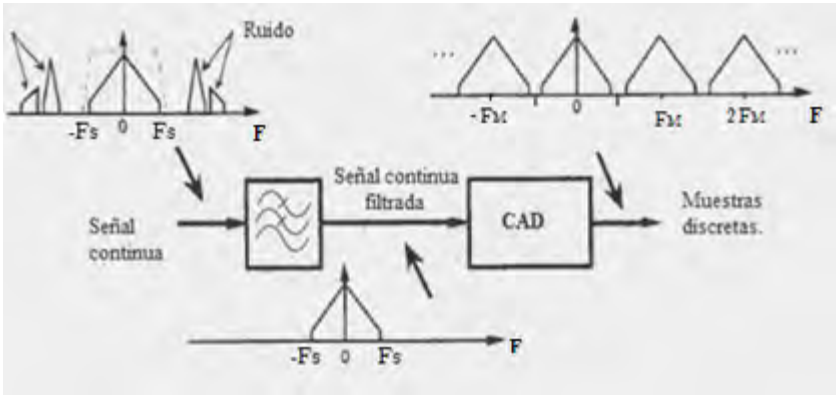

Fig. 2.1H Filtrado previo al proceso de conversión analógica digital [Lyons 10].

Por consiguiente, en el muestreo Nyquist se presenta solapamiento espectral debido a las componentes residuales que generan los filtros reales, a menos que se impongan condiciones muy estrictas sobre el filtro previo a la digitalización.Esto, resulta poco conveniente, ya que a medida que se incrementa la exigencia en un filtro, este resulta más difícil de diseñar e implementar, además de que su respuesta en fase comienza a ser no lineal (i.e. las diferentes componentes de la señal pueden presentar desfasamiento).

Por todo lo anterior, muestrear exactamente a dos veces la frecuencia, en casi todos los casos, no resulta conveniente [Wepman 95].

2.1.3.2 Sobremuestreo

Ante la imposibilidad de realizar muestreo Nyquist, la solucion que se elige con mayor frecuencia es muestrear a una tasa más elevada, lo que se conoce como *sobremuestreo.* Se ha presentado previamente que las réplicas espectrales de la señal analógica se encuentran ubicadas en múltiplos enteros de F_M. Por ello, al incrementar la frecuencia de muestreo, la separación entre replicas aumentará en la misma medida en que F_M lo haga (fig. 2.1I).

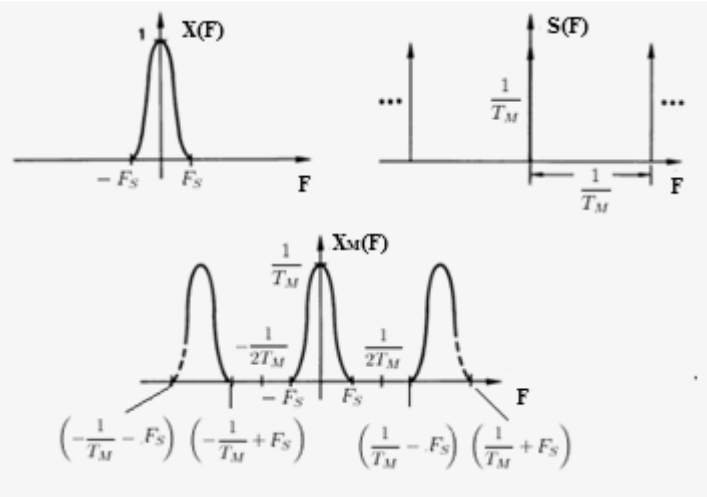

Fig. 2.1I Proceso de sobremuestreo, con una separación de las replicas en el espectro.

Este tipo de muestreo hace posible el uso de un filtro menos complejo y de banda de transición más amplia, sin que haya distorsión por solapamiento espectral. Aunque a priori parecería conveniente para cualquier tipo de señal, este cambio requiere contar con convertidores analógicos digitales más veloces, [Wepman 95], además, en el caso de una cantidad importante de señales de RF, los convertidores analógicos digitales no son capacez de muestrear a las frecuencias requeridas.

2.1.3.3 Muestreo en cuadratura.

Una segunda alternativa , consiste en descomponer a la señal de interés en dos canales de menor ancho de banda para muestrear posteriormente cada uno de ellos. El primer canal se obtiene al multiplicar la señal de interés por una función coseno, lo que se denomina componente en fase (Fig 2.1J). El segundo, conocido como componente en cuadratura, es el resultado de multiplicar la señal original por una función seno (Fig. 2.1K) [Wepman 95][Brigham 88].

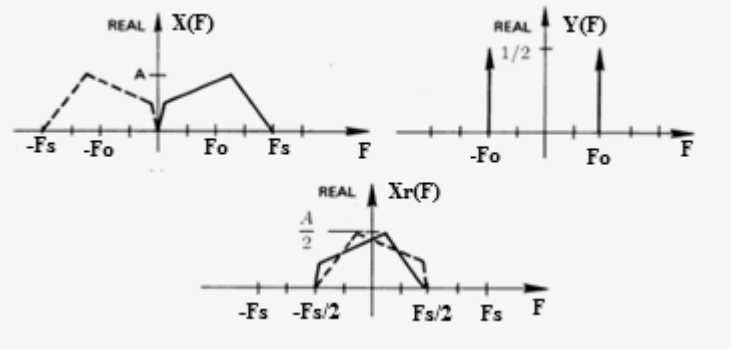

Fig. 2.1J Muestreo cuadratura (componente en fase). [Brigham 88].

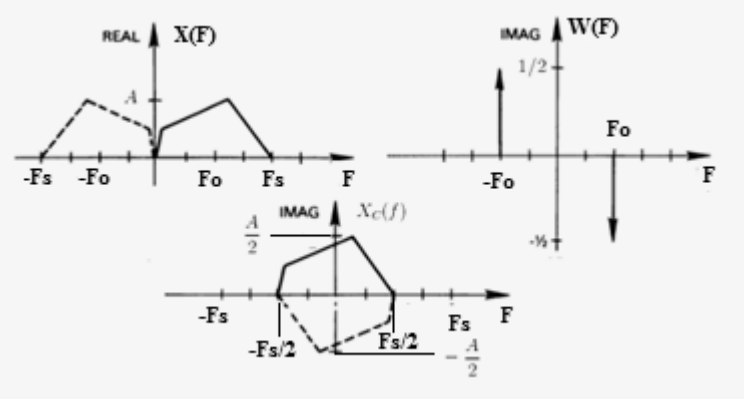

Fig. 2.1K Muestreo cuadratura (componente en cuadratura). [Brigham 88].

Cada una de las componentes ocupa únicamente la mitad del ancho de banda de la señal original, lo que posibilita realizar el muestreo de cada una de ellas a la mitad de la velocidad que se requería originalmente. A pesar de que se presenta solapamiento espectral, los resultados de estas dos operaciones pueden ser combinados para eliminarlo. En [Brigham 88] se presenta a detalle este procedimiento. A cambio, este tipo de muestreo impone el uso de dos CADs que deben estar sincronizados, lo cual puede resultar costoso comparado con el beneficio obtenido.

2.1.3.4 Muestreo paso-banda.

A pesar de que muestrear por debajo de la frecuencia de nyquist pueda producir solapamiento espectral, si la señal de interés es de tipo paso banda[1] es posible violar el teorema de Nyquist muestreando a una velocidad inferior a dos veces la frecuencia maxima de la señal y aún asi lograr una reconstrucción exacta de la señal, siempre y cuando la eleccion de la velocidad siga los siguientes criterios: En primer lugar *, la tasa de muestreo debe ser al menos dos veces el ancho de banda de la señal (i.e. F_2-F_1)* para evitar solapamiento. En segundo lugar, la frecuencia de muestreo debe encontrarse en alguno de los intervalos dados por las ecuaciones (2.1.5) y (2.1.6) [Vaughan 91].

$$\frac{2F_2}{k} \leq F_M \leq \frac{2F_1}{(k-1)} \tag{2.1.5}$$

Donde k es un valor entero que satisface:

$$2 \leq k \leq \frac{F_2}{(F_2-F1)} \tag{2.1.6}$$

Aunque estas condiciones permiten la reconstruccion de la señal, es importante mencionar que algunos intervalos validos para el muestreo generan espectros que se encuentran invertidos, lo cual puede no ser del todo conveniente. Esta condicion se presenta cuando la ecuación (2.1.5) toma valores de k pares, lo que puede pasarse por alto en casos donde el espectro de la señal paso-banda sea simétrico respecto a la frecuencia $(F_2 - F_1)/2$ [Lyons 10].

[1]Son señales en las que no se presentan componentes en frecuencia inferiores a una frecuencia mínima F_1 ni mayores a una máxima F_2.

Este tipo de muestreo es conveniente para señales de RF ó IF de banda angosta, como las que procesa un receptor digital de señales, ya que algunas de las réplicas a las que da origen el proceso de muestreo se encontrarán a frecuencias menores a la original; de entre estas réplicas, debe elegirse la más adecuada para trabajar. Así pues, el muestreo paso-banda permite que la tasa de muestreo sea mucho más baja que la que exige el muestreo uniforme convencional, lo que se traduce en un proceso de CAD que trabaja a una velocidad menor al mismo tiempo que hace posible efectuar muestreo y traslado del espectro en un solo procedimiento.

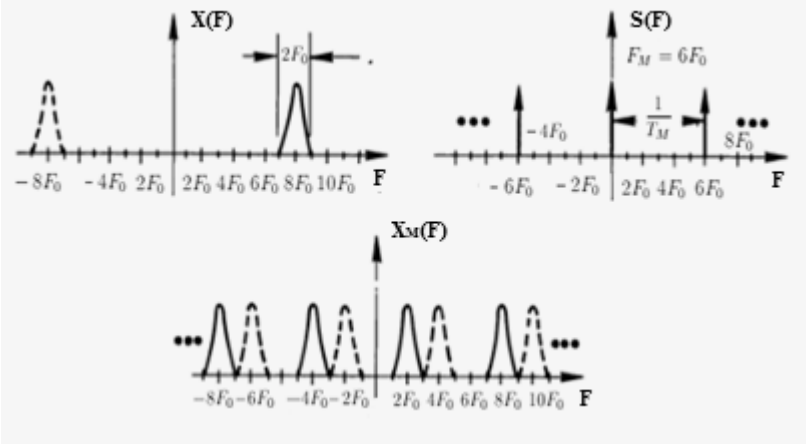

Fig. 2.1L Muestreo paso-banda: permite recuperar la señal y realizar una traslación del espectro a una frecuencia más baja [Brigham 88].

A pesar de la reduccion en velocidad de opreación del CAD, la implementación práctica tiene ciertas limitantes, siendo la más importante el que el convertidor analógico digital debe ser capaz de manejar el ancho de banda completo de la señal. En las hojas de datos de los CAD, esta especificación puede ser encontrada como *analog input bandwith* [Wepman 95].

Además, es importante considerar que la SNR[1] se degrada cuando se realiza muestreo paso banda, puesto que las salidas de los filtros limitadores previos a la digitalización presentan ruido, mismo que se incrementará por un factor k [Lyons 10], tal y como se muestra en la figura 2.1M.

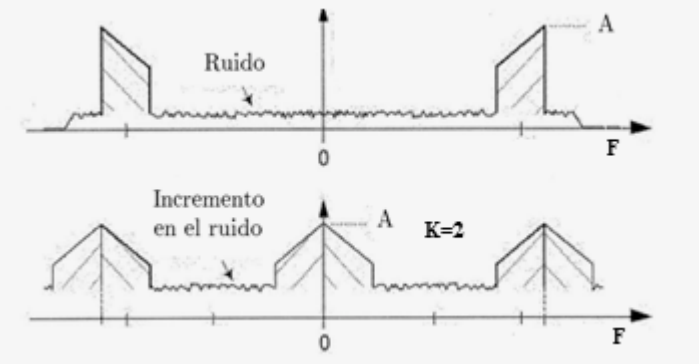

Fig. 2.1M Ruido en el muestreo paso-banda [Lyons 10].

Sin importar el tipo de muestreo que se emplee, el proceso da origen a una señal discreta, con valores de la amplitud de la señal analogica separados un intervalo de tiempo Tm. Para completar el proceso y obtener una señal que pueda ser procesada por hardware digital, es necesario convertir la señal discreta en digital, lo cual se realiza a traves del proceso de cuantización, presentado en la siguiente sección.

[1] La relación señal/ruido (*Signal to Noise Ratio*) se define como 10 veces el logaritmo en base diez del cociente de la potencia de la señal de interés dividida entre la potencia del ruido que presenta. La SNR se mide en decibeles[dB].

2.1.4 Cuantización.

Una señal digital es una secuencia de números que son representados mediante una cantidad finita de bits; es por ello que los valores en amplitud que pueden representarse con ellos ,también, son limitados.

El proceso para obtener una señal digital a partir de un conjunto de muestras es conocido como cuantización. Este procedimiento, consiste en definir diferentes niveles o intervalos validos para el valor numerico que pueden tomar las muestras y asignar un mismo valor a todas las muestras que se encuentren dentro de ese rango, lo cual puede ser conseguido a través de redondeo o truncamiento directo.

Estos rangos reciben el nombre de *niveles de cuantización*, mismos que pueden o no tener el mismo tamaño. Lo mas común, cuando los valores obtenidos son empleados para realizar cálculos numéricos, es emplear rangos de la misma dimensión, teniendo entonces, un proceso conocido como *cuantización uniforme* [Oppenheim 10].

Esta operación es no invertible y no lineal, pero, mediante ella se logra transformar una señal discreta x[n], con un rango continuo de amplitudes, en una secuencia digital, donde cada muestra adquiere uno de entre un conjunto finito de valores posibles. Los CAD únicamente permiten manejar señales con amplitudes en un rango dado, cuyos valores mínimo y máximo que se pueden denotar como V_1 y V_2. En el caso de la cuantización uniforme, este intervalo es dividido en M partes iguales, de valor $(V_2\text{-}V_1)/M$, a las que se representa con la letra griega Δ. Usualmente, la cantidad de niveles de cuantización se presenta en potencias de dos, puesto que la representación en hardware se realiza con números binarios.

Para generar la señal cuantizada, a la que puede denotarse como $x_Q[n]$, se toma cada valor de x[n] y se le asigna el valor del nivel de cuantización que se encuentre más cercano a esa muestra. Por ende, cuando $x_Q[n]$ pasa de un nivel al otro, no se presenta un cambio gradual en su amplitud sino que hace saltos de tamaño igual al del nivel de cuantización (Ver Fig. 2.1N).

Para cada valor de la señal cuantizada, existe una diferencia entre el valor de la señal original y el de la señal cuantizada, al que se denomina error o ruido de cuantización, mismo que tiene siempre una magnitud igual o menor a $\Delta/2$.

Este error es definido como:

$$e[n] = x[n] - x_Q[n] \tag{2.1.7}$$

Pon ende, la señal cuantizada es solo una aproximación de la señal discreta. La calidad de dicha representación depende de la magnitud de los niveles de cuantización. Por lo que puede ser mejorada al reducir el tamaño de los mismos (Figura 2.1O).

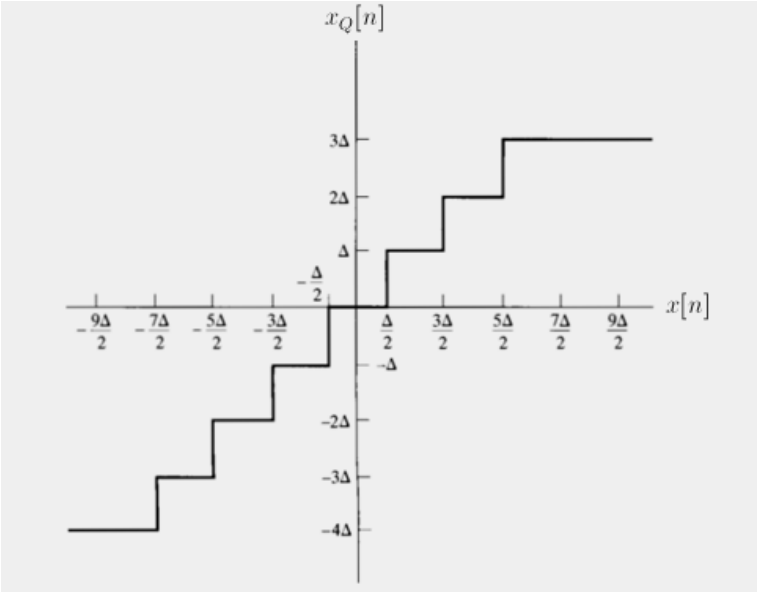

Fig. 2.1N Función de transferencia del proceso de cuantización.

La distorsión introducida por este proceso puede ser cuantificada si se asume que el valor del error está uniformemente distribuido entre todos los niveles de cuantización. En tal caso, la potencia del ruido introducido está dada como [Wepman 95]:

$$P_{qn} = \frac{\Delta^2}{12R} \tag{2.1.8}$$

Donde Δ es el tamaño del nivel de cuantización y R la resistencia de entrada del CAD.

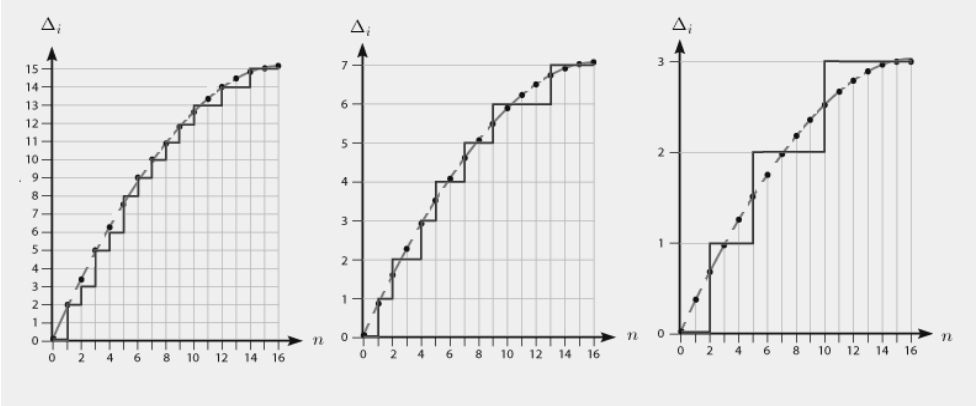

Fig. 2.10 Proceso de cuantización con 4, 3 y 2 bits (de izquierda a derecha).

2.1.5 Consideraciones prácticas.

En receptores digitales, donde la amplitud de la señal deseada cae dentro del *FRS* [1] del CAD, resulta útil cuantificar la SNR. Una buena aproximación puede obtenerse si se asumen las condiciones siguientes:

i) El ruido presente en la señal no tiene otro origen que el proceso de cuantización.
ii) Este error se encuentra distribuido de manera uniforme en todos los pasos de cuantización.

[1]Voltaje máximo permitido para el CAD

En tal caso, una señal senoidal con una amplitud igual al FSR del CAD presenta un SNR de:

$$SNR = 6.02b + 1.76 + 10log_{10}\left(\frac{F_M}{2F_{MAX}}\right)[dB] \tag{2.1.9}$$

Donde b es el número de bits, F_M la frecuencia de muestreo y $F_{MAX,}$ Ela frecuecnia mas grande presente en la señal. En esta expresión puede observarse que el sobremuestreo de la señal (incremento de F_M) mejora la SNR. Sin embargo, el origen del ruido de los CAD no es solo el proceso de cuantización; aparece tambien una degradación adicional causada por el *aperture jiter*[1] el cual aumenta al mismo tiempo que la frecuencia a la que se muestrea. La SNR producida por este fenómeno, está dada como:

$$SNR_{aj} = 20log_{10}\left(\frac{1}{2\pi\, t_a\, F_{max}}\right)[dB] \tag{2.1.10}$$

Donde t_a es el aperture jitter, el cual puede ser encontrado en las hojas de especificaciones de los CAD. Así pues, el SNR de cuantización y del jitter son considerados para obtener el SNR total.

Una vez que se cuenta con una representacion digital es posible hacer uso del hardware de procesamiento para extraer la informacion contenida en la señal o bien, realizar las operaciones requeridas sobre la señal de trabajo. La siguiente sección de este capitulo, se encarga de presentar las tareas que requieren ser realizadas para lograr que un receptor digital logre emular las capacidades de un receptor analogico.

[1] Es la variaciôn en el tiempo exacto del instante de muestreo.

2.2 Receptores digitales de señales

Como se ha presentado previamente, se pretende que un sistema de procesamiento digital logre desempeñar la misma función que un receptor analógico de señales, pero a través de dispositivos digitales. A continuación se presentan algunos esquemas que han sido propuestos para desempeñar esta tarea.

PRIMERA GENERACION:

En [Harris 04] se presenta una de las primeras arquitecturas usadas para intentar incorporar el procesamiento digital de señales en el esquema analógico tradicional.

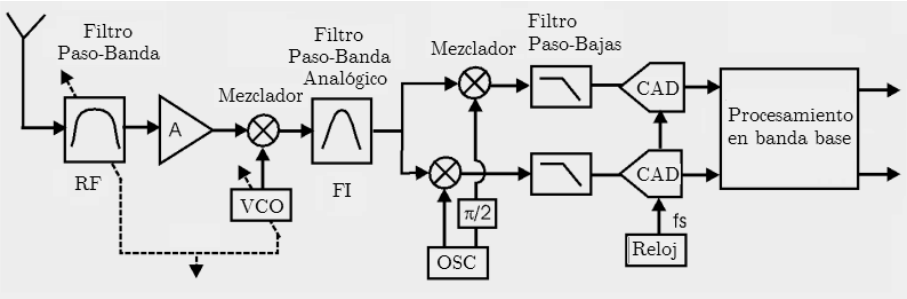

Fig. 2.2A Estructura de un Receptor Dual

El sistema, presentado en la figura 2.2A, comparte los bloques con el receptor analógico que permiten obtener la señal de FI. El hecho de contar con un medio digital al final del proceso permite descomponer la señal analogica en sus componentes en fase (I) y en cuadratura (Q), ya que la reconstrucción de la señal a partir de las muestras de sus componentes, puede ser realizada facilmente en el dominio digital. El muestreo en cuadratura de la señal de frecuencia intermedia permite reducir la tasa de muestreo de los CADs a la mitad.

2.2.1 Demodulación I/Q.

El esquema más simple de demodulación utiliza un multiplicador y un oscilador que genere una señal cosenoidal, como se observa en la (fig. 2.2B). Una multiplicación en el tiempo se traduce en una convolución de los espectros de las señales.

$$x(t)d(t) \leftrightarrow \frac{1}{2\pi}[X(\Omega) * D(\Omega)] \tag{2.2.1}$$

La señal cosenoidal puede expresarse como la suma de dos exponenciales complejas, lo que permite visualizar su espectro como dos impulsos localizados en Ω_c y $-\Omega_c$ (fig 2.2C).

$$\mathrm{d(t)} = \cos(t) = \frac{e^{j\Omega}+e^{-j\Omega}}{2} \tag{2.2.2}$$

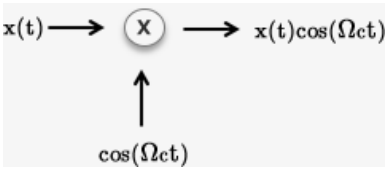

Fig. 2.2B Demodulación coherente.

Por lo tanto:

$$D(\Omega) = 2\pi\left(\frac{\delta(\Omega-\Omega_c)}{2}+\frac{\delta(\Omega-\Omega_c)}{2}\right)$$

$$D(\Omega) = \pi[\delta(\Omega+\Omega_c)+\delta(\Omega-\Omega_c)] \tag{2.2.3}$$

Por otra parte, la convulución espectral con un impulso esta dado como:

$$X(\Omega) * \delta(\Omega + \Omega_c) = \int_{-\infty}^{\infty} X(\Psi)\delta(\Omega + \Omega_c - \Psi)d\Psi$$

La función impuso es cero excepto en $\Omega + \Omega_c$, por lo tanto:

$$= \int_{-\infty}^{\infty} X(\Omega + \Omega_c)\delta(\Omega + \Omega_c - \Psi)d\Psi$$

$$= X(\Omega + \Omega_c)\int_{-\infty}^{\infty} \delta(\Omega + \Omega_c - \Psi)d\Psi$$

La integral de un impulso es igual a 1, asi que:

$$X(\Omega) * \delta(\Omega + \Omega_c) = X(\Omega + \Omega_c)$$

Es decir, la convolución del espectro de una señal con un impulso, centra el espectro en la posición de dicho impulso, por lo tanto, la convolución con una función cosenoidal en el dominio de la frecuencia produce un par de replicas espectrales centradas en Ω_c y $-\Omega_c$.

$$\frac{1}{2\pi}[X(\Omega) * D(\Omega)] = \frac{1}{2}[X(\Omega - \Omega_c) + X(\Omega + \Omega_c)] \qquad (2.2.4)$$

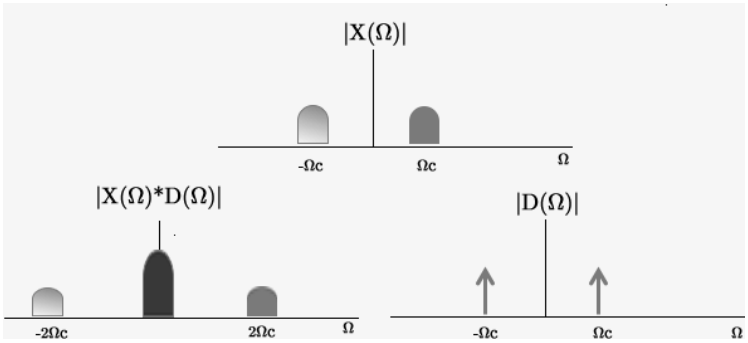

Fig. 2.2C Efectos en frecuencia de la demodulación coherente.

Al elegir una frecuencia Ω_c igual a la frecuencia central de la señal pasobanda, se genera[1] el espectro deseado en banda base junto con dos bandas laterales centradas en $\Omega = \pm 2\Omega_c$, mismas que pueden ser suprimidas por un filtro pasobajas y así conservar unicamente espectro de la señal en banda base (Véase Fig 2.2.C).

No obstante, este proceso solo es posible si el emisor y receptor se encuentran en fase, ya que un defasamiento produce distorsión según la diferencia entre fases. Esto se debe a que una señal coseno con defasamiento es igual a una diferencia de señales seno y coseno cuyas amplitudes dependen también del angulo de defasamiento.(Vease fig. 2.2D).

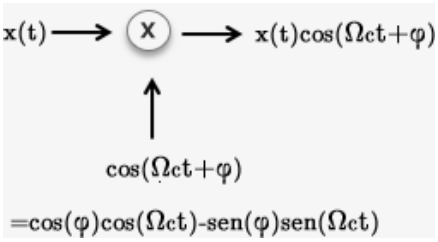

Fig. 2.2D Demodulación con defasamiento entre emisor y receptor

$$e(t) = \cos(\varphi)\cos(\Omega_c t) - sen(\varphi)sen(\Omega_c t) \tag{2.2.5}$$

$$E(\Omega) = \cos(\varphi)\, 2\pi\left(\frac{\delta(\Omega+\Omega_c)}{2} + \frac{\delta(\Omega-\Omega_c)}{2}\right) - sen(\varphi)\left(\frac{\delta(\Omega+\Omega_c)}{2j} - \frac{\delta(\Omega-\Omega_c)}{2j}\right) \tag{2.2.6}$$

$$\frac{1}{2\pi}[X(\Omega) * E(\Omega)]$$

$$= \frac{\cos(\varphi)}{2}[X(\Omega+\Omega_c) + X(\Omega-\Omega_c)] - \frac{sen(\varphi)}{2j}[X(\Omega+\Omega_c) - X(\Omega-\Omega_c)] \tag{2.2.7}$$

[1] Al centrar el espectro en $\pm\Omega_c$ se genera solapamiento espectral en $\Omega = 0$, la parte positiva del espectro centrado en $-\Omega_c$ y la parte negativa del espectro centrado en Ω_c, ambas de amplitud ½, la superposición genera el espectro deseado.

Este defasamiento, puede introducir no solo distorsión sino incluso la eliminación de la componente en banda base, tal y como puede observarse en la ecuación (2.2.7) y en la (fig. 2.2E). Observe, por ejemplo, que la magnitud del espectro en banda base desaparece cuando $\varphi = \frac{\pi}{2}$.

Este hecho resulta completamente indeseable, ya que impide una demodulación adecuada de la señal de interes. Se puede resolver este problema de varias formas, la más directa consiste en generar un mecanismo para ajustar la fase en el receptor hasta que el receptor y el emisor se encuentren en fase, aunque esto puede resultar en un sistema más complejo, dificil de implementar y por ende de mayor costo. Es por ello que en lugar de ajustar la fase del receptor a la del emisor, puede emplearse algún mecanismo que sea inmune a la diferencia entre fases. La demodulación I/Q analógica es un enfoque que presenta esta caracteristica y, al mismo tiempo, permite reducir la velocidad de operación del CAD. La demodulación I/Q, está basada en la propiedad de desplazamiento en frecuencia de la transformada de fourier:

$$x(t)e^{j\Omega_C t} = X(e^{j(\Omega-\Omega_{C)}}) \qquad (2.2.8)$$

La señal f(t)= $e^{j\Omega_C t}$ es compleja, por ende, el producto x(t)$e^{j\Omega_C}$ también lo es, el espectro del producto presenta solo una replica del espectro de x(t) centrada en Ω_C (Fig. 2.2G). En este esquema, un eventual defasamiento sólo produce un cambio por un factor e^{φ} en la amplitud de la señal (Fig, 2.2H) y por ende en el espectro.

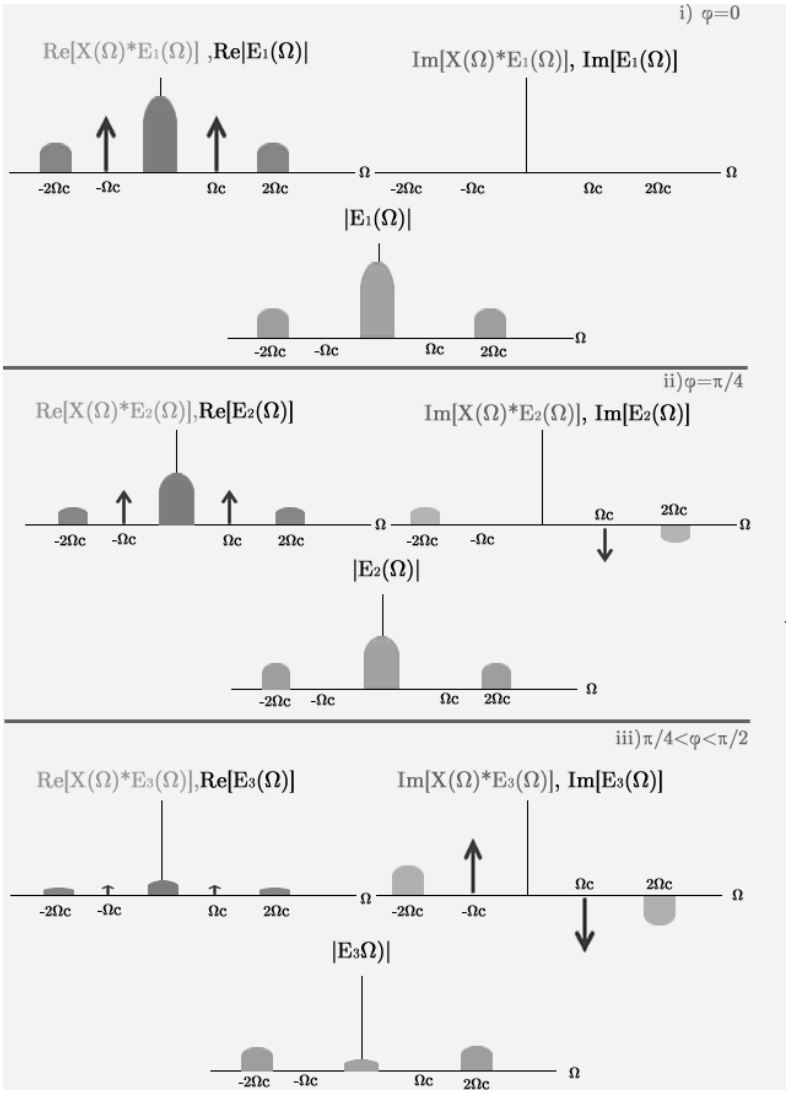

Fig. 2.2E Efectos del defasamiento entre emisor y receptor.

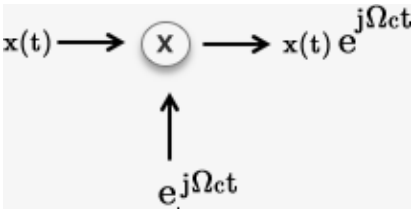

Fig. 2.2F Demodulación I/Q

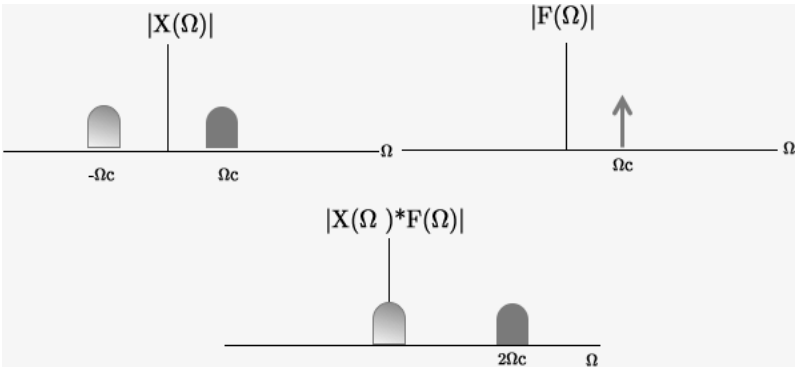

Fig. 2.2G Espectro en demodulación I/Q

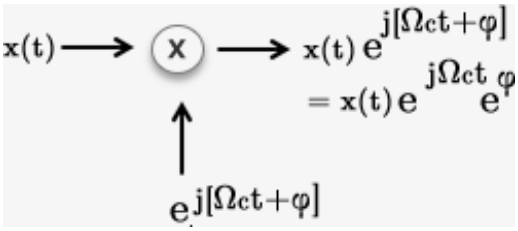

Fig. 2.2H Demodulación I/Q fuera de fase

El inconventiente en este caso, es que debe obtenerse el producto de una señal real y una señal compleja, lo cual resulta imposible de realizar de manera directa en el dominio analogico. Para resolverlo, es posible reexpresar la exponencial compleja haciendo uso de la siguiente identidad:

$$e^{j\Omega t} = \cos(\Omega t) + jsen(\Omega t) \tag{2.2.9}$$

La señal de entrada es una señal real, por lo cual, la parte real del producto $x(t)e^{j\Omega t}$ es igual a x(t)Re[$e^{j\Omega t}$], mientras que la parte imaginaria se obtendrá al multiplicar x(t) y Im[$e^{j\Omega t}$]. De (2.2.9) puede obtenerse que las partes real e imaginaria del producto $x(t)e^{j\Omega t}$, estarán dadas como $x(t)\cos(\Omega t)$ y $x(t)\,\mathrm{sen}(\Omega t)$, respectivamente. A diferencia de una exponencial compleja, las señales seno y coseno si pueden ser generadas directamente, lo que permite que las componentes real (I, componente en fase) e imaginaria (Q, componente en cuadratura) de la señal de interés puedan ser calculadas por separado, en canales diferentes. Notese que la señal resultante debe ser necesariamente compleja, ya que todas las señales reales poseen espectros en magnitud simetricos, condición que no cumple el espectro planteado previamente para este esquema de demodulación.

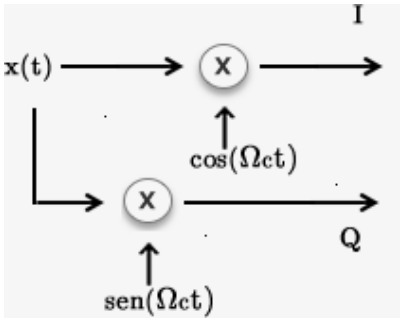

Fig. 2.2l Demodulación I/Q fuera de fase

2.2.2 Problemas de la demodulación I/Q analogica.

Si este proceso se realiza en el dominio analogico, se requerirá de dos mezcladores que se encargan de obtener la representación de la señal en sus componentes en fase y en cuadratura. En este punto, cada canal es filtrado y posteriormente digitalizado de manera independiente para ser procesado. Aunque la tasa de muestreo requerida para el proceso disminuye, exige contar con dos convertidores y dos mezcladores analógicos: estos últimos, sujetos a las limitantes y variaciones de los componentes analogicos. Se pueden mencionar dos de los problemas más importantes [Harris 04]:

i) La fase y la ganancia de las señales en cuadratura suelen exhibir un desbalance debido las diferencias entre los componentes de los filtros analógicos empleados.

ii) Los filtros analógicos usualmente requieren una banda de transición angosta, que en la mayoría de los casos conduce a distorsión por retardo de grupo[1].

El avance actual del PDS permite que estos problemas puedan ser resueltos a través de algoritmos adaptativos que eliminan gran parte de las imprecisiones y distorsión, una vez que la señal se encuentra digitalizada. No obstante, el objetivo de usar dispositivos digitales no es corregir las imprecisiones generadas por componentes analógicos sino evitarlos, por lo que este esquema dista de ser una solución óptima.

[1] Este tipo de distorsión y los efectos que tiene sobre las señales, se presentan en la tercera sección de este capítulo.

SEGUNDA GENERACIÓN.

El avance en el desarrollo de CADs dio origen a la segunda generación de receptores digitales, ya que hicieron posible la conversión analógica-digital directamente en FI y no en banda base como ocurria con los receptores de primera generación. Nótese que aún es necesario contar con componentes analógicos para trasladar la señal de interés de RF hasta FI.

Una vez que se cuenta con las muestras de la señal, se realizan las operaciones necesarias para extraer la información de interés en el dominio digital. En este esquema, el CAD debe operar a una frecuencia considerable pero ofrece eliminar los desbalances presentes en receptores de la primera generación.

La tarea de procesamiento se realiza a través de hardware digital (Figura 2.2J). Como se ha presentado, es deseable el uso de un sistema de demodulación I/Q para poder lidiar con un eventual desfasamiento entre emisor y receptor. En este caso se trabaja con señales digitales, las cuales exhiben espectros periodicos, por lo que la demodulación I/Q no genera resultados identicos a los que se obtienen en la demodulación analógica.

No obstante, la dualidad entre multiplicación en el dominio temporal y convolución en el dominio espectral se mantiene, con la unica diferencia de que la convolución se realizará entre dos funciones periodicas; asimismo, la propiedad de desplazamiento en la frecuencia, tambien es valida para señales discretas. Por lo anterior, la demodulación I/Q puede efectuarse sin problemas en hardware digital, una vez que se ha llevado a cabo el proceso de CAD, lo que permite eliminar las variaciones e impresiciones que exhiben los demoduladores I/Q, cuando son implementados con componentes analogicos.

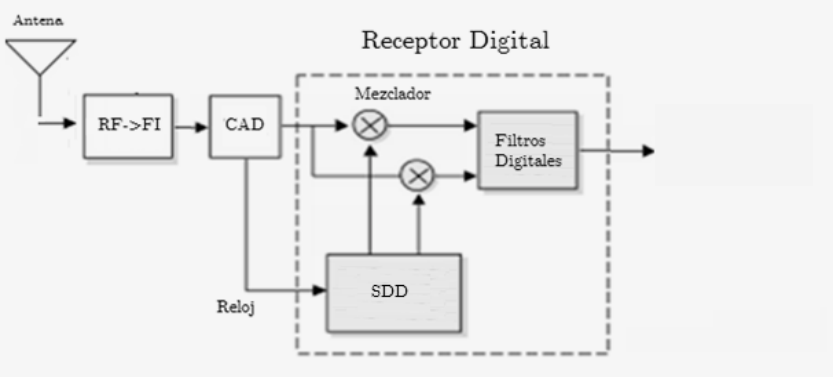

Fig. 2.2J Receptor digital y componentes.

Los componentes digitales que permiten implementar el demodulador I/Q en el dominio digital son:

i) Un sintetizador digital directo (SDD).
ii) Dos multiplicadores digitales signados.

El SDD se ocupa de generar muestras de las dos señales senoidales, operando con el reloj del CAD, para efectos de sincronización; mientras que los multiplicadores obtienen el producto de la señal a procesar con las señales generadas localmente. Posteriormente, un proceso de filtrado elimina el ruido y las componentes no deseadas en el espectro de ambas señales, al mismo tiempo, se reduce la tasa de muestreo (se eliminan muestras) para trabajar con la menor cantidad de datos posible, permitiendo mejorar el rendimiento y reducir la demanda en almacenamiento, en caso de ser necesario.

.

2.2.3 Sintetizador digital directo.

Se trata, como se ha descrito previamente, de un medio para generar señales digitalmente, con la ventaja de que permiten contar con frecuencia configurable. Provee ventajas de costo, alto desempeño y tamaño reducido; presentándose como una alternativa viable a los generadores de señales analógicos, ya que:

1) Exhibe mayor precisión y control en su operación con respecto a los generadores analogicos.
2) Elimina componentes de ajuste manual, y ofrece configuracion por medio de software.
3) No presenta variaciones por efectos de temperatura o envejecimiento.
4) Permite controlarse u optimizarse de manera remota.
5) Ofrece cambios inmediatos al variar la frecuencia, sin presentar los sobrepasos o efectos de asentamiento que se tienen en generadores analógicos.

El SDD más simple que puede ser implementado requiere:

iii) un reloj
ii) un contador binario.
iii) una memoria

La memoria se encarga de almacenar valores de la amplitud de un periodo de la señal y opera como una tabla de consulta (LUT) que se lee en forma repetida. El contador genera los valores necesarios para acceder a cada una de las localidades de la memoria y así generar la señal deseada.

Esta solución presenta un inconveniente importante: solamente es posible cambiar la frecuencia de la señal haciendo variar el reloj de referencia o modificando las muestras almacenadas en la memoria, lo que resulta impráctico y brinda poca flexibilidad.

Como alternativa, se añade usualmente un acumulador de fase para realizar el control o ajuste de la frecuencia generada a través de bits almacenados en localidades de memoria, lo que brinda mayor precisión (Fig. 2.2K).

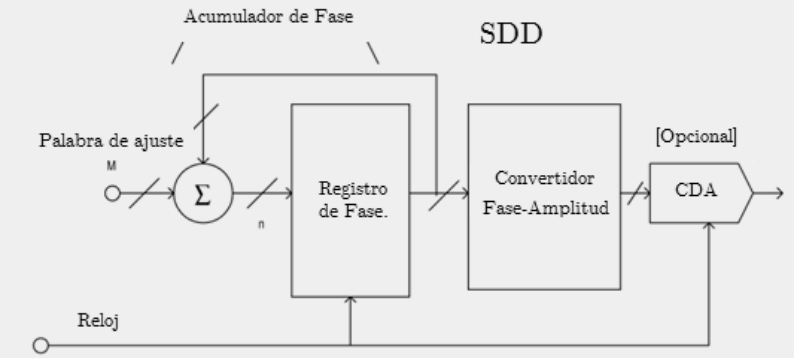

Fig. 2.2K Componentes de un sintetizador digital directo

La acumulación de fase se realiza a través de un contador de N bits y un registro. Para comprender mejor el proceso es útil visualizarlo como una "rueda de fase".

Cada punto en la rueda corresponde a una muestra de la señal senoidal y recorrer completamente la rueda, a velocidad constante, es equivalente a generar las muestras correspondientes a un periodo de la señal (Fig. 2.2L).

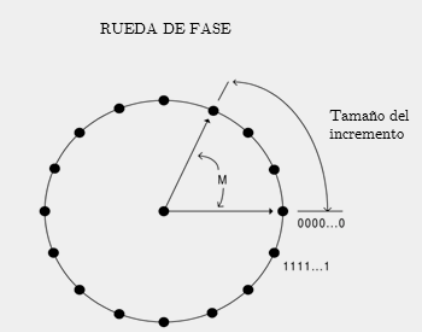

Fig. 2.2L Rueda de fase.

Tal y como puede verse en la figura 2.2K, Se hace uso del acumulador o registro de fase para almacenar la cantidad de pasos que han sido recorridos en la rueda. Este acumulador, se implementa como un contador que incrementa su valor cada vez que recibe un flanco de reloj.

El incremento que realiza el contador en cada ciclo está controlado por otro registro en el que se almacena la cantidad de pasos que se avanza en la rueda de fase en cada ciclo (Palabra de ajuste en la fig. 2.2K). Al modificar el valor en ese registro, aumenta o disminuye la velocidad con la que se recorre la rueda, consiguiendo con ello modificar la frecuencia de la señal generada. La salida obtenida del acumulador de fase no puede ser usada directamente para generar una onda senoidal, se requiere una mapeo entre la fase actual en el acumulador y la amplitud correspondiente (Convertidor Fase-Amplitud, fig 2.2K). Esto se realiza a través de un proceso de truncamiento de la fase actual al valor más cercano y una lectura del valor correspondiente para esa muestra en la memoria.

Gran parte de estos dispositivos aprovechan la naturaleza simétrica de las ondas senoidales y son capaces de sintetizar una onda senoidal completa a partir de 1/4 de ciclo.

2.2.4 Filtrado y decimado.

Al realizar el filtrado en forma digital se puede lograr fácilmente fase lineal, lo que elimina completamente la distorsión por retardo de grupo, incluso si el orden del filtro es elevado. Filtrar requiere realizar sumas, restas, acumulaciones de valores anteriores y/o multiplicaciones según el filtro que se emplee, siendo las multiplicaciones, las operaciones que más tiempo requieren para ser realizadas. Un RDS es un dispositivo de alto desempeño, por lo que resulta conveniente reducir al mínimo la cantidad de operaciones, y en particular multiplicaciones, que se realicen durante el proceso y así elevar la capacidad de procesamiento.

En busca de dicho objetivo, todas las estructuras y tipos de filtros que seran presentados posteriormente pueden ser útiles, cada uno de ellos ofreciendo ventajas y algunas limitantes, entre ellas destacan aquellas que permiten la eliminación de multiplicaciones, y constituyen una elección común entre los diseñadores de sistemas de PDS de alto desempeño.

En RDS es común encontrar filtros CIC como primera etapa de filtrado debido a que evitan uso de multiplicadores, reducen la tasa de muestreo de la señal y al mismo tiempo, la demanda sobre los componentes de procesamiento.

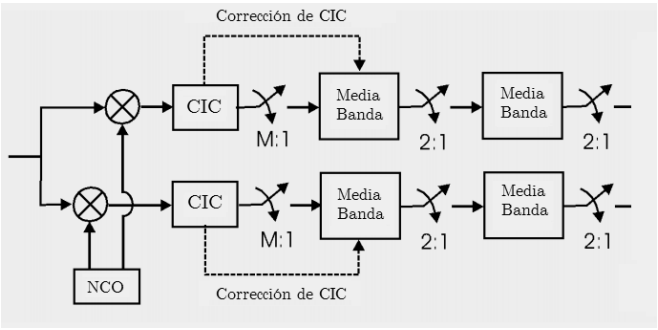

Fig. 2.2M Filtrado CIC+ Media banda.

En [Harris 04] se propone una arquitectura para RDS a base de filtros CIC decimadores y filtros de media banda tal como se muestra en la Fig. 2.2M.

La primera sección de filtrado CIC permite reducir la tasa de muestreo por un factor grande (entero). La irregularidad que ofrecen los filtros CIC, en sus bandas de paso y supresión, motiva que la segunda etapa de filtrado se realice con filtros de media banda y se diseñe con una respuesta en magnitud complementaria, lo que permite obtener una repuesta más plana en la banda de paso. Finalmente, se plantea una segunda etapa de filtros de media banda, con el objetivo de ajustar la ganancia, concluir el trabajo de filtrado y tener una mejor respuesta en frecuencia.

2.2.4.1 Receptor digital de señales y submuestreo.

Como se mencionó previamente, en este esquema aún es necesario un bloque analogico de traslación de RF a FI. La frecuencia de muestreo es usualmente de $Fm=2F_{MAX}+\Delta f$. Con Δf igual a la banda de transición del filtro empleado para limitar espectralmente la señal.

Las señales que motivan este trabajo son típicas señales paso-banda, por lo que cumplen con las condiciones para hacer uso de muestreo paso-banda y así reducir la demanda sobre el CAD. Dicha reducción abre las puertas para emplear los CADs disponibles actualmente (vease tabla 2.T1) para muestrear directamente

señales de RF y así implementar el receptor haciendo uso únicamente de la antena, el amplificador sintonizado y los mismos bloques que emplean los receptores digitales de frecuencia intermedia (Fig. 2.2N).

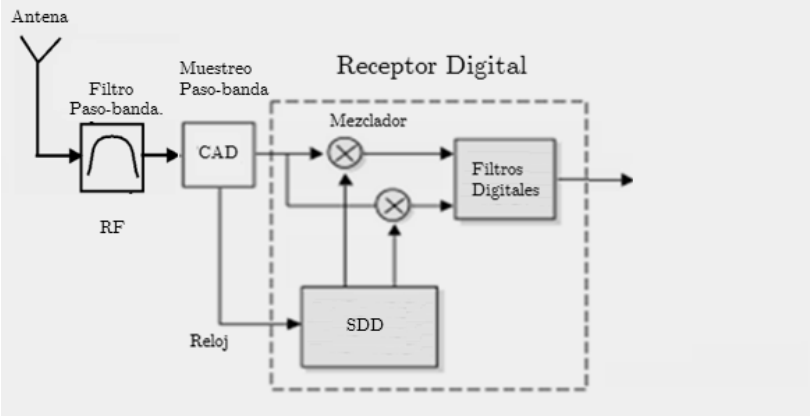

Fig. 2.2N Receptor digital para submuestreo pasobanda.

Modelo	bits	*Full Analog Bandwitdh*	F_M maxima	Costo/unidad
		Analog Devices		
AD9484	8	1[Ghz]	500[Mm/s]	36[USD]
AD9286	8	500[Mhz]	500[Mm/s]	36[USD]
AD9432	12	500[Mhz]	500[Mm/s]	
		National Semiconductors		
ADC12D1800	12	2.8[Ghz] 1.76[Ghz]	1.8[Gm/s] 3.6[Gm/s]	--
ADC12D1600	12	1.75[Ghz] 2.8[Ghz]	2/3.2[Gm/s] 1/1.6[Gm/s]	---

Tabla 2.T1 Algunas caracteristicas de convertidores analogicos disponibles en 2011.

Es muy importante destacar dos aspectos de este enfoque:

El primero de ellos es que el convertidor analogico que se elija, aun cuando no necesita ser capaz de tomar muestras a dos veces la frecuencia maxima de la señal, si debe ser capaz de detectar todas las variaciones que ocurren en ella para poder capturar de manera precisa el valor de la señal al tomar la muestra. Como se ha presentado en el capitulo 1, el *parametro full analog bandwith*, expresa la frecuencia maxima en donde el CAD ofrece una respuesta aceptable (3[dB] de atenuación). Por lo cual, la frecuencia maxima presente en la señal de interes debe ser menor a este valor. Como puede verse en la tabla 2.T1, hasta 2011, los convertidores permitían digitalizar señales con muestreo paso-banda siempre y cuando su frecuencia maxima fuese menor a 2.8[Ghz].

El segundo aspecto tiene que ver con la elección del ancho de banda, aunque *a priori* puede pensarse que es una caracteristica favorable en todos los casos. Entre mayor sea el ancho de banda del convertidor más ruido podra ingresar al mismo, con lo que el trabajo de filtrado puede complicarse significativamente.

Una vez presentado el sistema a realizar, se ofrece en la siguiente sección una presentación de la teoria basica de procesamiento digital de señales de tasa multiple, del diseño de filtros y de algunas estructuras y familias de filtros que se encuentran en la literatura que usualmente son consideradas en el desarrollo de sistemas de procesamiento de señales de alto desempeño.

2.3 Procesamiento de señales de tasa múltiple.

Para describir matemáticamente el comportamiento de un sistema de procesamiento digital de señales, se hace uso de un operador o mapeo T{.}, al que se denomina *sistema en tiempo discreto,* para expresar la forma en que puede obtenerse la señal de salida y[n] a partir de una señal de entrada x[n]; es decir, y[n]=T{x[n]}. Es común trabajar con los sistemas discretos que sean *lineales e invariantes en el tiempo* (SLIT)[1], debido a la facilidad que presentan para ser analizados y a que pueden ser implementados facilmente. En ellos se cumple que las señales x[n] e y[n] comparten una misma frecuencia de muestro F_M (Fig 2.3A).

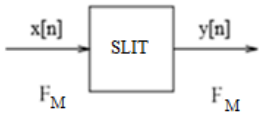

Fig. 2.3A Sistema Lineal Invariante en tiempo SLIT

En algunos casos, entre los que se encuentran los receptores digitales de señales, se requiere realizar cambios en la tasa de muestro. Los sistemas que lo permiten se denominan de *tasa múltiple* (STM), los cuales se presentan a continuación

---

[1]Son sistemas discretos que interactúan con su señal de entrada, x[n], a través de un proceso conocido como convolución, para obtener una salida y[n]. Esta última es denotada y definida respectivamente como:

$$y[n] = x[n] * h[n] \qquad y[n] = \sum_k x[k]h[n-k]$$

Siendo h[n] (*respuesta impulso*) la salida del sistema al tener un impulso unitario δ[n] a la entrada, donde:

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & otro\ caso \end{cases}$$

La posibilidad de cambiar la tasa de muestreo de una señal discreta resulta valida si se considera lo siguiente: Una señal continua $x_C(t)$ muestreada uniformemente a frecuencias distintas F_x y F_y donde ambas cumplen el *teorema de muestreo* , da origen a dos señales discretas x[n] y y[n] a partir de las cuales puede obtenerse una reconstruccion de $x_c(t)$ sin perdida de información. Por tanto y[n] y x[n] pueden contener la misma información a pesar de presentar una tasa de muestreo diferente.

La conversión de la tasa de muestreo (CTM) puede ser consecuencia natural de la aplicación, o bien, es impuesta para acceder a las ventajas asociadas con este enfoque.

2.3.1 Motivación.

Cambiar la tasa de muestreo de una señal resulta de interés porque permite:

- Realizar una tarea de procesamiento con desempeño mejorado y a un costo considerablemente más bajo en ciertas aplicaciones [Harris 04].

- Emplear algoritmos computacionalmente eficientes si la señal de interés es procesada a diferentes tasas en un mismo dispositivo [Crochiere 81].

- Interconectar dos o más dispositivos de PDS que operen a tasas diferentes, permitiendo con ello el intercambio de datos entre dispositivos.

En este trabajo, el uso de la conversión de la tasa de muestreo es conveniente, ya que permite reducir la demanda sobre los filtros, permitiendoles realizar una menor cantidad de opreaciones por unidad de tiempo, además permite reducir la cantidad de datos a procesar sin tener perdidas de información, lo que se traduce en una reducción en la cantidad de muestras a procesar.

La CTM encuentra algunas otras de sus aplicacicones en: sistemas de *PDS* en tiempo real, sistemas de radar, procesamiento de voz, procesamiento de imágenes, procesamiento de imágenes médicas y satelitales etc.

2.3.2 Descripción.

Existen dos formas de transformar x[n] en y[n], siendo ambas señales discretas y de frecuencias de muestreo distintas:

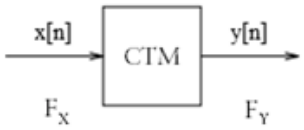

Fig. 2.3B Convertidor de tasa de muestreo.

1) *Analógico y digital:* El cual consiste en emplear un CDA para generar una señal continua x(t), a partir de una señal x[n] que presente una tasa de muestreo Fx, posteriormente se puede realizar un muestreo uniforme de x(t), a una tasa F_Y, y así producir a y[n]. Este esquema resulta impráctico a causa de la distorsión introducida por el CDA y el ruido de cuantización involucrado en el proceso de CAD.

2) *Digital:* Éste enfoque busca modificar únicamente los valores de una señal sin abandonar el dominio digital, a través de la incorporación o eliminación de muestras. Cualquiera de estos dos procesos van acompañados por una o más etapas de filtrado. Este esquema resulta más conveniente, ya que se puede realizarse completamente en el dominio digital, y en la práctica es el que cuenta con mayor aceptación.

La CTM digital se efectua de manera distinta si se desea aumentar (Interpolación) o disminuir (Decimacion) la frecuencia o tasa de muestreo de la señal de interés. Enseguida se realiza una descripción de ambos procesos.

2.3.2.1 Decimación:

Como se ha mencionado, este proceso consiste en la reducción de la tasa o frecuencia de muestro de una señal discreta x[n]. Por simplicidad, conviene analizar este proceso cuando la tasa es disminuida M^1 veces, donde M es un número entero positivo. Para realizar esta tarea, se necesitan dos etapas, la primera es una etapa de filtrado y la segunda es de compresión de la tasa de muestreo.

El filtrado, considerado como la disciplina más antigua en el campo del PDS, consiste en la supresión o atenuación de componentes en frecuencia del espectro de una señal. Para entender su importancia, como parte del proceso, conviene recordar que: el espectro en magnitud de las señales discretas esta formado por réplicas del espectro de la señal continua que dio origen a ellas, centradas en múltiplos enteros de la frecuencia de muestreo(Ecuación 2.1.4a). Es por ello que al reducir la frecuencia de muestreo (a través de decimación), la separación entre las réplicas tambien disminuye y, como consecuencia, si el factor de compresión produce una tasa F_Q demasiado pequeña, las replicas espectrales presentarán solapamiento. Ante este escenario, el filtrado permite reducir el tamaño de las replicas (eliminando parte de la información, si esta no resulta de interes), y así conseguir una reducción en la tasa de muestreo por factores más grandes.

[1]Se denomína *factor de decimación.*

Por su parte, la compresión de la tasa de muestreo radica en la eliminación de algunas de las muestras que componen la señal. Comprimir la tasa de una señal discreta s[n] por un factor M, proceso comunmente denotado como $M\downarrow$, se traduce en conservar uno de sus valores por cada M y eliminar los M-1 valores intermedios. (i.e. Conservar s[0], s[M], s[2M], etc. descartando todos los demás valores de la señal). La figura 2.3C presenta graficamente este proceso.

Al Comprimir s[n] se obtiene la señal q[n], lo cual se puede representar matematicamente mediante:

$$q[n] = s[Mn] \tag{2.3.1}$$

Siendo la relación entre las tasas de muestreo la siguiente:

$$F_Q = \frac{F_S}{M} \tag{2.3.2}$$

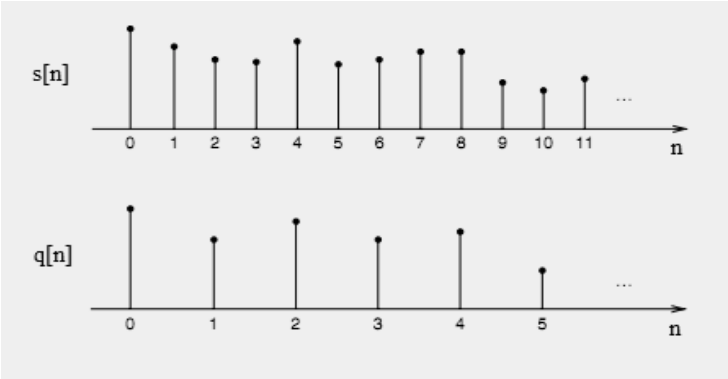

Fig. 2.3C Compresión de la tasa de muestreo ; M=2.

Una forma de visualizar lo que ocurre en el espectro de una señal s[n] al ser decimada por un factor M, es considerar que s[n] es resultado de muestrear una señal continua s(t) a una tasa F_M. Despues de reducir la tasa de muestreo por un factor M se obtiene una señal q[n] con una tasa de muesreo F_M/M. Por otro lado, si se muestrea a s(t) a una tasa F_M/M, se produce una señal discreta identica a q[n]. Por ende, para visualizar el espectro de la señal discreta con una tasa de muestreo F_M/M, basta con analizar el espectro de s(t) muestreada a esa frecuencia.

En la figura 2.3D puede observarse el resultado del proceso de decimado de una señal, variando sucesivamente el factor M. Si éste es lo suficientemente grande (M=4), existe distorsión del espectro original, generada por el solapamiento de replicas (Ver fig. 2.3D(c) y (d)).

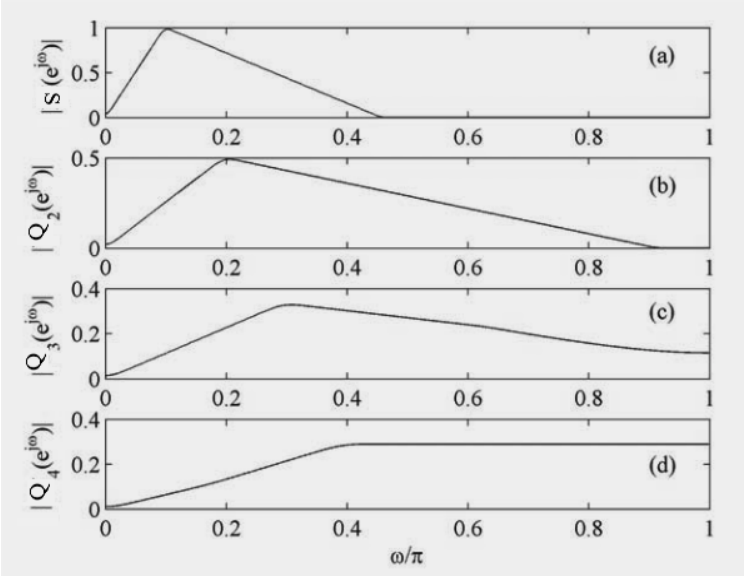

Fig. 2.3D (a) Espectro de q[n] , (b) Espectro de q[n] con M=2, (c) M=3, (d) M=4 [Milic 09].

En general, el solapamiento espectral puede prevenirse sí se garantiza que la tasa de muestreo F_Q, obtenida después de la compresión, es mayor al producto del factor de decimado y del ancho de banda de la señal, esto es:

$$F_Q > MB \tag{2.3.3}$$

Cuando la aplicación requiere una tasa menor a MB, es posible filtrar la señal de interés para limitar el ancho de banda y con ello obtener la tasa de muestreo deseada sin la aparición de solapamiento espectral. A manera de resumen, el proceso completo de decimado se presenta en la figura 2.3E.

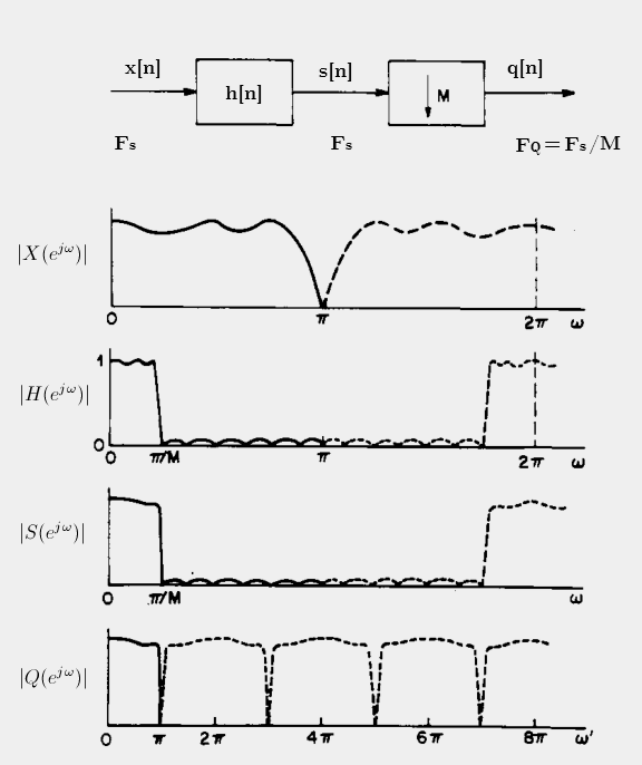

Fig 2.3E Proceso completo de decimado [Crochiere81]: x[n] señal original , s[n] filtrada, q[n] comprimida. h[n] filtro paso bajas.

2.3.2.2 Interpolación:

Es el proceso que consiste en incrementar la tasa de muestro de una señal x[n]. Por simplicidad, se analiza el caso donde la tasa se incrementa L veces, siendo L un número entero positivo. Para llevar a cabo este procedimiento sobre una señal discreta son necesarias dos etapas, la primera realiza una expansión de la tasa de muestreo y la segunda, efectua un proceso de filtrado.

La expansión de la tasa de muestreo de una señal discreta s[n] por un factor L, proceso denotado como ↑L, consiste en insertar (L-1) muestras iguales a cero entre cada una de las muestras de s[n] (i.e. añadir (L-1) muestras nulas entre s[0] y s[1], (L-1) entre s[1] y s[2] etc.)(Fig.2.3F)

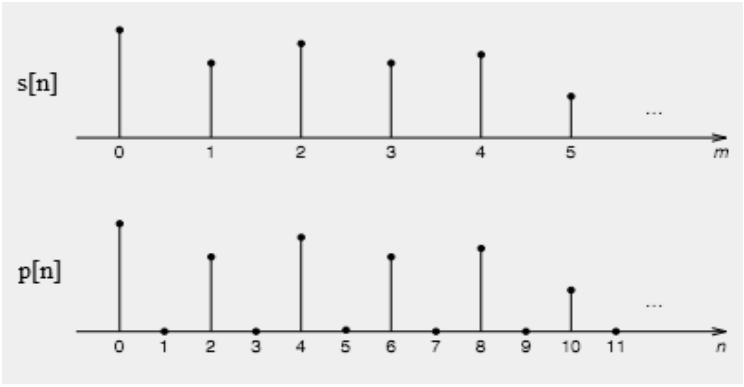

Fig. 2.3F Proceso de expansión de la tasa de muestreo, L=2.

De este proceso se obtiene la señal:

$$p[n] = \begin{cases} s[n/L]. & n = 0, \pm L, \pm 2L... \; etc. \\ 0 & otro \; caso \end{cases} \tag{2.3.4}$$

Esto es:

$$\begin{aligned} &p[0] = s[0]; \\ &p[1] = p[2] = \ldots = p[L-1] = 0; \\ &p[L] = s[1]; \\ &p[L+1] = p[L+2] = \ldots = p[2L-1] = 0; \\ &p[2L] = s[2]. \end{aligned}$$

Para comprender la relación entre las señales p[n] y s[n] en el dominio de la frecuencia, conviene analizar su transformada Z.

Esto es:

$$p(Z) = \sum_{n=-\infty}^{\infty} p[n]\mathbf{Z}^{-n} \tag{2.3.5}$$

$$= \sum_{n=-\infty}^{\infty} s[n/L]\mathbf{Z}^{-n} \tag{2.3.6}$$

$$con\ n = 0, \pm L, \pm 2L \ldots etc$$

$$si\ m = n/L;\ \ n = mL.$$

$$p(Z) = \sum_{m=-\infty}^{\infty} s[m]\mathbf{Z}^{-mL} = S(Z^L) \tag{2.3.7}$$

Con este resultado, es posible obtener el espectro de magnitud de la señal resultante evaluando la transformada Z en el círculo unitario, es decir:

$$Z = e^{j\omega}$$

$$S(Z^L)_{z=e^{j\omega}} = S(e^{j\omega L}) = P(e^{j\omega}) \tag{2.3.8}$$

De la ecuación (2.3.8) se observa que el espectro de la señal cuya tasa de muestreo ha sido expandida $P(e^{jw})$ es una versión de $S(e^{jw})$ con un escalamiento en frecuencia, ya que ω es reemplazada por ωL. Conocida la naturaleza periódica del espectro de una señal discreta, el resultado de este escalamiento es la inclusión de (L-1) réplicas del espectro original que previamente se encontraban fuera del periodo fundamental.

El trabajo de la segunda etapa, el proceso de filtrado, es (idealmente) eliminar completamente las replicas que aparecen en el proceso de expansión, con lo cual se logra la interpolación exacta de la señal. No obstante, ésa operación no puede ser completamente realizada, debido a la demanda de alta eficiencia computacional impuesta por la CTM [Lyons 10].

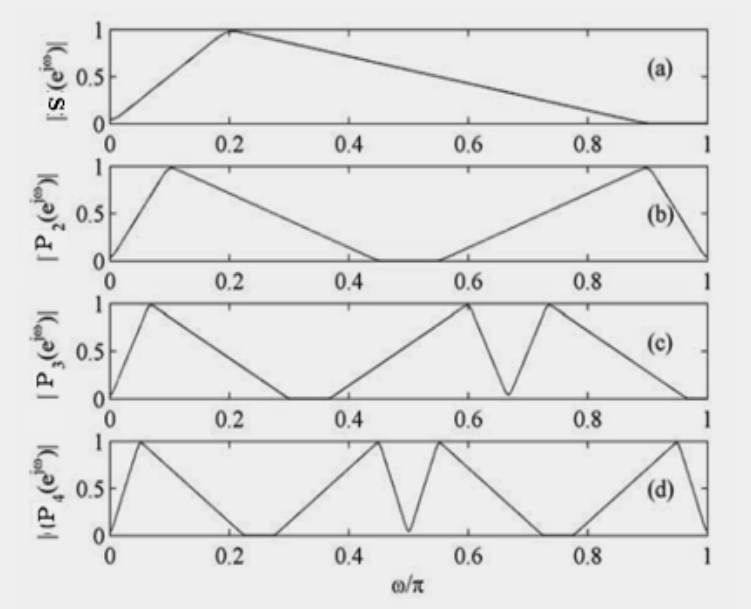

Fig. 2.3G (a) Espectro de s[n] y expansión (b) L=2, (c) L=3, (d) L=4 [Milic 09].

La figura 2.3G ilustra el fenómeno de aparición de imágenes, donde el espectro de la señal original se ve "comprimido", lo que genera produce la inclusión de imágenes del espectro original que anteriormente se encontraban fuera del periodo fundamental del espectro. La figura 2.3H presenta un resumen del proceso de interpolado.

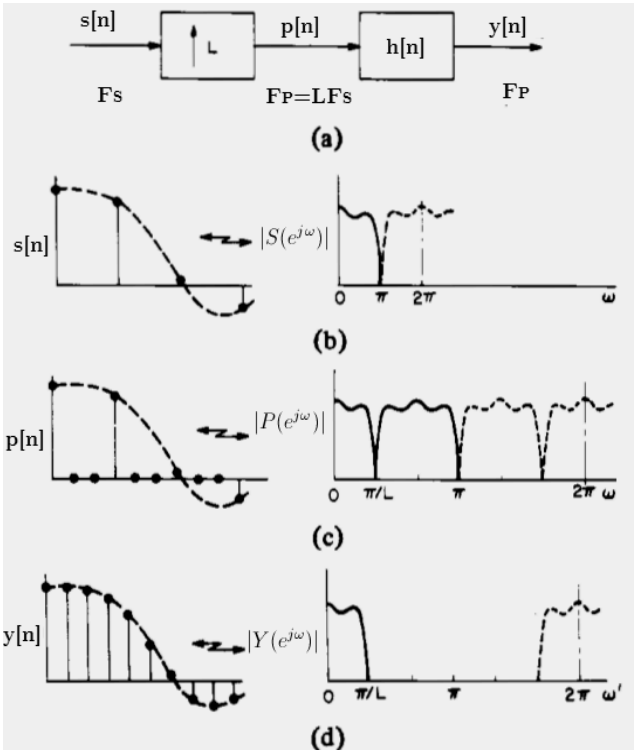

Fig. 2.3H Resumen del proceso de interpolación, caso señal paso bajas [Crochiere 81]. s[n] señal original, p[n] generada al expandir, y[n]señal interpolada y h[n] filtro.

La elección del factor de conversión de la tasa de muestro dependerá de la señal a tratar y en muchos casos la aplicación puede demandar un cambio de tasa por un factor que no es entero. En tal caso, es posible realizar conversiones de la tasa de muestreo por factores racionales L/M, combinando una interpolación por un factor L y un decimado por un factor M (Fig. 2.3I).

Para este procedimiento, conviene realizar primero la interpolación y enseguida el decimado, puesto que este arreglo permite tener una etapa de filtrado posterior a la expansión y una previa a la compresión que operan a la misma frecuencia, por lo que pueden ser combinados en un solo filtro.

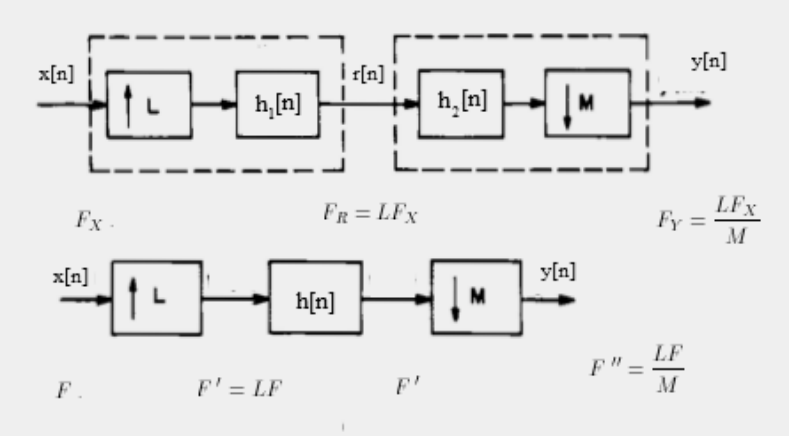

Fig. 2.3l Conversión de la tasa de muestreo por un factor L/M. [Crochiere 81]

El desempeño de un sistema CTM, ya sea que se trate de diezmado o interpolación, está determinado principalmente por las características del filtro [Milic 09], lo que hace importante describir las alternativas existentes para su diseño y así elegir el más conveniente. La siguiente sección ofrece un analisis de las estructuras, tipos de filtros y métodos de diseño más usados en sistemas de CTM.

2.3.3 Filtros para procesamiento de señales de tasa múltiple.

El proceso de filtrado como parte de CTM es un problema clásico de filtrado que no impone restricciones sobre esta tarea, por tanto, es posible hacer uso de cualquier técnica o método a disposición [Crochiere 81]. No obstante, resultaría contradictorio emplear filtros digitales de orden alto (los cuales requieren gran cantidad de cálculos por cada muestra a procesar), puesto que una de las ventajas principales de cambiar la tasa de muestreo es precisamente la reducción de la complejidad computacional.

Así pues, se pretende satisfacer las especificaciones requeridas al mismo tiempo que se provee un filtrado computacionalmente eficiente, haciendo siempre un balance entre el desempeño y la complejidad de la tarea.

2.3.3.1 Filtros IIR y FIR.

Los sistemas discretos lineales e invariantes en el tiempo (LIT) presentan la forma más simple de realizar procesos de filtrado, ya que su comportamiento puede ser expresado a través de ecuaciones en diferencias lineales de coeficientes constantes.

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m] \tag{2.3.9}$$

Para la implementación de estos sistemas, unicamente se requieren retardos, multiplicaciones y sumas, lo que facilita su desarrollo en hardware digital y constituye una de las razones por las que resultan convenientes.

Estos sistemas pueden ser clasificados de acuerdo a la duración de la respuesta que presentan al ser exitados con un impulso unitario $\delta[n]$ pudiendo ser ser de duración finita, filtros *FIR,* o de duración infinita filtros *IIR.*

Los filtros FIR se obtienen cuando en la ecuación en diferencias (2.3.9) se presentan coeficientes $a_k=0$, para $k\neq 0$. Por simplicidad se considera $a_0=1$, y entonces se obtiene:

$$y[n] = \sum_{m=0}^{M} b_m x[n-m] \qquad (2.3.9a)$$

Es posible observar de 2.3.9a que para calcular la salida y[n] se requiere unicamente conocer el valor de de la señal de entrada x[n] actual y las M-1 entradas anteriores.

Se puede demostrar que los coeficientes b_k son identicos a los valores de la respuesta impulso del sistema, por lo tanto:

$$y[n] = \sum_{m=0}^{M} h[n] x[n-m] \qquad (2.2.9b)$$

La implementación de este tipo de filtros, puede realizarse directamente a través de la estructura presentada en la figura 2.3.J

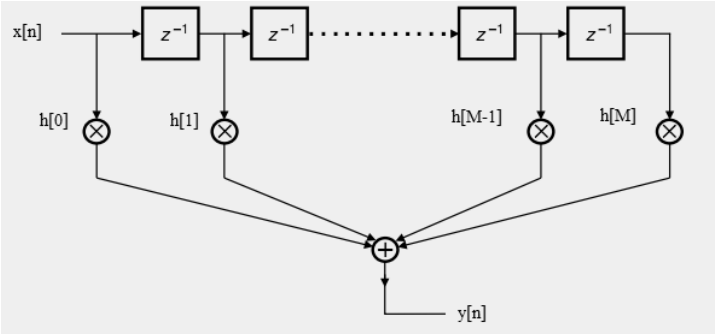

Fig 2.3J Estructura directa de un filtro FIR.

Así pues, diseñar este tipo de filtros, consiste en encontrar los valores h[0], h[1], ... ,h[M-1] que generen un espectro de magnitud lo mas cercano posible a más a la respuesta en frecuencia deseada. La principal ventaja de este tipo de filtros es que pueden exhibir fase lineal (i.e. retardo de grupo constante, véase la sección 2.2.3.4) aunque es difícil conseguir respuestas en magnitud con caídas repentinas con un orden de filtro bajo.

Por su parte, los filtros IIR se obtienen cuando los coeficientes a_k de la ecuación (2.3.9) son no nulos ,al menos para algun a_k ,con $k \neq 0$.

$$y[n] = \sum_{m=0}^{M} b_m x[n-m] - \sum_{k=1}^{N} a_k y[n-k] \qquad (2.2.9c)$$

Para obtener la salida de este tipo de filtros, es necesario concoer el valor actual de la entrada x[n], asi como los M-1 valores anteriores, ademas de los N-1 valores de salida calulados previamente. Esta retroalimentacion, explica cómo un impulso unitario $\delta[n]$ llega a mantenerse en la salida del sistema de manera permanente (Respuesta al impulso de duración infinita).

La forma más directa de implementar un filtro IIR se presenta en la figura 2.3K

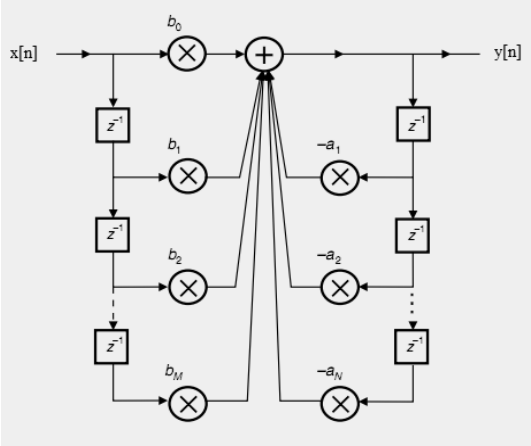

Fig. 2.3K

Estructura básica de un filtro IIR.

A diferencia de los FIR, los filtros IIR no presentan fase lineal, pero a cambio ofrecen una respuestas de magnitud cercanas a la ideal usando filtros con menos coeficientes, por lo que resultan de interés cuando es necesaria una alta selectividad en respuestas de magnitud. Para diseñar un filtro IIR se requiere encontrar los valores $a_1, a_2 \dots a_N$ y $b_0, b_1 \dots b_M$ que permitan obtener la respuesta en frecuencia que más se ajuste a los requerimientos. A continuación se presenta una tabla comparativa entre ambos tipos de filtros.

FIR	IIR
Ventajas: +Diseños de fase lineal son obtenidos fácilmente. +Muchas técnicas de diseño disponibles +Buenas características de cuantización (i.e. el ruido producido por el redondeo puede ser minimizado con una longitud de palabra razonablemente larga en la mayoría de los diseños prácticos) +Siempre son estables +No presentan ciclos límite	**Ventajas:** +Características en magnitud arbitrarias pueden ser fácilmente obtenidas. +Los efectos producidos por longitud de palabra finita(ruido por redondeo, ciclos limites, cuantización de coeficientes) han sido ampliamente estudiados y pueden ser controlados. +Existen una gran cantidad de tecnicas disponibles para filtros analogicos que pueden ser empleadas de manera conjunta con mapeos al dominio digital para el diseño de filtros.
Desventajas: -Respuestas en frecuencia arbitrarias deben ser aproximadas suficientemente bien con respuestas impulso largas. -El número de coeficientes requeridos para obtener bandas de transición angosta suele ser mayor que con filtros IIR	**Desventajas:** -Pueden ser inestables. -Los metodos de diseño suelen ser más complicados.

Tabla 2.T2 Comparativa entre filtros FIR y IIR[Crochiere 81].

Como se ha visto, la salida y[n] de un filtro IIR depende de los valores de salida previos y[n-1], y[n-2], por lo que cada uno de ellos debe ser calculado. Esto resulta ineficiente en procesos de diezmado, donde un proceso de compresión de la tasa de muestreo, posterior al filtrado, unicamente conservará los valores y[n], y[Mn], y[2M], eliminando una cantidad importante de valores que han sido calculados. En contraparte, en un filtro FIR la salida y[n] no depende de salidas previas, lo que permite calcular solamente aquellos valores de salida que vayan a ser conservados despues de realizar el proceso de decimado, por lo cual son una elección común para realizar procesos de decimado.

Conviene resaltar que la elección de filtros para desarrollar una tarea de CTM depende de un número importante de factores, como son: recursos de procesamiento a disposición, velocidad del hardware a disposición, tasa de operación del filtro, especificaciones para la banda de paso y banda de supresión, rizos máximos tolerados en ambas bandas etc. Estos deben considerarse al momento de elegir el enfoque más conveniente de acuerdo a la aplicación.

La presentación que se realiza sobre filtrado, en las siguientes secciones, comienza con algunos aspectos de interés en el diseño de filtros, para después abordar algunos métodos de diseño. Posteriormente se estudian algunas estructuras convenientes para CTM y, finalmente, algunas clases especiales de filtros que presentan ventajas en aplicaciones de tasa múltiple.

2.3.4 Filtrado FIR y fase lineal.

El filtrado ideal (irrealizable físicamente) posee características que pueden ser muy deseables, pero no absolutamente necesarias en diversas aplicaciones. La primera de ellas es que la respuesta en magnitud puede no ser constante en la región de frecuencias de interés (banda de paso), ni en la banda de frecuencias a atenuar o suprimir (banda de supresión), en lugar de ello se puede tolerar una pequeña variación en su magnitud (a las que se denomina comunmente como rizos)(Fig.2.3L)

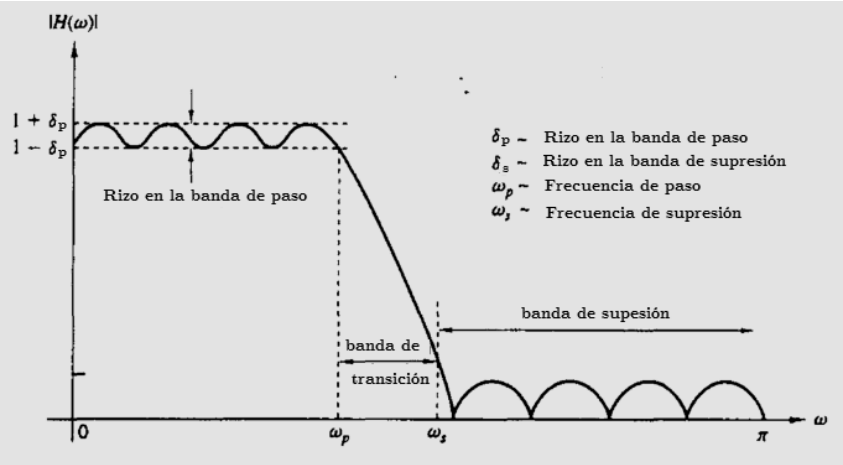

Fig. 2.3L Esquema de tolerancias para filtrado[1]

Además, pueden no exhibir respuesta en fase igual a cero (Retardo nulo entre la entrada y la salida del filtro). Ante la incapacidad de contar con esta condición, la aproximación más aceptable [Oppenheim 10] es el filtrado con fase lineal, lo que significa que los retrasos que exhiben las componentes en frecuencia de la señal, al filtrar, son proporcionales a su frecuencia; permitiendo así mantener relación de fase entre ellas y conservar la forma de onda de la salida**.** Esto permite obtener las componentes en frecuencia deseadas sin distorsion, introduciendo únicamente un retraso en la salida.

Para determinar si un filtro presenta fase lineal es necesario obtener el retardo *retardo de grupo*, definido como la variación de la fase (retrasos que sufren las componentes en frecuencia a su paso por el filtro) con respecto a la frecuencia. Cuando el retardo de grupo es de valor constante, el filtro presenta fase lineal.

$$\tau(\omega) = -\frac{d}{d\omega}\angle|H(e^{j\omega})| \tag{2.3.10}$$

[1] Resulta difícil visualizar la variación en la banda de supresión en una escala lineal, por lo que las respuestas en magnitud suelen ser presentadas en escalas logarítmicas, cuyas unidades son los decibeles.

De manera general, se tiene fase lineal cuando la respuesta en frecuencia del un sistema puede expresarse como:

$$H(e^{j\omega}) = A(e^{j\omega})e^{-\alpha j+\beta j} \tag{2.3.11}$$

Donde $A(e^{j\omega})$ es una función real y α,β constantes reales.

En filtros FIR, esta condición se obtiene fácilmente si se garantiza que la respuesta impulso h[n] es simétrica (h[n]=h[M-1-n]) o anti simétrica (h[n]=-h[M-1-n]). Es importante tener presente la longitud N del filtro, ya que impone ciertas restricciones sobre la respuesta en frecuencia. La tabla 2.T3 ofrece un resumen de las diferentes alternativas de filtros FIR de fase lineal y sus características; posteriormente se presentan secciones que describen metodos de diseño de este tipo de filtros.

	Simetría	N	Restricciones	Observaciones
Tipo I	h[n]=h[M-1-n])	Par	----------	Permite cualquier filtro
Tipo II	h[n]=h[M-1-n])	Impar	H(π)=0	No permite paso altas
Tipo III	h[n]=-h[M-1-n])	Par	H(π)=H(0)=0	Solo permite paso banda
Tipo IV	h[n]=-h[M-1-n])	Impar	H(0)=0	No permite paso bajas

Tabla 2.T3 Filtros FIR de fase lineal.

Esta aparente restricción que la simetria impone sobre los coeficientes del filtro resulta benéfica, ya que permite hacer uso de estructuras que reducen la cantidad de hardware necesario para su implementación; en especifico, permite reducir la cantidad multiplicadores necesarios a la mitad. (Vease Fig 2.3M)

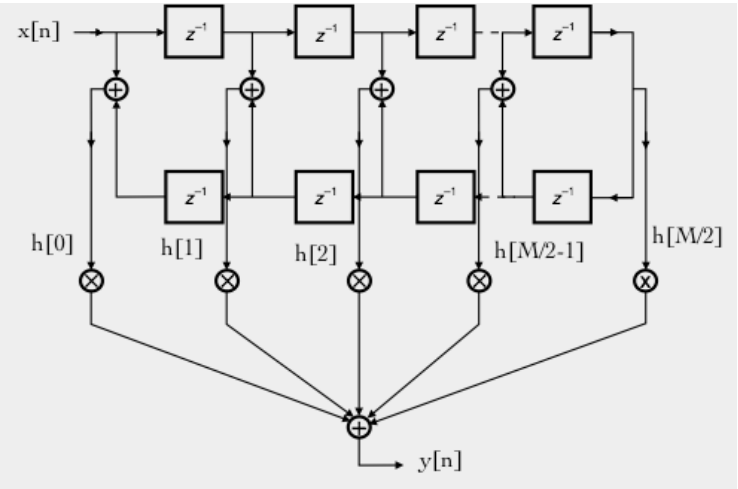

Fig. 2.3M Estructura para filtro FIR de fase lineal, M par.

2.3.4.1 Método de ventanas.

Este método constituye la manera más simple y directa de diseñar un filtro FIR. Para el caso de un filtro paso-bajas se toma como base la respuesta en frecuencia del filtro ideal:

$$H_{id}(e^{j\omega}) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \omega_c < |\omega| < \pi \end{cases} \tag{2.3.12}$$

Misma que corresponde a un sistema con una respuesta impulso infinita:

$$h_{id} = \frac{sen[\omega_c n]}{n\pi} \quad -\infty < n < \infty \tag{2.3.12a}$$

La ecuación (2.3.13a) corresponde a un sistema no causal (irrealizable físicamente). Además, posee un numero infinito de valores, por lo que es necesario limitar de alguna manera la cantidad de valores que la conforman. Para conseguirlo, se puede multiplicar a $h_{id}[n]$ (de longitud infinita) por una función w[n] en la que unicamente algunos de sus valores son no nulos.

$$h_w[n] = h[n]w[n] \tag{2.3.13}$$

El producto de la respuesta impuso del filtro ideal y la función ventana produce una secuencia numérica que corresponderá a la respuesta impulso de un filtro FIR cuya respuesta en frecuencia, será a una aproximación de la respuesta ideal que se haya tomado como base.

La ventana más simple que produce un truncamiento directo de la respuesta impulso ideal, se conoce como ventana rectangular.

$$w_R[n] = \begin{cases} 1, & 0 \leq n \leq M \\ 0, & otro\ caso. \end{cases} \tag{2.3.14}$$

Donde M es la longitud deseada para el filtro.

Al emplear una ventana rectangular, la eliminación de muestras de la respuesta ideal origina la presencia del *fenómeno de gibbs*, que se manifiesta en un rizo en las bandas de paso y de supresión. Se ha observado que la amplitud de dicho rizo no decrece al incrementar la longitud del filtro, por ello, la ventana rectangular no es comúnmente implementada en la práctica. [Crochiere 81].

En la busqueda por mejorar la calidad de la respuesta en frecuencia, se han propuesto multiples funciones ventana, las cuales ofrecen diferentes caracterisicas, de entre las cuales puede elegirse aquella que ofrezca el resultado mas adecuado para la aplicación. La figura 2.3N y la tabla 2.2T presentan algunas funciones ventana y sus caracteristicas. En este tipo de diseño, se cuenta dos parámetros para modificar la respuesta del filtro, la forma de la funcion ventana w[n] y la longitud de la misma.

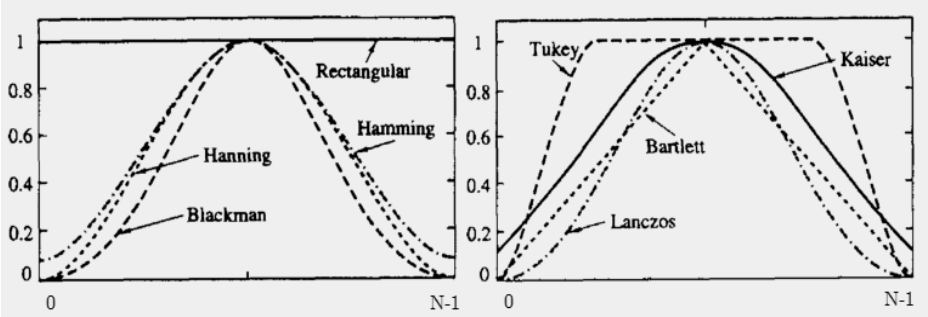

Fig. 2.3N Formas de varias funciones ventana [Proakis 07].

Ventana	Ancho del lóbulo principal[1]	Atenuación del lóbulo lateral(dB)[2]
Rectangular	4π/N	-13
Bartlett	8π/N	-27
Hanning	8π/N	-32
Hamming	8π/N	-43
Blackman	12π/N	-58

Tabla 2.T4 Características en frecuencia de varias ventanas

Observe de la tabla 2.T4 que solo el ancho del lóbulo principal depende de la longitud del filtro, por ende el camino a seguir cuando se emplea el metodo de ventanas suele ser: elegir la ventana que brinde la atenuación deseada en la banda de supresión, para posteriormente ajustar el ancho de la banda de paso modificando la longitud del filtro.

[1] En la respuesta de magnitud en escala logarítmica, el lóbulo principal hace referencia a la banda de paso del filtro.
[2] El lóbulo o lóbulos laterales, hacen referencia al pico máximo en el inicio de la banda de supresión.

El metodo de ventanas resulta fácil de usar, ya que existen expresiones disponibles para la obtención de los coeficientes de las ventanas que se ajusten a diferentes requerimientos; no obstante, este método exhibe ciertas limitaciones:

i) Los rizos en la respuesta en magnitud son de la misma amplitud, mientras que en la práctica, la tolerancia en el valor de la amplitud aceptable para cada región del espectro es distinta ya que las variaciones tolerables en la banda de paso pueden ser hasta 10 o 100 veces mayores que en la banda de supresión [Harris04].

ii) Se cuenta con un control muy limitado en la elección de las frecuencias de corte y de paso, exhibiendo errores considerables respecto a las caracteristicas deseadas en la mayor parte de los casos.

iii) Es común encontrar un filtro FIR con una cantidad menor de coeficientes que sea capaz de satisfacer una condición deseada.

Se ha encontrado que flexibilizando los requerimientos en la variación máxima para en la banda de paso, da origen a un diseño menos estricto y por ende de menor complejidad computacional, la siguiente sección presenta algunas alternativas disponibles que hacen uso de este hecho.

2.3.4.2 Diseño optimo de filtros FIR.

Se han desarrollado varios procedimientos computacionales que, a través de iteraciones, permiten obtener el filtro con el mejor ajuste a los requerimientos impuestos, con la menor cantidad de coeficientes posible. La idea detrás de estos algoritmos es, como se ha comentado anteriormente, imponer tolerancias diferentes para el rizo en la banda de paso y de supresión, lo que ofrece una mayor flexibilidad que el método de ventanas.

Dos enfoques para diseño han sido propuestos [Oppenheim 10].

- Herrmann, Shlüsel et al. Introdujeron un procedimiento en el que la longitud del filtro N, δ_P y δ_S se mantienen fijos y ω_P, ω_S se hacen variar de manera iterativa hasta encontrar la respuesta óptima.

- Parks, McClellan, Rabiner. Propusieron un algoritmo que mantiene los valores N, ω_P, ω_S , así como el cociente δ_P/δ_S ,constantes, variando consecutivamente δ_P o δ_S.

Este último ha ganado mayor aceptación dada su mayor flexibilidad y gran eficiencia computacional . Su planteamiento es transformar un problema de diseño de filtros en una aproximación polinomial, debido a que la respuesta en frecuencia de un filtro de fase lineal, puede expresarse como un polinomio en potencias de cosenos[1.]

La solución a este problema está basada en el teorema de alternancia, el cual establece que existe solamente un polinomio, denominado polinomio de Tchevichev, el cual presenta orden mínimo y satisface al esquema de tolerancias planteado [Meyer 07]. La aproximación polinomial es planteada con un algoritmo que permite minimizar el valor de una función de error $E(\vartheta)$ haciendo uso de las funciones: $W(\vartheta)$, la función de peso del error y $T(\vartheta)$ la función objetivo (Respuesta deseada), modificando $H(\vartheta)$.

$$E(\theta) = W(\theta)[H(\theta) - T(\theta)] \qquad (2.3.15)$$

El algoritmo encuentra la ubicación de las desviaciones máximas que aparecen en relación con la respuesta original, para posteriormente reducir el tamaño del error más grande en cada iteración.

[1] La teoría del algoritmo es presentada a detalle en [Parks 73].

Este procedimiento permite encontrar siempre la *respuesta óptima* al problema, garantizando que el filtro encontrado para un esquema de tolerancias dado, presenta la respuesta con el menor error.

Puesto que este metodo requiere conocer la longitud del filtro con la que se debe iterar, comunmente se apoya de una formula para estimar la longitud del filtro en función de los requerimientos [Oppenheim 10]:

$$N = \frac{-10log_{10}\left(\delta_1\delta_2\right) - 13}{2.34(\omega_s - \omega_p)} \tag{2.3.16}$$

En la actualidad, existen un gran numero de programas de computo, desarrollados para encontrar los coeficientes h[n] a partir de un esquema de tolerancias que facilitan el diseño de filtros, por lo que este método es elección común en el diseño de filtros FIR de fase lineal.

2.3.5 Implementación polifase.

Como se ha presentado previamente, decimar e interpolar son procedimientos que imponen el uso de filtros eficientes. Sin embargo, la implementación directa, de un sistema de CTM, implica realizar calculos que pudieran no ser necesarios [Lyons 10]. En el caso de decimado, el proceso de compresión de la tasa de muestreo posterior al filtrado, elimina muestras que han sido calculadas de manera inecesaria. Por su parte, en la interpolación, deben ser realizadas una cantidad importante de multiplicaciones por cero, cuyo resultado, evidentemente, no requiere efectuarse puesto que el resultado se conoce previamente.

Una forma de optimizar el rendimiento, se logra imponiendo el uso de una *estructura polifase,* la cual permite reducir de manera importante la cantidad de calculos a realizar y con ello, la cantidad de hardware requerido en la implementación. Esta estructura esta basada en la *descomposición polifase* de una señal discreta, que se presenta a continuación.

2.3.5.1 Descomposición polifase.

Cualquier señal discreta x[n] puede ser separada en M componentes, denominadas polifase (Denotadas , P_0, P_1.. $P_{(M-1)}$), cuya superposición permite obtener la señal original.

$$x[n] = x_{P0}[n] + x_{P1}[n] + x_{P2}[n] + \ldots + x_{P(M-1)}[n]. \tag{2.3.17}$$

Dónde

$$x_{Pk} = x[n]s_k[n] \tag{2.3.18}$$

Siendo

$$s_k[n] = \begin{cases} 1 & n = k, k \pm M, k \pm 2M \ etc. \\ 0 & otro \ caso \end{cases} \tag{2.3.19}$$

En la figura 2.3O puede observarse graficamente que la superposición de sus componentes polifase da origen a la señal x[n].

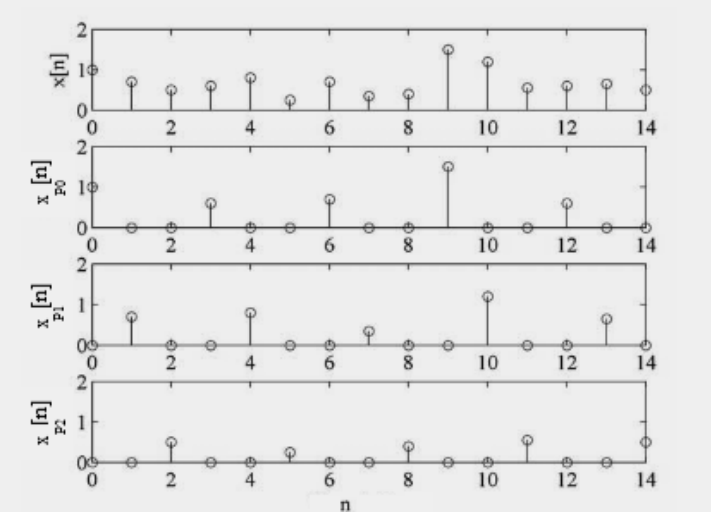

Fig. 2.2O Descomposición de x[n] en 3 señales polifase Po, P1, y P2 [Milic 07].

2.3.5.2 Estructuras polifase.

Al descomponer la respuesta impuso de un filtro en M subsecuencias polifase, se obtienen estructuras de filtrado eficientes, que permiten explotar el paralelismo, al efectuar operaciones básicas simultáneamente. Si se expresa la k-ésima componente polifase de la respuesta impulso del filtro como:

$$p_k[n] = h[Mn + k] = h_k[Mn] \tag{2.3.20}$$

la respuesta impulso estará dada como la suma de todas las componentes:

$$h[n] = \sum_{k=0}^{M-1} h_k[n-k] \tag{2.3.21}$$

Con una transformada Z.

$$H(Z) = \sum_{k=0}^{M-1} Z^{-k} H_k(Z^M) \tag{2.3.22}$$

De (2.3.22) se puede obtener la estructura necesaria para la implementación del filtro, tal como se presenta en la figura 2.3P.

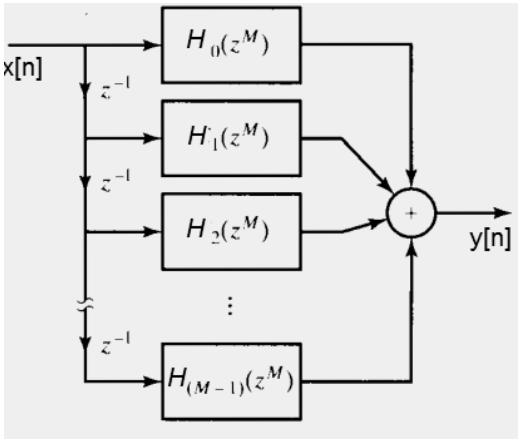

Fig. 2.3P Descomposición polifase de un filtro.

Esto permite que la carga de trabajo de un filtro de longitud N sea distribuida en M filtros de longitud menor (N/M).

En procesos de decimado, esta tarea se realiza tal como se muestra en la figura 2.3Q a). Al decimar, es conveniente descomponer la respuesa impulso del filtro en una cantidad de componentes polifase igual al factor de decimado, tal y como se muestra en 2.3Q b). Puede mostrarse que es posible intercambiar el orden de los bloques de filtrado y compresión sin alterar el sistema, y con ello conseguir *una estructura que filtra solo aquellas muestras que no serán eliminadas.*

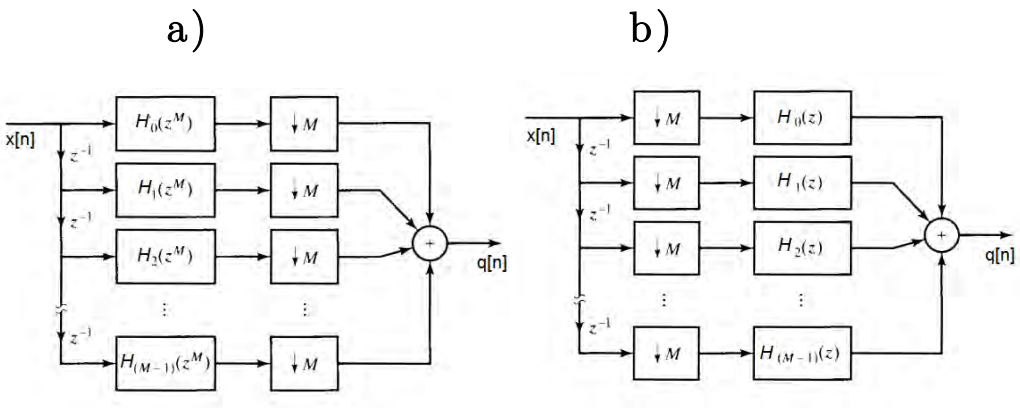

Fig. 2.3Q Descomposición polifase de un decimador.

Así pues, la eficiencia radica en que la implementación directa requiere N multiplicaciones y (N-1) sumas por unidad de tiempo; mientras que en una implementación polifase, con M filtros de N/M coeficientes. se requieren $\frac{1}{M}$(N/M) multiplicaciones por filtro y $\frac{1}{M}$(N/M-1) sumas. Lo que se traduce en un total de (N/M) multiplicaciones y (N/M-1) sumas por cada valor que ingresa al filtro. [Oppenheim 10]

De manera análoga, una descomposición polifase puede aplicarse a un interpolador, al mismo tiempo que el intercambio de bloques, con lo cual se consigue eliminar cálculos no necesarios dentro del proceso y reducir la complejidad computacional en los filtros. Su implementación directa requiere NL multiplicaciones y (NL-1) sumas por unidad de tiempo, mientras que una implementación polifase, tiene L filtros con N/L coeficientes, logrando reducir el proceso a solo L(N/L) multiplicaciones y L(N/L-1) + (L-1) sumas[1] (Fig. 2.3R).

a) b)

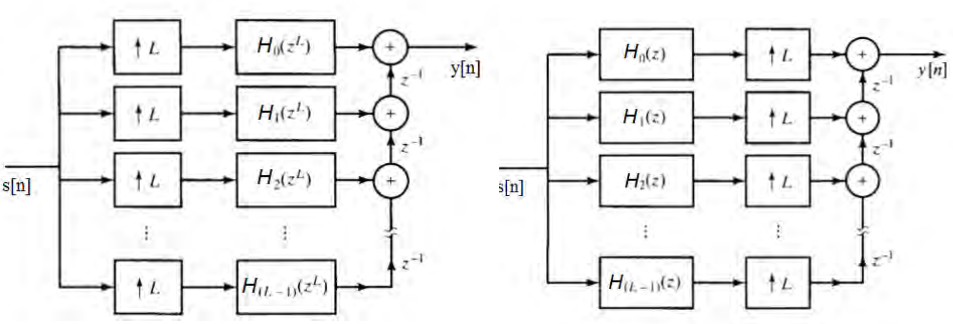

Fig. 2.3R Descomposición polifase de un interpolador [Oppenheim 10].

Este tipo de estructuras son sencillas de realizar ya que los bloques de compresión, expansión y retrasos, pueden ser reemplazados por interrumptores rotatorios conocidos como modelos de conmutador [Vaidyanathan 93], que se implementan fácilmente en hardware digital como máquinas de estado. (Fig. 2.3Ra).

[1] Únicamente es posible si se elige una cantidad de componentes igual al factor de interpolación L=N.

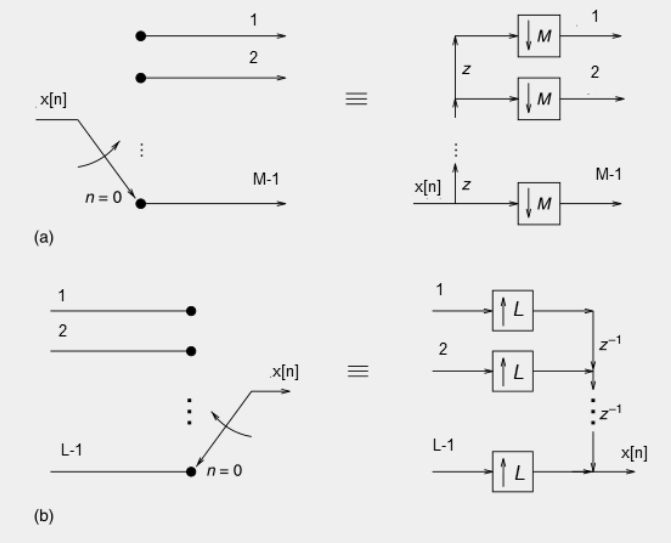

Fig. 2.2Ra Modelo de conmutador de filtros polifase [Vaidyanathan 93].

2.3.6 Implementaciones multietapa.

Diversas áreas del PDS, entre las que se encuentran los RDS, requieren filtros con banda de paso angosta y tolerancias en los rizos relativamente bajas. Además, se conoce que la longitud de un filtro es proporcional, de manera inversa a la magnitud de las tolerancias δ_S y δ_P, y a la longitud de la banda de transición. Es por ello que el diseño e implementación de filtros que satisfagan los requerimientos con un costo computacional razonable, son difíciles y a veces imposibles de implementar con una sola etapa de filtrado.

Una forma de resolver este problema es reemplazando un bloque de CTM convencional por un grupo de sub elementos de conversión conectados en cascada. Haciendo uso de esta técnica, se consigue que las especificaciones para cada uno de los elementos del grupo sean más flexibles [Milic 07], ya que los requerimientos de un solo filtro son compartidos entre todas las etapas del proceso, consiguiendo con ello trabajar con un grupo de filtros de orden menor.

A priori parecería que incrementar la cantidad de etapas[1], aumenta la cantidad de cálculos a realizar. No obstante, se ha mostrado [Crochiere 81] que esta técnica ofrece:

- Reducir significativamente el costo computacional total en comparación con una implementación de una sola etapa.
- Disminuir el número de localidades necesario para almacenar los coeficientes y resultados intermedios.
- Simplificar los problemas de diseño de filtros.
- Menor sensibilidad a los efectos de tener longitud de palabra finita. (e.g. Ruido por redondeo, sensibilidad de los coeficientes etc.)

En el caso de un decimador, las primeras etapas de filtrado trabajan a tasas de muestreo elevadas, poseen banda de transición amplia y, por ello, son filtros con longitudes pequeñas. Por su parte, las últimas etapas poseen una banda de transición angosta, pero operan a frecuencias de muestreo bajas. Esta combinación conduce a que la cantidad de cálculos se mantenga baja en todas las etapas.

Un análisis similar conduce a la justificación del uso de filtros en varias etapas para interpoladores, donde las especificaciones más restrictivas se encuentran en las etapas iniciales que operan siempre a frecuencias más bajas.

[1]Para poder efectuar la CTM en múltiples etapas, es conveniente que los factores de conversión L y M puedan ser expresados como un producto de N números enteros, caso en el cual se podrá diseñar un sistema de cambio de la tasa de muestreo con N etapas.

Es importante destacar que en la implementación en varias etapas, la estructura interna y tipo de cada filtro que puede usarse no tiene restricción alguna, por lo que puede emplearse cualquier tecnica o estructura disponible. Para decimado, es necesario que cada etapa del procesamiento esté protegida contra el solapamiento espectral desde banda base hasta la frecuencia $F_X/2M$. De esta forma, todo el ruido encontrado fuera de esta banda, es removido en las etapas posteriores, lo cual puede observarse en la figura 2.3S.

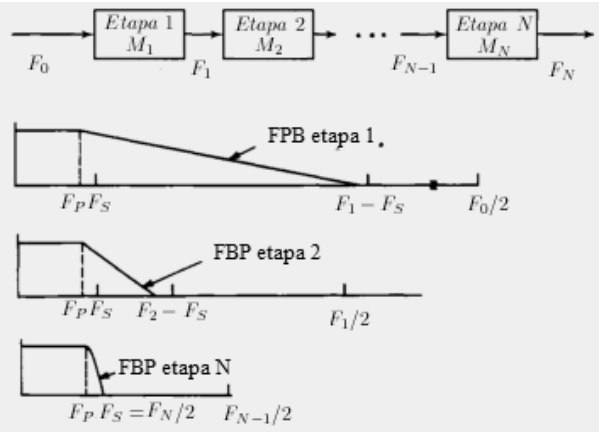

Fig. 2.3S Decimador Multietapa [Crochiere 81].

Este procedimiento resulta particularmente útil cuando:

i) El factor de interpolación o decimado es grande: empíricamente, se han visto buenos resultados con valores mayores a 20 [Lyons10].

ii) El factor de conversión es racional L/M y cercano a uno, con factores L y M relativamente grandes.

No obstante, es necesario considerar que:

- Todas las etapas de CTM introducen distorsión (solapamiento espectral en el caso de compresión e imágenes en la respuesta en frecuencia para el caso de los expansores) y el filtrado no permite eliminar completamente estos fenómenos.

- Es difícil elegir los valores apropiados para el numero de etapas a emplear y el factor de conversión que debe emplearse en cada etápa[1].

- Al tener un sistema de CTM de N etapas, las imperfecciones que se presentan en la bandas de paso en etapas intermedias se acumulan, por lo que podría ser necesario imponer restricciones más estrictas en cada etapa que las que se requieren para implementar una sola etapa[2,3].

[1] Es siempre benéfico diseñar decimadores de varias etapas con un factor de decimado cada vez menor (i.e. $M1>M2>M3\ldots..M_N$).
En el caso de un interpolador, resulta conveniente que el factor de interpolación aumente con cada etapa (i.e $L1<L2<L3\ldots..L_N$).[Lyons10]

[2]En el caso más restrictivo, cada una de las N etapas debe ser diseñada con una tolerancia de $\delta p/N$. Donde δp es la tolerancia requerida para el sistema en su conjunto y N la cantidad de etapas a implementar.

[3]Empíricamente, se conoce que es posible probar en primera instancia con un valor cercano a δp en cada etapa y analizar los resultados, para reducirlo,solo en caso de ser necesario [Meyer 07].

2.3.7 Filtros FIR de banda L-ésima.

Además de modificar la estructura de los filtros para reducir el número de operaciones a realizar en procesos de CTM, algunos estudios sobre las propiedades de los filtros han dado como resultado una clase de filtros FIR de fase lineal que han sido de gran interés en sistemas de comunicaciones [Lyons 10].

Se trata de una familia de filtros conocidos como filtros de banda L-sima (FBL), que se caracterizan por exhibir valores nulos en su respuesta impulso de acuerdo a un patrón definido.

Para conseguir esta propiedad, se deben imponer algunas condiciones a la respuesta en frecuencia deseada:

i) Frecuencia de paso y de supresión dadas como: [1,2]:

$$\omega_p = (1-\rho)\pi/L \qquad \omega_s = (1+\rho)\pi/L \tag{2.3.23}$$

y relacionadas entre si de acuerdo a :

$$\omega_p - \omega_p = 2\pi\rho/L \tag{2.3.24}$$

ii) Tolerancias en las bandas de paso y supresión que cumplan con:

$$\delta_p = (L-1)\delta_s \tag{2.3.25}$$

iii) Respuesta impulso con un número impar, 2K+1, de coeficientes

---

[1] L es cualquier número entero positivo.

[2] ρ es un valor en el intervalo (0,1) y su elección determina la relación entre ω_S y ω_P , tal como está expresado por (2.3.24).

Si se garantizan tales condiciones durante el diseño del filtro, éste presentará (Fig. 2.3T):

- Frecuencia de corte, ω_c , localizada en π/L.
- Ceros en su respuesta impulso cada L muestras, a partir del valor intermedio h[K].
  (i.e. h[K±L]= h[K±2L]= h[K±3L]...etc. Serán nulos).
- h[K]=1/L.
- h[n] con valores simétricos respecto a h[K] y, con ello, fase lineal.

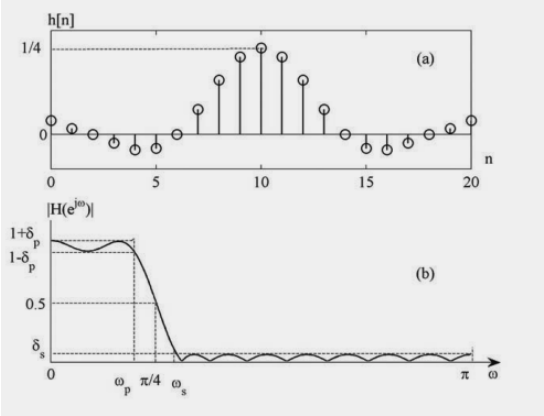

Fig. 2.3T Filtro de banda L-ésima con L=4.[Milic 07]

Al observar figura 2.3T se observa en a), la aparición de ceros en las muestras h[10±4], h[10±8] y una amplitud en h[10] igual a1/4, así como 21 coeficientes simétricos con respecto a h[10] y en b) que la frecuencia de corte aparece en $\pi/4$.

2.3.7.1 Filtros de media banda.

Un análisis de los filtros de banda L-sima permite observar que al reducir el valor L, aumenta la cantidad de valores nulos de su respuesta impulso h[n]. Por consiguiente, para maximizar la cantidad de ceros que aparecen y con ello reducir la cantidad de operaciones a efectuar, se elige L=2, lo que permite lograr filtros con casi la mitad de coeficientes iguales a cero. Con ello, la cantidad de operaciones que éste deberá realizar para obtener una salida se verá reducida prácticamente a la mitad con respecto a otro filtro FIR con la misma longitud [Milic 07].

Así pues, de acuerdo con (2.3.24) y (2.3.25) los filtros deberán ser diseñados para exhibir frecuencias de paso y supresión que cumplan con:

$$\omega_s = \pi - \omega_p \tag{2.3.26}$$

Además de tolerancias de la misma magnitud para ambas bandas.

$$\delta_P = \delta_s = \delta \tag{2.3.27}$$

Con ello, se obtendán filtros donde:

$$h[K] = 1/2 \tag{2.3.28}$$

$$h[K \pm 2] = 0 = h[K \pm 4] = h[K \pm 6] = h[K \pm 8] \ldots etc$$

Con frecuencia de corte ubicada en $\pi/2$, coeficientes simétricos respecto al valor central, fase lineal y respuesta en frecuencia simétrica.

$$H(\omega) + H(\pi - \omega) = 1 \tag{2.3.29}$$

Asimismo, los filtros de media banda resultan de interés debido a que permiten realizar eficientemente procesos de interpolación o decimado cuando se tienen factores de CTM múltiplos de dos.

Para obtener la estructura que permite implementar este tipo de filtros, se calcula su transformada Z a partir de (2.3.28) y se toma en cuenta que su respuesta impulso es simétrica, con lo cual :

$$\begin{aligned} H(Z) = h[0] + h[2]Z^{-2} + h[4]Z^{-4} + \ldots \cdot \\ + h[k-1]Z^{-(k-1)} + 0.5Z^{-k} + h[k-1]Z^{-(k+1)} + \ldots \\ + h[2]Z^{-(2k-2)} + h[0]Z^{-2k} \end{aligned} \tag{2.3.30}$$

Que puede ser expresada como:

$$H(Z) = H_{P0}(Z^2) + Z^{-1}H_{P1}(Z^2) \tag{2.3.31}$$

Siendo:

$$\begin{aligned} H_{P0}(Z) \;=\; h[0] + h[2]Z^{-1} + h[4]Z^{-2} + \ldots \\ + h[k-1]Z^{-(k-1)/2} + h[k-1]Z^{-(k+1)/2} + \ldots \\ + h[2]Z^{-(2k-2)/2} + h[0]Z^{-2k/2} \end{aligned} \tag{2.3.32}$$

$$H_{P1}(Z) = 0.5Z^{-(k-1)/2} \tag{2.3.33}$$

Finalmente, al reescribir (2.3.30) se obtiene una implementación computacionalmente más económica, muy usada en aplicaciones de procesamiento digital de alta frecuencia y ancho de banda angosto [Lyons 10](Fig. 2.3.U), la cual presenta, un filtro con la casi mitad de sus coeficientes iguales a cero. Además, los k coeficientes no nulos seran simetricos, por lo que puede usarse una estructura de fase lineal para implementar $H_{PO}(Z)$, lo cual reduce el numero de multiplicaciones necesarias a solo una cuarta parte.

$$H_{P0}(Z) = h[0](1+Z^{-2k/2})+h[2](Z^{-1}+Z^{-(2k-1)/2})+\ldots \\ +h[k-1](Z^{-(k-1)}+\ Z^{-(k+1)}) \qquad (2.3.34)$$

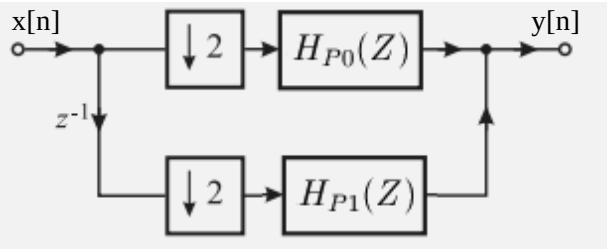

Fig. 2.3U Implementación de filtro de media banda

Ahora bien, multiplicar como parte de un proceso de filtrado demanda capacidad y tiempo de procesamiento, lo cual se resalta en el hecho de que durante tiempo fuese considerado como un parámetro para medir la capacidad de procesamiento de un sistema. [Lyons 10]. Los receptores digitales, al trabajar con señales de alta frecuencia, requieren procesar datos a tasas muy elevadas, que requieren el uso de las mejores herramientas a disposición; Sin embargo hasta este el momento se han presentado vías para reducir el número de multiplicaciones que se realizan durante el filtrado pero ninguna de ellas logra eliminarlas por completo. La siguiente parte de este capítulo discute una forma de eliminar completamente las multiplicaciones.

2.3.8 Filtros peine integrador en cascada (CIC).

Existe una clase de filtros simples que permiten realizar procesos de filtrado haciendo uso unicamente de sumas, restas y acumulaciones; se trata de los filtros de *respuesta impulso rectangular*, los cuales quedan descritos por la siguiente expresión:(Fig. 2.3V)

$$h_C[n] = \frac{1}{N}\begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & otro\ caso \end{cases} \tag{2.3.35}$$

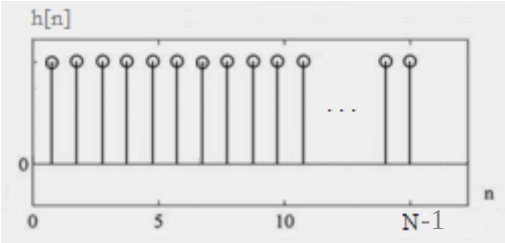

Fig. 2.3V Filtro de media móvil o filtro sinc

De (2.3.35) puede observarse que la salida de este filtro está dada como:

$$y[n] = \frac{1}{N}(x[n] + x[n-1] + x[n-2] + ... + x[n-(N-1)]) \tag{2.3.36}$$

y su transformada Z estará dada como:

$$H_C(Z) = \frac{1}{N}\sum_{k=0}^{N-1} Z^{-k} \tag{2.3.37}$$

Una implementación más eficiente es obtenida sí se efectua la sumatoria (2.3.37):

$$H_C(Z) = \frac{1}{N}\left(\frac{1 - Z^{-N}}{1 - Z^{-1}}\right) \tag{2.3.38}$$

Obteniéndose con ello el filtro:

$$y[n] = \frac{1}{N}(x[n] - x[n - N]) + y[n - 1] \tag{2.3.39}$$

Note que para implementarse, son necesarios únicamente, dos sumadores, independientemente de la longitud N del filtro

La ecuación (2.3.38) suele implementarse en 2 etapas, un filtro peine(comb) y un integrador conectados en cascada[1].

Parte comb	$(1 - Z^{-N})$	$\frac{1}{N}(x[n] - x[n - N])$
		$+$
Parte Integrador	$(1 - Z^{-1})^{-1}$	$y[n - 1]$

Este tipo de filtros poseen la ventaja de no requerir espacio para almacenar los valores de su respuesta impulso, dado que todos son iguales a uno. Presenta, además, respuesta de magnitud paso bajas y fase lineal, puesto que sus coeficientes son simétricos.

[1] Se emplea el termino filtro CIC (*Cascaded Comb Integrator*), cuando el filtro de respuesta impulso rectangular es implementado de acuerdo a la ecuación (2.3.38).

Por otro lado, al reexpresar la ecuación (2.3.38) como:

$$H_C(Z) = \frac{1}{z^{(N-1)}} \frac{Z^N - 1}{z - 1} \tag{2.3.40}$$

De donde puede observarse que los ceros de dicha funcion de transferencia son las N raices del polinomio en z del numerador, las cuales estan distribuidos de manera uniforme sobre el la circunferencia unitaria. Esto se traduce en una respuesta en frecuencia con maxima atencuación en dichos puntos, la Fig.2.3W presenta la respuesta en magnitud de un filtro CIC con N=10;

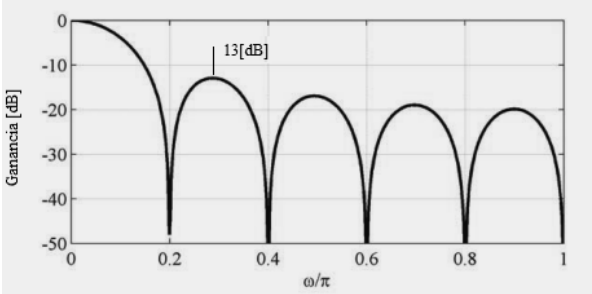

Fig. 2.3W Respuesta en magnitud de un filtro CIC. N=5.

En cuanto a su respuesta en frecuencia, los filtros CIC exhiben atenuación de 13 [dB] en el primer lóbulo lateral de su respuesta de magnitud, contra los 60-80[dB] que muchas aplicaciones requieren [Harris 04]. Es posible mejorar dicha respuesta si se hace uso de varios filtros CIC conectados en cascada Con ello se consigue que las componentes no deseadas o imágenes generadas disminuya al hacer uso de varias etapas [Hogenauer 81] (Fig.2.3X). En (2.3.41) esta expresada la función de transferencia resultante del filtro CIC cuando se tienen k etapas.

$$H_C^k(Z) = \left[\frac{1}{N} \left(\frac{1 - Z^{-N}}{1 - Z^{-1}} \right) \right]^K \tag{2.3.41}$$

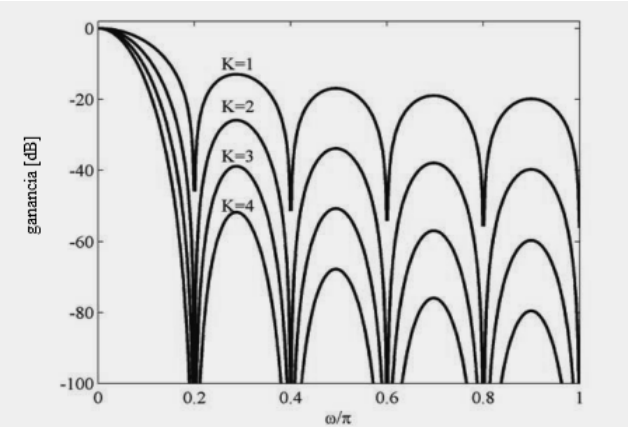

Fig. 2.3X Respuesta en magnitud de varios filtros CIC con k igual al número de etapas [Milic09]

2.3.8.1 Filtros Hogenauer.

En 1981 Eugene Hogenauer [Hogenauer 81] propone hacer uso de filtros CIC para CTM, ya que resultan convenientes cuando se tienen factores de conversión grandes y ancho de banda reducido. Estos filtros permiten el uso de decimadores e interpoladores computacionalmente eficientes cuando se elige un factor de conversión igual a la longitud del filtro (i.e L=N ó M=N), lo que permite que los filtros *peine* sean reemplazados por diferenciadores (Fig.2.3Y, Fig.2.3Z).

Su campo de aplicación, aparece en tareas de procesamiento donde la economía computacional es crítica; donde la tasa de muestreo es tan elevada, que el uso de multiplicadores resulte poco viable; o en casos donde se requieran factores de conversión muy grandes y la cantidad de coeficientes a almacenar resulte un inconveniente.

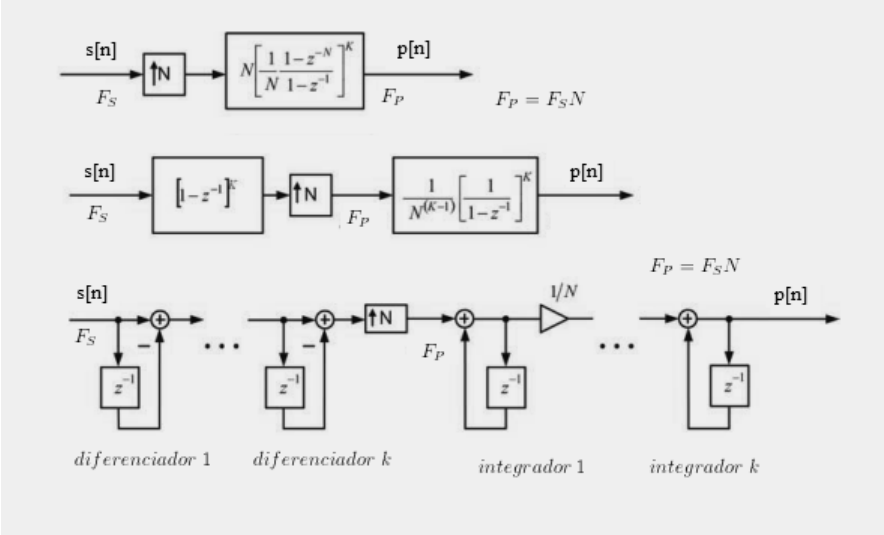

Fig. 2.3Y Filtro Hogenauer interpolador [Milic 07].

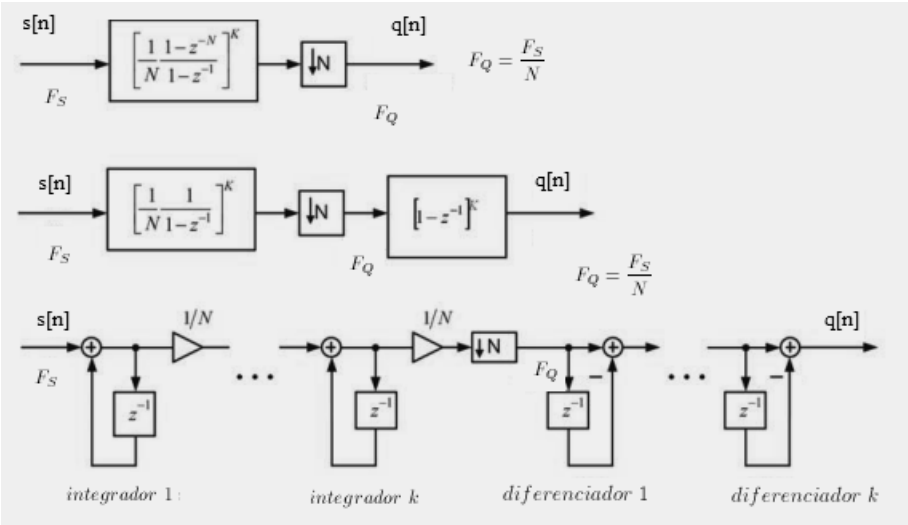

Fig. 2.3Z Filtro Hogenauer decimador [Milic 07].

Al hablar de este tipo de filtros, es importante resaltar que se trata de estructuras eficientes que no están restringidas a cambios de tasa en potencias de dos como los filtros de media banda [Hogenauer 81].

Sin embargo:

i) La envolvente de su respuesta de magnitud, es una función *sinc,* lo que produce una banda de paso irregular.

ii) La respuesta en frecuencia depende únicamente de 2 parámetros, la longitud del filtro N (Debiendo ser igual al factor de conversión L, o M) y la cantidad de etapas K, lo que conduce a un control pobre de la banda de paso.

Por ende, los filtros CIC usualmente están acompañados de una etapa posteriores de filtros FIR de alto rendimiento cuya tarea consiste en compensar las características que presentan los filtros CIC, tanto en la banda de paso como en la de supresión [Lyons 10][Milic 07].

Así pues, con ayuda de los filtros CIC, algun otro tipo de filtro puede ser diseñado para operar a tasa de muestreo baja, donde el número de multiplicaciones por segundo es menor [Hogenauer 81].

2.3.8.2 Desbordamiento y tamaño de acumuladores.

Debido al uso de integradores, existe realimentación positiva en el sistema y los valores en las localidades de memoria que almacenan los resultados pueden presentar desbordamiento. Este hecho se ha estudiado y se sabe que no tiene consecuencias si es empleada aritmética de *complemento a dos* y el rango del sistema de numeración es igual o excede la magnitud máxima esperada en la salida del filtro [Milic 07][Hogenauer 81]. La longitud de palabra que permite que no se presente desbordamiento debe ser de al menos.

$$[W_0 + K\,log_2(N)]\,bits \tag{2.3.42}$$

Donde Wo es la longitud de la palabra de entrada, K el numero de etapas y N el número de bits.

Una vez conocidas las herramientas teoricas que permiten el diseño de un receptor digital, es necesario conocer el hardware digital en el que el diseño puede ser implementado, para lo cual se hace un analisis de los requerimientos y de diversas plataformas disponibles.

2.4 Hardware para PDS

2.4.1 Formatos numéricos.

La implementación de un gran número de tareas de procesamiento digital de señales puede realizarse a través de hardware digital que permita ejecutar: adiciones, multiplicaciones y almacenamiento de información sobre los datos representados en forma digital; existen diversas formas de representar información digitalmente, las cuales se conocen como formato de dato. Su elección determina la precisión y exactitud de la implementación de cualquier algoritmo de procesamiento [Harris 04]. Formatos simples conducen a diseños de hardware sencillos y al mismo tiempo, imprecisos, mientras que formatos complejos dificultan la implementación en hardware pero permiten mayor precisión y/o presentan menos susceptibilidad a errores.

En hardware digital, los números se representan comunmente, como un conjunto de dígitos binarios o *bits* capaces de exhibir dos valores 1 ó 0. Cada grupo de bits capaz de representar un dato suele denominarse palabra binaria el número de bits empleados para la representación se conoce como longitud de palabra. Las señales con las que se trabaja en PDS, pueden exhibir valores positivos y negativos, por lo que un formato adecuado para representar números debe ser capaz de representarlos a ambos. El formato más simple que permite realizarlo es el de *magnitud signada*. Caso en donde se dedica usualmente el primer bit de la palabra binaria para indicar el signo: 0(+) y 1(-).

A pesar de ser simple, no es conveniente para operaciones aritmeticas, ya que suele ser dificil de manejar; además, existen dos formas de representar al cero, lo cual no resutla conveniente.

Un esquema simple más conveniente y es una eleccion común para diseños de hardware para PDS [Harris 04], es el de complemento a dos, debido en gran parte a que permite realizar adiciones y substracciones empleando el mismo hardware y a que los efectos por desbordamiento pueden manejarse mejor [Lyons 10]. En este formato cada digito representa alguna potencia entera de dos, según la posición que ocupe en el número binario. Si se cuenta con N bits, pueden representarse números en el rango $[-2^{(N-1)},2^{(N-1)}-1]$. Para obtener la representación de un numero positivo, basta con encontrar aquellas potencias enteras del numero dos que sumadas den como resultado el numero deseado y a continuación asignar 1 a aquellos bits que representen dichas potencias, dejando en cero los bits restantes. Mientras que para obtener la representación negativa de cualquier número se toma como base el numero con signo positivo, se complementa y finalmente se suma a una palabra binaria de la misma longitud con cero en todos sus bits excepto el menos significativo, descartando todos los bits que se encuentren más allá de la longitud de la palabra original.

Además de números negativos, un formato de datos en PDS debe permitir representar números no enteros o enteros con parte decimal. Una forma simple de conseguirlo es emplear algunos de los bits que representan la magnitud para la parte entera y los restantes para la parte decimal (Fig. 2.4A), lo que se conoce como representación en punto fijo. Esta representación puede ser usada tanto en magnitud signada o bien con complemento a dos, ya que las operaciones aritméticas no se ven afectadas por la posición que ocupe el punto decimal.

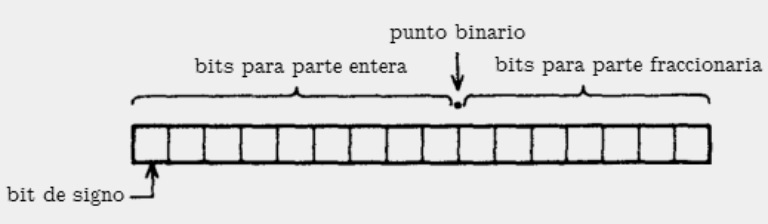

Fig. 2.4A Representación en punto fijo y magnitud signada.

Es importante resaltar que para estas representaciones donde el punto decimal siempre ocupa la misma posición (punto fijo) deben analizarse los requerimientos de procesamiento. Emplear más bits para la magnitud permite ampliar el rango de valores permitidos pero disminuye la precisión. En contraste, hacer uso de más bits para la parte fraccionaria aumenta la calidad de la representación pero reduce el número de valores que pueden ser representados.

Una forma diferente de expresar números reales a través de bits, se conoce como formato de *punto flotante*, también llamada notación real. En este caso, un número es expresado en la forma:

$$x = m2^e \tag{2.4.1}$$

La variable m se conoce como mantisa y e como exponente; esta notación es definida comúnmente para tener una m entre $\frac{1}{2}$ y 1[Harris 04]. Los valores numericos de m y e son expresados como números de punto fijo (Fig 2.4B) y el valor del número almacenado en una palabra binaria en punto flotante se obtiene al desarrollar la ecuación 2.4.1.

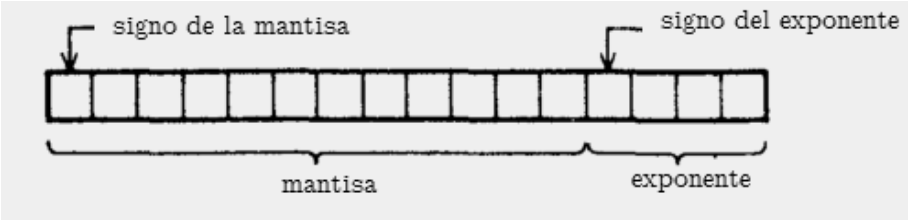

Fig. 2.4B Representación en punto flotante

Esta representación ofrece mayor precisión y es conveniente para realizar cálculos complejos de manera precisa; sin embargo, suele resultar más complicado implementar funciones simples y los cálculos son más lentos; por ello su uso es limitado en aplicaciones de procesamiento digital de señales de alto desempeño.

En lo que se refeire a hardware para implementar un sistema de PDS, es posible hacer uso de microprocesadores de propósito general, procesadores digitales de señales y hardware de arquitectura configurable. En las siguientes secciones de este capitulo se realiza una breve descripción de estas alternativas, resaltando las ventajas y desventajas que representa el uso de cada dispositivo, buscando haciendo enfasis en las restricciones que imponen una arquitectura y un conjunto de instrucciones fijas comparadas con con la versatilidad ofrecida por el hardware configurable.

2.4.2 Microprocesadores.

Los microprocesadores son circuitos electrónicos integrados que poseen una gran cantidad de elementos de procesamiento que les permiten realizar una gran cantidad de tareas. Están basados en el *modelo secuencial de Von Neumann* (MVN) (Fig. 2.4C) y poseen una arquitectura (disposición de sus componentes e interconexiones entre ellos) fija, con un conjunto de instrucciones que no puede ser modificado.

En el MVN se cuenta con una sola memoria (que permite almacenar datos e instrucciones a realizar), una unidad de control (para regular el procesamiento de datos) y una unidad para realizar operaciones aritméticas y lógicas (ALU). Estas unidades interactúan durante la ejecución de cada instrucción para ejecutar diversas tareas; el ciclo de cada instrucción cuenta con tres etapas que son necesarias sin importar la operación a realiazr:

En la primera etapa se realiza la obtención de la instrucción desde una dirección de memoria la unidad de control carga el código de la instrucción a realizar. Durante la segunda erapa (decodificación), la instrucción es interpretada para producir un código de operación (opcode) que permite a la ALU ejecutar la tarea deseada así como obtener la dirección en memoria de él o los operandos necesarios. Finalmente, en la etapa de ejecución se realiza la operación indicada por el opcode para producir el resultado y escribirlo en la memoria.

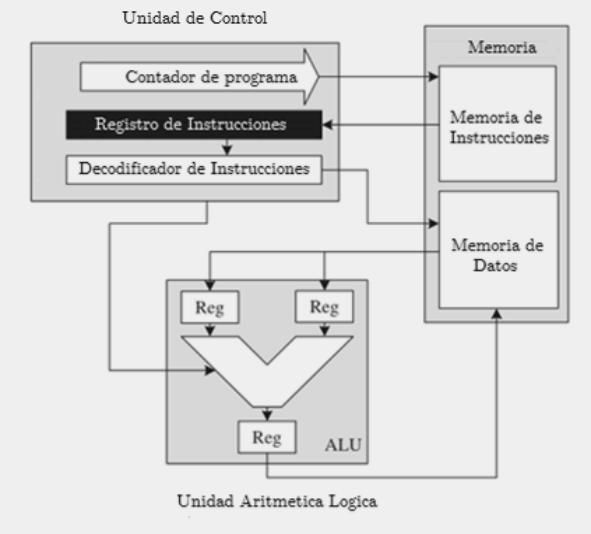

Fig. 2.4C Arquitectura Von Neumann .

2.4.2.1 Programación.

Emplear un procesador de propósito general ofrece al diseñador la posibilidad de indicar las instrucciones a ejecutar a través de dos vías.

1) Con lenguaje de alto nivel como C ó C++, caso en el cual no es necesario conocer la forma en que el procesador está constituido internamente. En este esquema, el desempeño se sacrifica en aras de brindar una mayor facilidad para programar.
2) A través de lenguaje ensamblador, para lo cual debe conocerse la arquitectura interna del procesador y el set de instrucciones básicas del mismo. Esto ofrece un mejor desempeño y mayor control sobre la forma en que las instrucciones son ejecutadas. Aunque la tarea de programación se complica de manera significativa.

2.4.2.2 Limitantes.

En general, esta clase de dispositivos posee un conjunto de instrucciones grande que permite realizar una gran cantidad de operaciones. No obstante, procesar señales digitalmente únicamente requiere algunas de ellas. Asimismo, muchos algoritmos PDS involucran operaciones repetitivas que pueden realizarse de manera simultánea y una gran cantidad de datos circulando a través de la unidad de procesamiento. Cuando se emplea este tipo de dispositivos las unidades del procesador que no son requeridas en un momento dado tienen que esperar hasta que se pasa el control a ellas para ejecutar su tarea. Lo anterior presenta un escenario desfavorable para PDS, conduciendo a desempeños pobres [Ifeachor 01]. Además, ejecutar las operaciones de una por una hace que una gran cantidad de transistores estén inactivos, consumiendo potencia, sin que contribuyan al desempeño del dispositivo. Por ello, la implementación de diseños de PDS en estos sistemas es poco conveniente.

2.4.3 Procesadores digitales de señales (DSPs).

Con la idea de orientar los microprocesadores hacia el procesamiento digital de señales, se han realizado una serie de modificaciones que les permiten alcanzar un mayor rendimiento. Los primeros dispositivos con estas características, a los que se denominó *procesadores digitales de señales* aparecieron en los años 80s.

Para que un microprocesador pueda ejecutar una instrucción nueva debe esperar a que el ciclo obtiene-decodifica-ejecuta previo haya concluido. Esto se debe a que:

i) Si se deseara obtener una nueva instrucción, durante la etapa de decodificación, la unidad de control no estaría disponible para cargar una nueva instrucción.

ii) Al pretender extraer una instrucción nueva de la memoria, durante la etapa de ejecución, esta no podría ser leída; dado que ese tiempo está reservado también para que se escriba en ella el resultado de las operaciones realizadas.

Observe que existiría la posibilidad de extraer una nueva instrucción durante la etapa de ejecución, si los datos y las instrucciones (programa) estuviesen almacenados en memorias diferentes. Lo que ofrecería un mejor desempeño. Con dicha idea en mente, los procesadores digitales de señales se basaron en la arquitectura Harvard, la cual, a diferencia de la Von Neumann, cuenta con una memoria para el programa y otra para los datos (Fig. 2.4D).

Asimismo, cuentan con hardware dedicado que permite realizar operaciones especificas con los datos, tal como las unidades MAC, las cuales permiten multiplicar dos datos y sumar el resultado a un valor acumulado con tan solo una instrucción.

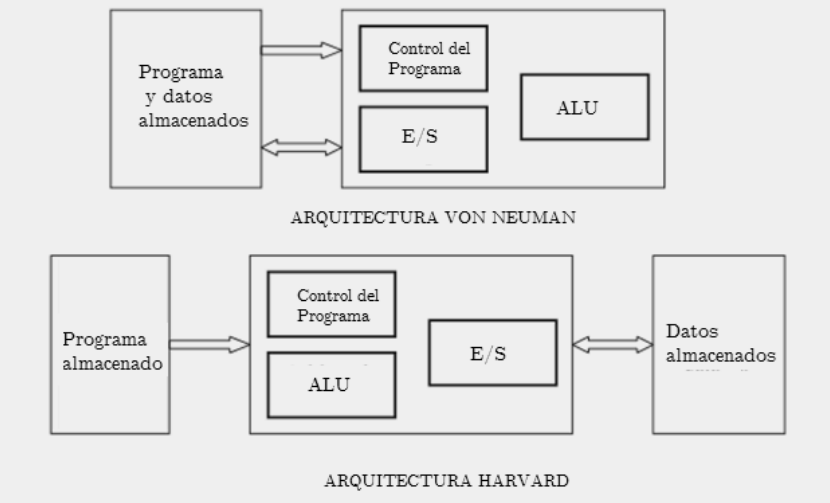

Fig. 2.4D Arquitecturas Harvard y Von Neuman.

Desde su creación, han ocurrido muchos avances en este tipo de arquitecturas [Lee88].

i) Se ha aumentado el tamaño de los buses permitiendo transmitir más datos al mismo tiempo y ejecutar algunas instrucciones en forma paralela.

ii) La inclusión del *pipelining,* que consiste en la descomposición de un proceso en tareas más pequeñas, las cuales se realizan simultáneamente durante la ejecución, eleva la velocidad del procesamiento.

iii) La elección de punto fijo para realizar operaciones, puesto que la aritmética de punto flotante es más costosa y se ejecuta 50% más lento[1].

iv) Una mayor cantidad de *memorias de datos,* que permiten acceder a múltiples datos de manera simultánea. En algunos casos pueden ocurrir hasta seis consultas a memoria en cada ciclo de instrucción, haciendo uso de memorias pequeñas, rápidas y simples.

v) Poseen *modos de direccionamiento* indirecto, que permiten consultar varias localidades de memoria simultáneamente, sin que esto implique indicar la dirección de cada una de ellas, permitiendo emplear pocos bits para representar la instrucción.

vi) Un conjunto de instrucciones especiales optimizadas para DSP, que conducen a un código más compacto, ocupando menos espacio en memoria y produciendo con ello un incremento en la velocidad de ejecución de los algoritmos de PDS.

[1] Al usar este tipo de aritmética, el programador tiene que poner atención en el desbordamiento y el escalamiento.

2.4.3.1 Programación

Al igual que los microprocesadores, las instrucciones a realizar se describen con lenguajes de alto o de bajo nivel, dependiendo del nivel de rendimiento deseado y de la capacidad de los compiladores a disposición. Aunque los leguajes de bajo nivel ofrecerán siempre un mejor desempeño, el uso de lenguajes como C o C++ es fundamental en procesadores muy complejos, ya que la programación con lenguaje de bajo nivel resulta más difícil de emplear.

2.4.3.2 Limitantes

i) La cantidad y el tipo de elementos para procesamiento esta predefinida, por lo que es necesario adaptar los algoritmos a una arquitectura dada, asi que el hardware disponible no resulta óptimo para todas las aplicaciones.

ii) Es necesario un proceso de lectura e interpretación de las instrucciones a realizar (etapas de obtención y decodificación del ciclo de instrucción) durante el procesamiento, lo cual consume tiempo y reduce el rendimiento.

iii) Aunque existen nuevas arquitecturas con muchas unidades operando en paralelo, están limitadas en número y predefinidas por el fabricante; en ciertas aplicaciones cuando se requiere realizar operaciones a muy alta velocidad son ineficientes y conducen a desempeños pobres [Villasenor 98].

Estas limitantes han motivado que en años recientes el interés en el cómputo configurable crezca de manera importante, en búsqueda de una forma mucho más versátil de procesar señales digitalmente.

Con este enfoque es posible diseñar el hardware específicamente para la aplicación de interés, al mismo tiempo que se aprovecha la posibilidad de ejecutar tantas operaciones en forma concurrente como se desee. Además, permite realizar la interpretacion y decodificacion de las instrucciones durante la etapa de diseño del dispositivo y así utilizar el tiempo solo para procesamiento, lo que mejora significativamente el desempeño. La siguiente sección, se ocupa de describir diversas alternativas de arquitectura configurable disponibles.

2.4.4 Hardware con arquitectura configurable.

2.4.4.1 Circuitos integrados de aplicación específica (ASICs).

La manera óptima de realizar el diseño de una arquitectura, para una aplicación en particular, en cuanto a desempeño, consumo de potencia y cantidad de silicio empleado se refiere, es a través de *circuitos integrados de aplicación específica* [Maxfield 04]. Estos son capaces de realizar una gran cantidad de funciones digitales en la forma de un circuito integrado.

En sus inicios, el diseño de estos dispositivos debía hacerse a nivel de máscaras fotosensibles y se fabricaban a la medida. Gradualmente el diseño comenzó a orientarse al concepto de *celdas básicas*. Como ejemplos se pueden mencionar: Las *micromatrices* introducidas por Fairchild en los 60s, que contenían bloques con alrededor de 100 transistores y donde la tarea del diseñador consistía en diseñar las conexiones o Los *micromosaicos* introducidos poco tiempo después que permitían generar hasta 150 compuertas en silicio, a partir de una ecuación booleana almacenada en un archivo de texto.

Con la llegada de la tecnología CMOS, se hizo posible realizar los primeros *Arreglos de compuertas, los cuales son* dispositivos constituidos por varias celdas, que contenían transistores y resistencias, separadas por canales que permitían la interconexión entre ellas.

Se trata del primer dispositivo donde el fabricante ofrecía una serie de librerías y funciones predefinidas a los ingenieros de diseño, lo que facilitaba en gran medida la tarea de la programación. Ofrecían una ventaja de costo debido a la simplicidad del proceso de diseño, pero el resultado estaba lejos de ser óptimo debido al desperdicio de algunos componentes.

Hoy en día hay dos tipos de ASIC dominando el mercado:

i) De *celda estándar,* con origenes a principios de los ochentas. Para el diseño de estos dispositivos, los fabricantes ofrecen una librería de funciones diversas; además, contienen elementos digitales básicos o complejos incluyendo microprocesadores o memorias RAM y ROM. No hacen uso del concepto de celda básica y no hay componentes prefabricados en el chip, lo que permite que la cantidad de componentes y las interconexiones sean cuasi óptimas.

ii) *Estructurados*, surgidos a principios de los 90s. En ellos cada celda básica se realiza mediante multiplexores o tablas de consulta (LUT). Contienen, además, una mezcla de componentes prefabricados como algunos registros y memorias RAM. Se pretende un diseño simple y rápido, ya que únicamente es necesario realizar las conexiones entre componentes prefabricados a través de capas de metalización; reduciendo así el costo y tiempo del diseño. Sin embargo, son tres veces más grandes y consumen de dos a tres veces más potencia que su contraparte de celda estándar [Maxfield 04].

2.4.4.1.1 Limitantes

Si bien el diseño de arquitecturas con ASIC ofrece la alternativa con el mejor rendimiento, existen algunos inconvenientes a considerar:

i) El proceso de diseño y construcción es sumamente demandante en tiempo y costo.

ii) El diseño final queda fijo en el silicio y no puede ser modificado despues de haber sido fabricado, por lo que no pueden corregirse los diseños y el dispositivo se vuelve obsoleto si la aplicación requerida cambia.

iii) No pueden ser programados en un laboratorio o en un ambiente de desarrollo y pruebas.

iv) El costo por dispositivo es muy elevado, únicamente es conveniente cuando los dispositivos son producidos en gran cantidad.

2.4.4.2 Dispositivos lógicos programables complejos (CLPDs)

Desde los años ochenta aparecieron circuitos lógicos programables que permitían realizar funciones diversas, tales como máquinas de estado, funciones booleanas o memorias. Los dispositivos lógicos programables complejos son dispositivos que emplean el principio de celdas básicas, originado en los ASICs. (Fig. 2.4E). Están constituidos por algunas celdas separadas por conexiones configurables que permiten la interconexión entre ellas y con los pines de entrada y salida. La mayor parte de los CPLDs tienen celdas básicas basadas en un PAL, permitiendo implementar una función booleana simple en cada una de ellas. La reconfiguración de estos dispositivos es posible debido a que las partes configurables son celdas EPROM o E^2PROM y en algunos casos SRAM.

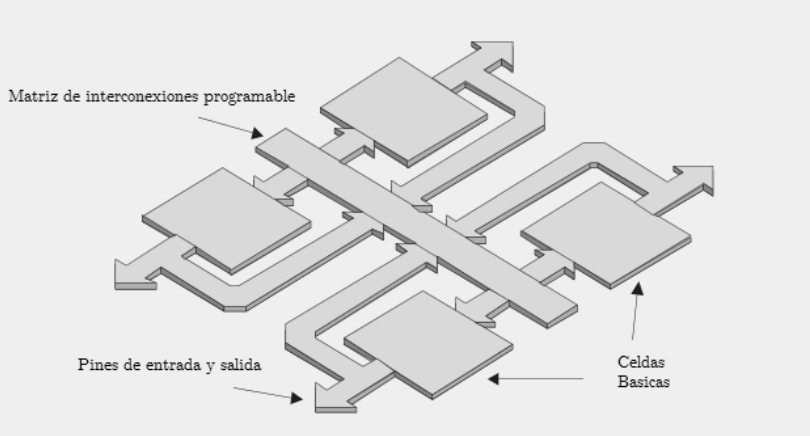

Fig. 2.4E Arquitectura CPLD genérica [Maxfield 04].

2.4.4.2.1 Limitantes

Estos dispositivos son reconfigurables y flexibles pero contienen un número limitado de compuertas, lo que les permite implementar únicamente funciones pequeñas y relativamente simples. Además, no poseen ningún elemento que optimice operaciones aritméticas, vitales en el PDS, por lo que no resultan convenientes para este tipo de tareas.

2.4.4.3 Arreglos de compuertas programables en campo (FPGAS).

En los años 80s Xilinx desarrolla el primer dispositivo que busca reunir lo mejor de los dispositivos para diseño de hardware más usados hasta ese momento. Se pretendía, pues, contar con la gran capacidad ofrecida por los ASIC, para realizar funciones grandes y complejas; así como, con la flexibilidad y versatilidad brindada por la reprogramabilidad de los CPLDs. Con esto se obtiene lo mejor de ambos esquemas a un costo mucho menor respecto a un ASIC.

Los FPGAs[1], contienen bloques lógicos configurables (islas) rodeados con conexiones configurables. (Fig. 2.4F). Estas últimas consisten en segmentos metálicos de longitud fija, que son interconectados con interruptores programables. Las interconexiones metálicas son de diferentes longitudes y permiten la comunicación entre los bloques y de los bloques con los pines de entrada/salida (Fig. 2.4F).

De inicio, eran empleados como herramienta para prototipos ASIC y no eran vistos como dispositivos efectivos para el cómputo de funciones complejas debido a su consumo de potencia relativamente alto y bajas velocidades de reloj [Villasenor 98]. No fue sino hasta los años 2000 que hicieron su aparición los primeros FPGAs de alto desempeño, capaces de realizar funciones equiparables a las de un ASIC.

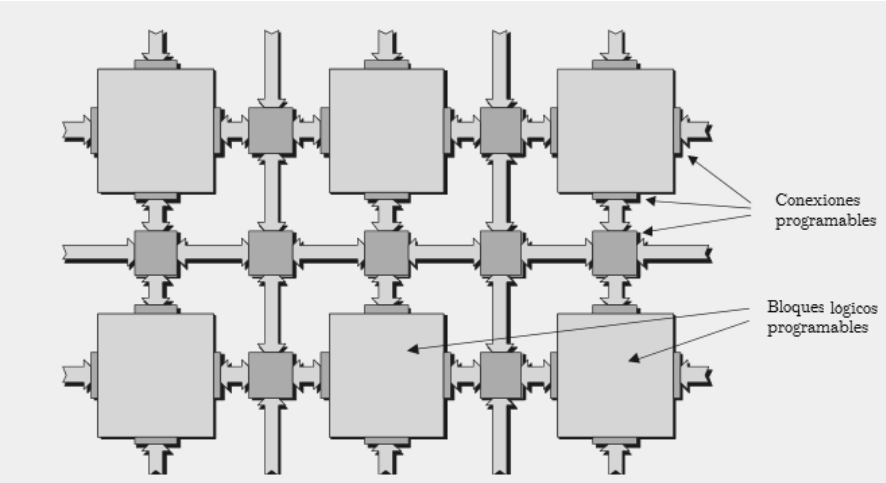

Fig. 2.4F Arquitectura FPGA genérica [Maxfield 04].

[1] La parte de *programable en campo*, del nombre de los FPGAs, se refiere a hecho de que la programación toma lugar en el lugar de trabajo. Contrario a los arreglos de compuertas que se programan durante el proceso de fabricación.

Además de la gran cantidad de lógica programable presente en el dispositivo, se han incluido una serie de elementos, tales como: bloques de memoria RAM embebidos, núcleos de procesadores y bloques de entrada/ salida de alta velocidad, que lo han catapultado a una mayor cantidad de aplicaciones. Entre estas aplicaciones destaca, evidentemente, el procesamiento de señales.

2.4.4.3.1 Tecnologías empleadas en FPGAs.

La mayoría de los FPGAs poseen celdas de configuración a base de memoria estática de acceso aleatorio SRAM. Esta tecnología ofrece una implementación rápida de nuevos diseños y se encuentra siempre a la vanguardia. Es una elección común entre fabricantes de FPGAS debido a que el costo de investigación y desarrollo en esta tecnología corre a cargo de las compañias que fabrican memorias para computadoras [Maxfield 04]. Una celda SRAM completa, que permite almacenar 1 bit está constituida por 6 transistores, tal y como se muestra en la figura 2.4G.

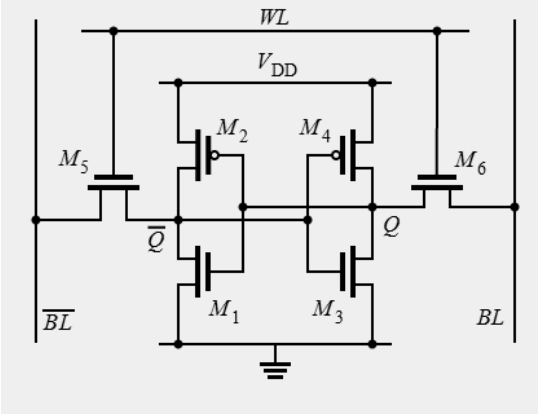

Fig. 2.4G Celda SRAM

Dada la estructura de la celda básica, la tecnología SRAM es volátil, lo que significa que los bits almacenados en estas celdas únicamente se mantienen mientras el dispositivo está energizado, en caso contrario existe una pérdida total de información. Lo anterior hace necesario un proceso de reconfiguración cada vez que se inicia el dispositivo, por lo que se requiere un dispositivo adicional para programación del FPGA, aumentando la cantidad de recursos y espacio necesarios para su implementación.

Como alternativa, los fabricantes ofrecen dispositivos basados en tecnología *antifuse* que presentan diversas ventajas respecto a los FPGA-SRAM. Se trata de dispositivos no volátiles que permiten una disponibilidad inmediata al encender, sin requerir un proceso de configuración inicial y eliminando con ello la necesidad de dispositivos adicionales. Asimismo, permiten un ahorro en el consumo de potencia del 20% en standby, con respecto a los dispositivos SRAM [Maxfield 04].

La tecnología *antifuse*[1] (antifusible) ofrece celdas que se comportan como circuito abierto cuando no están programadas, debido a la alta impedancia producida por un canal de silicio amorfo. A través de la aplicación de voltajes relativamente altos en las entradas del dispositivo es posible hacer crecer un canal entre dos placas metálicas, transformado al silicio amorfo (dieléctrico) en poli silicio (conductor) permitiendo así a la celda comportarse como circuito cerrado. La programación del dispositivo consiste en hacer crecer estos canales para realizar interconexiones entre los elementos internos del FPGA (Fig. 2.4H).A pesar de las bondades que presentan, poseen la grave desventaja de que solo es posible programarlos una vez, por lo que, al igual que los ASIC, no permiten realizar cambios en los diseños ni corregir errores cometidos. Por ello, no son una elección conveniente en ambientes de desarrollo o pruebas sino de producción a gran escala.

[1]El nombre *antifuse* (antifusible) surge de la idea de que el comportamiento de las celdas es opuesto al de un fusible, ya que al ser "programado" pasa de ser circuito abierto a circuito cerrado.

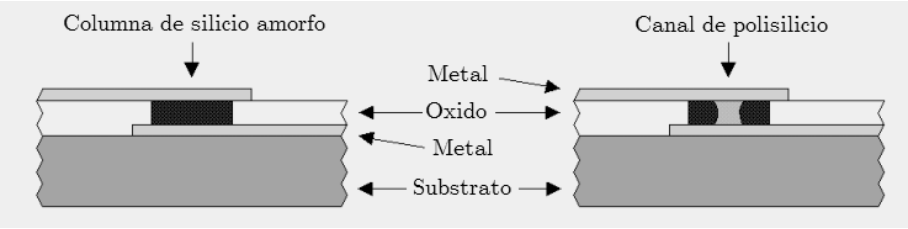

Fig. 2.4H Celda Antifuse [Maxfield 04].

Un enfoque prometedor son los dispositivos basados en memoria *FLASH* y *E^2PROM,* ya que son capaces de almacenar contenido no volátil y son reprogramables. Además, tienen celdas más pequeñas respecto a las SRAM, lo que se traduce en dispositivos más compactos.

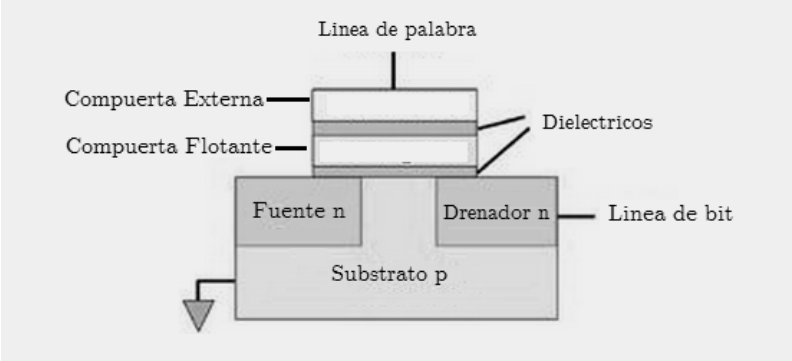

Fig. 2.4I Celda Flash.

En estas celdas, los datos se almacenan en la forma de carga eléctrica acumulada al interior de la compuerta flotante; por lo que es posible tomar ventaja de la naturaleza analógica de este dispositivo y almacenar diferentes niveles de voltaje para representar más de un bit por cada celda (Fig. 2.4I).

No obstante, poseen un consumo de potencia relativamente alto en standby debido a la gran cantidad de resistencias pull–up [Maxfield 04]. Esto puede explicar que hasta el día de hoy no son muchas las compañías que ofrecen dispositivos basados en esta tecnología.

2.4.4.3.2 Arquitecturas FPGA.

Las variantes en estos dispositivos no se presentan únicamente en la tecnología de las celdas de programación, es posible, además, encontrarlas en el tipo de elementos con los que están construidos los bloques lógicos básicos. Existen dos alternativas: multiplexores y por tablas de consulta LUTs (Look-up Tables).

Los bloques lógicos basados en multiplexores son capaces de implementar funciones lógicas empleando exclusivamente estos dispositivos. Resultan ventajosos para implementar funciones lógicas simples, debido a que el retardo de las señales es relativamente pequeño.

Por su parte, las arquitecturas basadas en tablas de consulta son líderes en lo que se refiere a operaciones aritméticas con números binarios, ya que contienen componentes adicionales diseñados para optimizar esta tarea. Los bloques básicos poseen algunas señales de entrada como índice de una tabla de consulta; el contenido es almacenado de manera que permite acceder al valor de una función lógica según el índice presente en sus entradas.

Una técnica común para implentar las funciones es almacenar bits en celdas RAM y usar las entradas como índice para acceder a cada celda SRAM usando varias compuertas de transmisión; éstas se comportan como circuito abierto o cerrado según el valor que reciba. Véase la figura 2.4J.

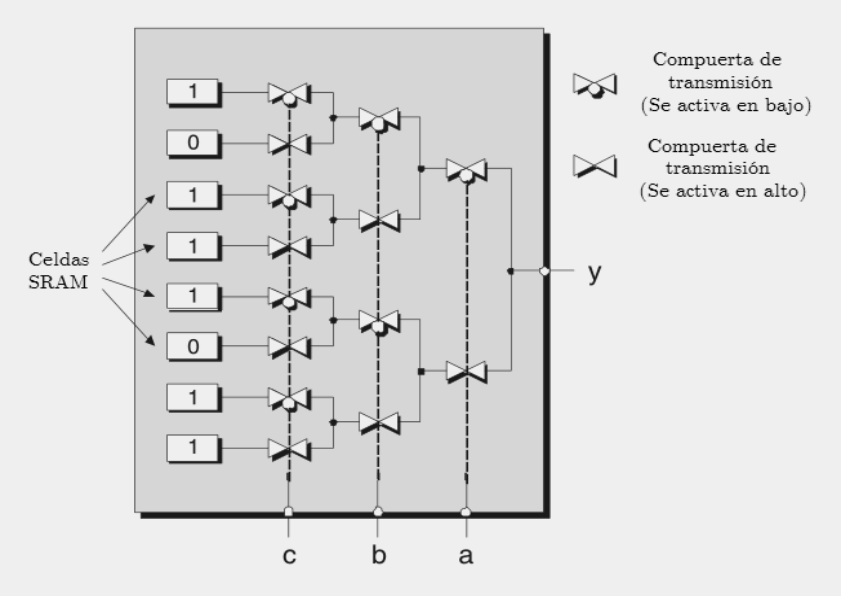

Fig. 2.4J Arquitectura LUT de compuertas de transmisión [Maxfield 04]

Los bloques LUT son muy eficientes, en términos de implementación y retardos entrada-salida, cuando se tienen funciones lógicas relativamente grandes. Sin embargo, si las operaciones son simples pueden resultar lentas respecto a bloques basados en multiplexores, además de que algunos de los elementos al interior del bloque se verán desperdiciados.

El uso de los FPGAs en telecomunicaciones, redes y otras áreas que requerían procesar datos numéricos, durante los años 90, impulsó un gran desarrollo de estas arquitecturas, por lo que la mayor parte de los FPGAs disponibles actualmente contienen LUTs en sus bloques lógicos básicos. Inicialmente presentaban tres entradas, pero gradualmente se fueron introduciendo diseños que contenían cuatro, cinco y hasta seis entradas por bloque lógico; en la actualidad, parece haber un consenso general en el balance desempeño-número de entradas, ya que la mayoría de los FPGAs contienen LUTs de cuatro entradas [Maxfield 04].

Las ventajas ofrecidas por las celdas basadas en tablas de consulta se deben a que cada bloque contiene elementos adicionales como multiplexores, compuertas y registros que potencializan su capacidad de procesamiento de datos numéricos.

2.4.4.3.3 Elementos para procesamiento digital de señales

Algunos de los elementos que los FPGAs contienen como hardware especializado para optimizar su desempeño en el procesamiento de señales son:

i) Cadenas de acarreo rápido. Cuando una operación aritmética (adición o multiplicación) se realiza, el tiempo necesario para ejecutarla se incrementa de manera proporcional al número de bits de los operandos, ya que el resultado para un bit intermedio depende del resultado de las operaciones realizadas con los bit anteriores (acarreo). Los FPGAS basados en LUTs proveen hardware adicional y las interconexiones adecuadas para calcular los acarreos sin necesidad de esperar resultados previos, permitiendo realizarlas en un tiempo constante que no depende de la longitud de los operandos.

ii) Memorias RAM embebidas, denominadas memorias de bloque. Se trata de pequeños módulos de almacenamiento que operan a gran velocidad y pueden ser encontrados en la periferia del FPGA o distribuidos a lo largo de su superficie con cierto grado de separación entre ellas. Las memorias pueden ser empleadas de manera independiente o en forma conjunta para almacenar datos (Fig. 2.4K).

iii) Multiplicadores o unidades MAC embebida*s.* La implementación de algunas funciones de procesamiento de señales son más lentas si se realizan conectando muchos bloques pequeños para este fin. Por ello, muchos FPGAS contienen multiplicadores o unidades de multiplicaióón y acumulación (MAC) diseñadas con hardware, las cuales pueden ser implementadas fácilmente y ofrecen un mejor desesmpeño.

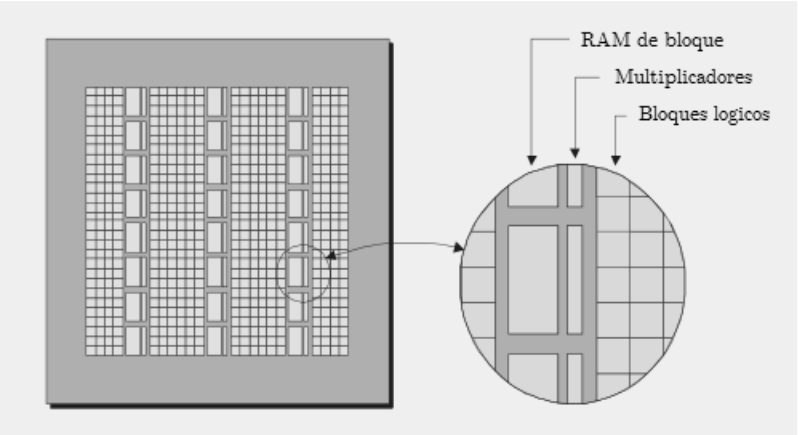

Fig. 2.4K Bloques embebidos en FPGAs [Maxfield 04].

iv) Microprocesadores embebidos. Dada la necesidad de emplear una gran cantidad de bloques lógicos básicos para realizar operaciones aritméticas complejas, algunos FPGAS contienen núcleos de procesadores en su interior, que permiten ayudar al procesamiento de datos numéricos y cálculos complejos, aunque pueden requerir una gran cantidad de bloques lógicos básicos.

2.4.4.3.4 Descripción del hardware

Para que un dispositivo lógico programable pueda realizar las tareas deseadas no puede emplearse un lenguaje convencional, puesto que no existe un set de instrucciones ni unidades que permitan la decodificación y la lectura de las mismas.. La herramienta que permite configurar las interconexiones para una gran cantidad de dispositivos de arquitectura configurable, tales como FPGAs, CPLDs y ASICs se denomina *lenguaje de descripción de hardware* (LDH).

Se trata de un conjunto de expresiones que permiten la descripción formal del comportamiento de los circuitos electrónicos digitales. Su principal característica es que permite expresar la tarea que el hardware realizará independientemente de la arquitectura en la que será implementado. Lo anterior permite describir su operación y posteriormente elegir el hardware que resulte más adecuado para la aplicación.

La principal diferencia estos lenguajes y uno de programación de software, es que brindan la posibilidad de expresar tiempo y paralelismo de manera explicita. Además, al generar un modelo de un circuito antes de ser construido, permite su análisis en un simulador. Lo anterior permite al diseñador conocer si la solución se comporta de acuerdo a lo deseado. Las instrucciones se procesan a través de un *compilador lógico;* con los datos que de él se obtienen es posible configurar el dispositivo.

En el compilador lógico, el LDH es sometido a un proceso conocido como síntesis, en el cual se adaptan las instrucciones a un tipo de hardware en concreto. Posteriormente se realiza un proceso de optimización con el objetivo de ajustar los requerimientos a la mayor velocidad o el menor espacio posibles.

Más tarde, durante el proceso de ubicación y enrutamiento, los bloques digitales se ubican de la mejor forma posible, procurando que aquellos bloques que operen de manera conjunta queden lo más cerca posible, minimizando los retardos de propagación de las señales. En este punto, el diseño puede ser simulado (simulación post routing) ofreciendo una idea clara de la forma en que la implementación va a comportarse en el dispositivo, considerando la forma en que los bloques están interconectados. Por último, el software genera las interconexiones para FPGAs o CPLDs o bien generara las máscaras para ASICs.

Al hacer uso de lenguajes de descripción de hardware, es posible representar por medio de instrucciones diferentes niveles de abstracción. El nivel más bajo es el estructural, con el cual un circuito lógico se describe a partir de conexiones entre interruptores o compuertas lógicas.

El siguiente nivel es el funcional, que permite expresar el comportamiento del circuito como funciones booleanas; en este caso no resulta de interés el tipo de dispositivos que se empleen siempre y cuando cumplan con la función lógica deseada. Aquí es donde se introduce el uso de símbolos condicionales. Finalmente, el nivel de abstracción más alto es el de comportamiento, que se refiere al hecho de poder hacer uso de constructores abstractos como ciclos de repetición y procesos; así como elementos algorítmicos, tales como adiciones o multiplicaciones para indicar al compilador lógico la tarea que deben realizar los circuitos a interconectar.

En la actualidad existen diversos lenguajes de descripción de hardware; sin embargo, los fabricantes de FGPAs, CLPDs y ASICs hacen uso normalmente de dos de ellos VERILOG y VHDL. Ambos ofrecen las características necesarias para la descripción de hardware y no existe un parámetro claro para determinar cuál de los dos es mejor, ya que ambos ofrecen capacidades similares.

Entre las características que presenta VERILOG se tienen: es relativamente más fácil de aprender, dadas sus similitudes con el lenguaje c; es un lenguaje interpretado, posee tipos de datos fijos. Sin embargo, posee poca capacidad de reutilización en los diseños.

Por su parte, VHDL, aunque en comparación con VERILOG pudiese ser más difícil de aprender, permite al usuario crear sus propios tipos de datos, es un lenguaje compilado, y permite replicar estructuras fácilmente.

La figura 2.4L presenta un diagrama del alcance de los lenguajes de descripción de hardware más populares, según el nivel de abstracción al que se desee trabajar.

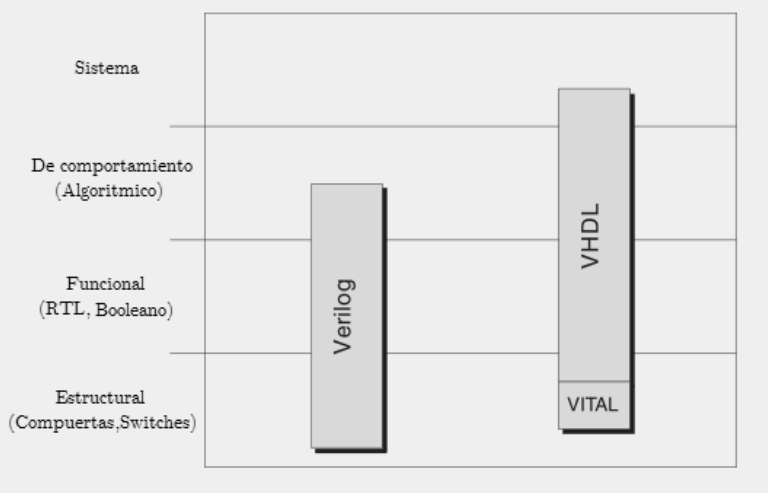

Fig. 2.4L Niveles de abstracción en LDH [Maxfield 04].

Este trabajo se orienta al uso del lenguaje VHDL, principalmente por que presenta: solidez en los niveles de abstracción funcional y de comportamiento, más versatilidad en la descripción de hardware, y capacidad de reutilización.

2.4.4.3.5 Limitantes

Se debe mencionar que si bien los FPGAs ofrecen soluciones útiles y flexibles, el mejor desempeño será siempre alcanzado por ASICs, ya que la versatilidad conduce a perdidas en rendimiento. Lo anterior obliga a que siempre se evalúe el costo del desarrollo con respecto al rendimiento deseado. Si el dispositivo desea producirse a gran escala, un desarrollo ASIC resultará más conveniente. Asimismo, los FPGAs presentan ciertas desventajas en lo que se refiere a la implementación de algoritmos muy complejos, o muy precisos, que requieran involucrar representaciones numéricas elaboradas. El gasto requerido en bloques lógicos puede ser muy grande y el resultado final podría no ser mucho mejor que el ofrecido por procesadores digitales de señales, quienes ofrecen soluciones más simples en aplicaciones de alta complejidad computacional. No obstante, los FPGAs continúan desarrollándose y evolucionando en busca de superar estos inconvenientes; la introducción de núcleos de procesadores en el FPGA es un claro ejemplo de ello.

Capítulo 3

Diseño del receptor digital.

3.1 Condiciones para el diseño del receptor.

Para la realización de este trabajo, no se cuenta con un CAD con las condiciones necesarias para esta aplicación, sino unicamente con una tarjeta de desarrollo con un FPGA; por lo cual, no conviene contruir la parte analogica del sistema. En lugar de ello, se hace uso de MATLAB para la generación de las señales que se espera recibir en la entrada del convertidor y se simula el proceso de CAD. Este proceso genera las muestras que son suministradas a la parte del sistema implementada en el FPGA, es decir a: los osciladores, los multiplicadores digitales y los filtros (Vease figura 3.1A).

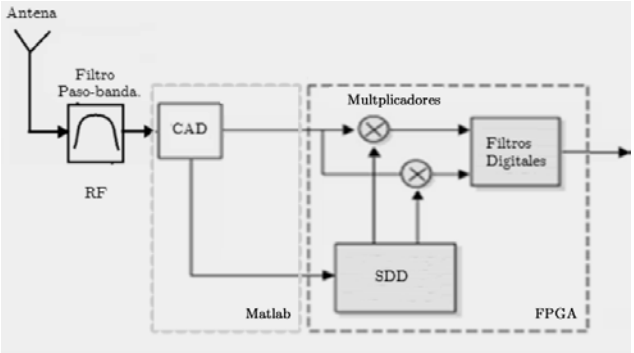

Fig 3.1A Metodo de implementación del receptor digital

3.1.1 Ventajas y limitantes asociadas al uso de la simulación.

El hecho de simular las señales en matlab, en lugar de implementar el proceso de generación, modulación, transmisión, recepción y amplificación analogicamente, simplifica en gran medida el proceso de diseño del receptor. Por una parte, permite centrarse únicamente en el diseño del hardware digital sin considerar las limitantes que puedan estar asociadas al hardware analogico.

Por otra, permite establecer una gran variedad de condiciones para que el receptor pueda ser probado y validado. Sin embargo, la simulación de las señales por computadora tambien impone condiciones sobre el sistema y puede limitar la velocidad de operación del mismo, debido a la necesidad de contar con una interfaz de comunicación FPGA-Computadora lo suficientemente rapida para permitir el envio de datos, y lo suficientemente simple para no complicar de manera significativa el diseño del receptor, ni emplear una cantidad excesiva de recursos del FPGA para dicho fin. Este hecho puede ser evaluado una vez que se conocen las caracteristicas y alcances del hardware disponible, el cual se presenta en la siguiente sección de este capitulo.

3.1.2 Hardware a disposición y sus alcances.

En lo que se refiere a la implementación en el FPGA, es importante mencionar que si bien el enfoque que suele seguirse al diseñar un sistema digital en hardware de arquitectura programable es generar un diseño que satisfaga un conjunto de especificaciones dadas, y posteriormente elegir el hardware que permita implementarlo con el menor costo posible, en este trabajo se sigue un camino diferente. Se parte de una tarjeta de desarrollo disponible y se analizan los alcances que presenta para concluir acerca de las caracteristicas que puede tener el sistema al operar en esa tarjeta. Se pretende con ello garantizar la implementación del sistema en el FPGA, y asi conocer de primera mano los problemas y limitantes asociados con la operación en el hardware. Este hecho permitirá comprender y explorar aquello que usualmente se obvia en implementación de este tipo de sistemas, buscando con ello abrir las puertas a trabajos futuros de procesamiento de señal en dispositivos FPGA, de tal manera que se pueda simplificar su trabajo y plantear alcances más ambiciosos.

3.1.3 Tarjeta de desarrollo Xilinx Spartan3E.

Los FPGA son circuitos integrados que suelen tener una gran cantidad de pines de entrada y salida, por lo cual los fabricantes emplean encapsulados de montaje superficial como el que se presenta en la figura 3.1B.

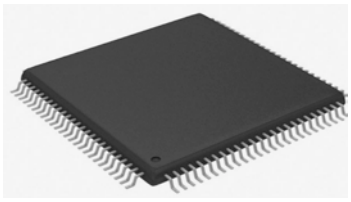

Fig. 3.1B Encapsulado FPGA

El montaje superficial obliga a realizar un circuito impreso e instalar en él los componentes necesarios para que el FPGA pueda operar. Para simplificar esta tarea los fabricantes ofrecen tarjetas de desarrollo que integran un FPGA listo para ser programado, junto con componentes analogicos y digitales para usarse de manera conjunta con el FPGA, permitendo explorar sus capacidades más facilmente, generar diseños elaborados y realizar pruebas de manera versátil y simple.

Para el desarrollo de este trabajo, se cuenta con una tarjeta de desarrollo de la marca Xilinx, modelo Spartan 3E presentada en la figura 3.1C.

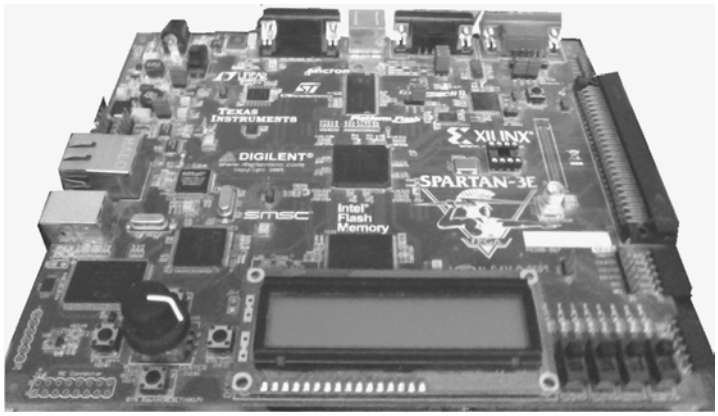

Fig. 3.1C Tarjeta de desarrollo Spartan 3E de Xilinx

Según los datos ofrecidos por el fabricante, la tarjeta de desarrollo integra, entre otros elementos:

- Un FPGA Xilinx XC3S500E, con encapsulado de 320 pines y hasta 232 pines disponibles para el usuario.
- Un CPLD de 64 Macroceldas XC2C64A.
- 64 Mbytes de memoria SDRAM externa.
- 16 Mbytes de memoria paralela flash(Intel StrataFlash).
- 2 Mbytes de memoria serie SPI flash(STMicro).
- Un puerto PS/2 para ratón o teclado.
- Un puerto VGA.
- Un Puerto Ethernet 10/100 PHY.
- 2 puertos RS232 estilos DTE y DCE("hembra" y "macho").
- Reloj oscilador a 50[Mhz]
- Puerto USB para programación.
- 3 Conectores Diligent de 6 pines.
- Un CDA de cuatro salidas SPI.
- Un CAD de dos entradas SPI.
- Un encoder rotatorio.
- 8 Leds.
- 4 botones push.
- 4 botones deslizables.
- Pantalla LCD de 2 lineas y 16 caracteres.

Como puede observarse de esta lista, la tarjeta cuenta con elementos de diversa indole: memorias externas para datos: pantalla LCD o puerto para monitores VGA, los cuales permiten mostrar resultados obtenidos al interior del FPGA; puertos para ratón, o un teclado, que permiten enviar datos al interior mismo; así como, puertos como el RS232 para el envio y recepción de información a dispositivos que cuenten con soporte para dicho fin. Destaca el hecho de que la tarjeta cuente con un puerto USB , ya que permite la programación del FPGA de manera muy simple, desde practicamente cualquier computadora.

Aunque el uso de estos componentes externos es atractivo, es importante hacer una evaluación antes de hacer uso de alguno de ellos, ya que el control de los mismos debe ser descrito con LDH y por lo que consume recursos lógicos al interior del FPGA, lo cual, además de la complejidad que añade al desarrollo de un sistema, puede conducir a un consumo importante de recursos del FPGA.

3.1.3.1 FPGA XCS500E

En lo que se refiere al FPGA, presente en la tarjeta de desarrollo, el cual es el XC3S500E de Xilinx, cuenta con 1,164 bloques logicos configurables (BLC). Se trata de un arreglo de 46 reglones por 34 columnas. Cuenta con 45Kbytes de memoria de bloque[1] (20 memorias de 2.25kbytes) y 20 multiplicadores de 18 bits.

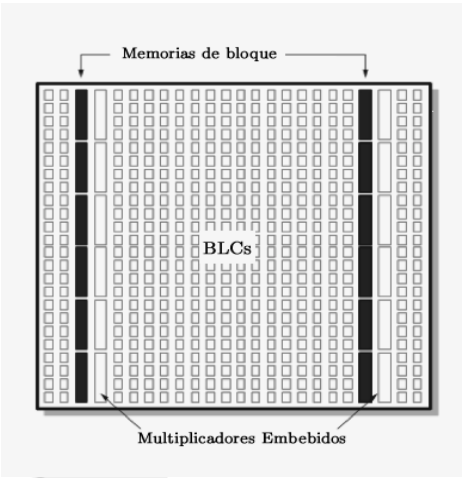

Fig. 3.1D Estructura interna del FPGA

Los BLCs son los bloques que conforman los FPGA de Xilinx. Cada BLC contiene 4 sub elementos a los que se denomina *slice.* Cada *slice* contiene 2 LUTs, 2 elementos de almacenamiento, asi como compuertas y multiplexores que permiten efectuar operaciones aritmeticas de manera eficiente.

De los 4 *slices* en cada CLB, un par son de tipo L (SLICEL) y los dos resantes de tipo M (SLICEM). El primer grupo permite implementar únicamente funciones logicas y aritmeticas, mientras que el segundo puede ser, además, una memoria[2] o un registro de corrimiento; ambos se ilustran en la figura 3.1E.

[1] Las memorias al interior del FPGA que no forman parte de un CLB se conocen como memorias de bloque.

[2] Cuando se emplean CLBs del FPGA para almacenamiento de datos se dice que se hace uso de memoria distribuida.

Fig. 3.1E Tipos de *slice* disponibles.

Los FPGAs de Xilinx cuentan, además de los elementos lógicos para aritmética, con conexiones para el mismo propósito; en conjunto permiten la implementación eficiente de sumas, restas y multiplicaciones. Si bien las operaciones aritmeticas pueden implementarse únicamente con LUTs, todas ellas requieren calcular varias salidas, lo cual conduciría al uso de una cantidad importante de LUTs para una operación simple si no se contase con los recursos lógicos para operaciones aritmeticas.

Para ilustrar esta idea, considérese el caso simple de una adición de dos bits. En dicho caso se requiere una compuerta XOR, para generar la suma y una compuerta AND, para generar el acarreo (operación se conoce como medio sumador, ya que no considera acarreo de entrada). Con dos medios sumadores se puede construir un sumador completo, conectando la salida del primero de ellos y el acarreo de entrada en las entradas del segundo medio sumador, y generando el acarreo de salida con una compuerta OR que tenga como entradas los dos acarreos de salida de los medios sumadores, tal y como se muestra en la figura 3.1F.

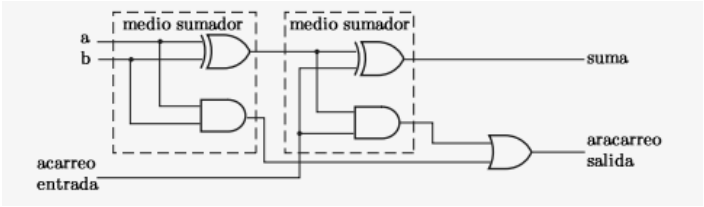

Fig. 3.1F Sumador completo.

Para esta operación, se requieren de al menos dos tablas LUT de 4 entradas, por cada bit que presenta la palabra a sumar. Este tipo de implementación hace que el segundo medio sumador tenga que esperar a que el primer medio sumador calcule su salida; además el acarreo de salida no podra ser ejecutado hasta que el segundo medio sumador ofrezca un valor de salida. El uso del metodo carry-look ahead , presentado en la figura 3.1G, para efectuar este cálculo permite una mejor implementación, reduciendo el retardo y la cantidad de LUTs.

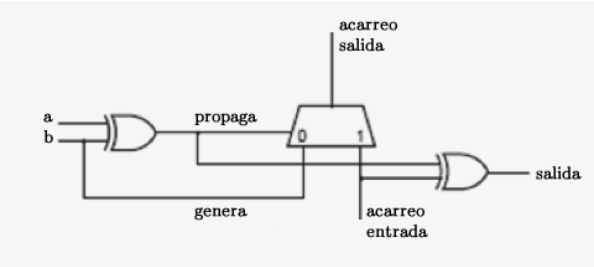

Fig. 3.1G Sumador Carry-look ahead.

Entonces, para obtener el valor de la suma, se efectuan dos medias sumas mediante compuertas XOR, la primera de ellas con los valores a sumar, y la segunda con el resultado de la media suma y el acarreo de entrada.

El acarreo de salida o se *propaga* o se *genera*, En el primer caso se alude a que el valor del acarreo de salida será igual al valor del acarreo de entrada, el cual ocurre cuando ambas entradas, a, b, presentan valores distintos. El segundo caso, en donde el acarreo se genera, ocurre cuando ambas entradas son del mismo valor; en tal caso, el acarreo tendrá el mismo valor que las entradas, por lo que se deberá reflejar este valor en el acarreo de salida. El manejo de este proceso se hace a través de un multiplexor cuya linea de selección es la compuerta XOR que produce la media suma, misma que, además, permite saber si ambas entradas son iguales o no.

Se trata de 3 funciones lógicas que no pueden ser combinadas en una sola LUT, en cambio Xilinx integra dos de ellas (el multiplexor y la segunda compuerta XOR de la figura 3.1G) al interior de cada slice para permitir emplear una sola LUT (medio slice) por cada bit de los sumandos. Asimismo, los multiplexores estan conectados entre si, para proveer una respuesta más veloz, en lo que se conoce como cadena de acarreo.

Además, cada slice provee una compuerta AND adicional, que permite realizar multiplicaciones eficientemente con ayuda de las cadenas de acarreo. El XC3S500E de Xilinx cuenta con 52 cadenas de acarreo, lo que le permite manejar operaciones de hasta 184 bits por columna.

3.1.3.2 Limitantes que impone la tarjeta sobre el diseño.

Para la implementación del receptor, de acuerdo a lo que se ha presentado con anterioridad, se requiere de:

- Memoria para almacenar los valores de las señales seno y coseno de los osciladores y logica de proposito general para el control de los mismos.
- Dos multiplicadores signados para el demodulador I/Q.
- Hardware para implementar, adiciones, multiplicaciones y retardos requeridos para las etapas de filtrado.
- Memoria para almacenar los coeficientes de los filtros.
- Elementos lógicos de propoósito general para la elaboración de los circuitos de control y sincronización de todos los elementos internos.

- Componentes logicos al interior del FPGA y uno de los puertos a disposición para realizar la interfaz que permita la comunicación con la computadora.
- Botones y leds presentes en la tarjeta para realizar pruebas y depurar el sistema.
- Puerto USB para cargar el hardware descrito en el FPGA.

Considerando las necesidades y los recursos proporcionados por la tarjeta, se pueden enlistar ciertas limitantes que pueden aparecer en el diseño del sistema. La primera es en relación al numero de multiplicadores disponibles, ya que se cuenta con 20 bloques, de los cuales 2 estan destinados al demodulador I/Q. Lo cual, en caso de que se deseen emplear los recursos de propositito general para las demas partes del sistema, dejaría unicamente 18 multiplicadores (9 para procesar la componente en fase y 9 para la componente en cuadratura), para la elaboración de una etapa de filtrado, por lo cual el número de coeficientes de los filtros debe ser reducido.

En segundo lugar, es fundamental, construir una interfaz para la comunicación con la computadora. De los puertos disponibles, el RJ45, a pesar de su gran velocidad, requiere de una cantidad importante de recursos para implementarse; por su parte, el puerto RS232 no solo puede utilizarse fácilmente y con poco hardware, sino que además, permite establecer fácilmente una comunicación con el entorno de trabajo MATLAB, para el envio y recepción de las muestras, por lo que se opta por dicho puerto para la realización de este trabajo. Desafortunadamente, existe una limitante importante con esta elección: la velocidad a la que pueden enviarse y recibirse datos es muy limitada, tal como como se puede ver en la tabla 3T1.

El FPGA disponible cuenta con una señal de reloj de 50[Mhz], por lo cual podría recibir información proveniente de la computadora a una tasa de hasta una muestra cada ciclo de reloj como máximo, es decir, 50,000,000 [muestras/segundo]. Lamentablemente, el envio de información a través del puerto RS232, en el mejor de los casos, se limita al envio de 96,216 muestras cada segundo, lo cual significaría realizar un procesamiento a una velocidad demasiado baja.

Si se tratase de procesamiento en tiempo real, podrian procesarse señales que puedan muestrearse hasta los 96.2[Khz], lo cual resulta demasiado bajo para lo que se pretende en este trabajo.

bits/s	muestras/s[1]
2,400	240
4,800	480
9,600	960
14,400	1,400
19,200	1,920
28,800	2,880
38,400	3,840
57,600	5,760
115,200	11,520
230,400	23,040
460,800	46,080
921,600	96,216

Tabla 3T1 Velocidades de comunicaciòn RS232[1]

Ante esta problemática, y con el objetivo de analizar las capacidades máximas del FPGA se plantea lo siguiente: enviar información a la velocidad que el puerto serie lo permite, no necesariamente a la mas elevada, almacenar una cantidad significativa de muestras en la memoria disponible y posteriormente extraer las muestras de la memorias a la maxima velocidad a la que puedan ser procesadas. Al concuir el procesamiento, se pueden almacenar también los resultados en la memoria disponible y finalmente enviarlos hacia la computadora, nuevamente a baja velocidad, para ser analizados en MATLAB.

Esta solución resulta viable, puesto que permite la implementación del sistema haciendo uso de una interface de comunicación sencilla con el FPGA, sin invertir gran cantidad de recursos lógicos en ello, permite observar la operación de los componentes a velocidades elevadas, donde pudiesen aparecer errores por los retardos en los componentes, y visualizar su comportamiento en los limites de operación del dispositivo.

[1] considerando palabras de 8 bits, un bit de paro y un bit de inicio

Al mismo tiempo, este diseño permitiría hacer uso de otra interfaz de comunicación más veloz, en algun otro dispositivo, eliminar la interfaz serie y hacer uso del hardware diseñado para operar a velocidades elevadas.

3.1.4 Núcleos de propiedad intelectual.

Ademas de las tarjetas de desarrollo, los fabricantes de FGPAs suelen brindar herramientas que permiten la generación automática de código para cumplir con alguna función requerida por el diseñador, sin la necesidad de conocer la forma en que el hardware debe describirse lo que permite acelerar el proceso de desarrollo de un sistema robusto y complejo.

Algunos de estos núcleos son gratuitos, sobre todo aquellos que ejecutan funciones simples y suelen ser usados en ambitos académicos. Mientras que aquellos que ofrecen soluciones más complejas, pensadas por ejemplo para aplicaciones industriales, requieren la adquisición de una licencia para su uso.

Los núcleos de propiedad intelectual permiten realizar de manera automatizada, entre muchas otras funciones: comparadores, acumuladores, contadores, registros de corrimiento, interfaces de comunicación con los puertos, multiplicadores, manejo de memorias, osciladores e incluso operaciones como diseño de filtros CIC , filtros FIR y transformadas.

A pesar de sus ventajas, los nucleos de propiedad intelectual pueden también restringir un diseño, ya que en ocasiones la generación automática dificulta en gran medida el control que se tiene sobre la implementación, puesto que el diseñador puede encontrarse con LDH generado automáticamente que no cumple completamente con la tarea deseada o bien, que resulte muy dificil de interpretar por el diseñador lo cual minímiza la posibilidad de ser modificado. Así pues, el uso exclusivo de núcleos de propiedad intelectual para desarrollar el sistema no es conveniente debido a la perdida en la fllexibilidad que puede estar presente.

Considerando lo anterior, puede pensarse en hacer uso de ellos sólo en algunas partes del sistema, como en el caso de los osciladores digitales que ofrecen una complejidad importante, ya que por si mismos han sido tema de desarrollo de trabajos de tesis, por lo cual los núcleos de propiedad pueden resultar convenientes si se considera que sus parámetros y características no varian tanto de un sistema a otro como para requerir un conocimiento profundo sobre su implementación. Se opta pues, por hacer uso del nucleo disponible (DSS compiler) para dicho fin, cuya creación y configuración es simple y se explica posteriormente.

Como ultimo aspecto, relacionado con las condiciones para diseño, resta realizar un análisis del rendimiento que tienen las operaciónes aritméticas al interior del FPGA y así, evaluar las capacidades de procesamiento del mismo.

3.1.5 Análisis del rendimiento.

Para poder evaluar el rendimiento del FPGA es necesario conocer la rapidez con la que se pueden realizar operaciones aritméticas, ya que este parámetro determinará la cantidad de muestras que pueden ser insertadas al FPGA por cada ciclo de reloj.

Como se mencionó previamente, una de las ventajas de la descripción de hardware es la posibilidad de simular en la computadora el comportamiento exacto que tendrá cuando sea ejecutado en el FPGA. Haciendo uso de esta posibilidad, se describió hardware para efectuar el producto de dos valores binarios y al realizar la simulación *post-routing,* en ISIM de Xilinx, se obtuvieron los tiempos necesarios para efectuar dicha operación según la longitud de palabra. Estos se muestran a continuación:

bits	Tiempo[ns]
8	9.44
16	10.961

Tabla 3T2 Tiempo de ejecución para la multiplicación.

La figura 3.1H Muestra la ejecución de la multiplicación signada de 16 bits, siendo a y b los factores y p el producto.

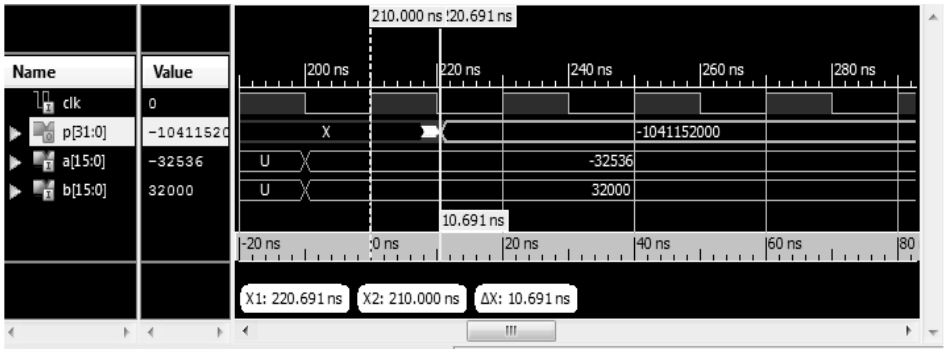

Fig. 3.1H tiempo de ejecución de una multiplicación de 16 bits[1]

Observese, de la tabla 3T2, que ambos resultados son favorables, ya que el dispositivo cuenta con un oscilador de 50[Mhz], que corresponde a un periodo de 20[ns]; asi que, ambas operaciones estarán listas en menos de un ciclo de reloj lo cual permite multiplicar números de estas longitudes a máxima velocidad del dispositivo.

Un proceso similar, para la adición de números, brindó los siguientes resultados:

bits	Tiempo[ns]
8	6.18
16	6.18
32	6.18

Tabla 3T3 tiempo de ejecución para la adición.

[1] se considera el tiempo de ejecución a partir del flanco de subida del reloj cuando se realiza la descripción de hardware : `if(clk'event and clk='1')then p<=a*b end if;`

La figura 3.3I muestra el caso de la adición de dos números de 16 bits donde x, y son los sumandos y s la suma:

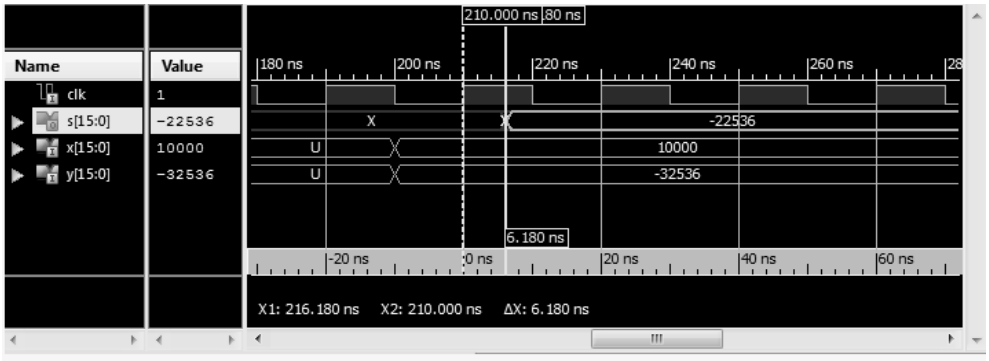

Fig. 3.1I Tiempo de ejecución, adición de 16 bits[1]

Con estos resultados puede observarse que tal y como lo indica el fabricante, el incremento en la longitud de la palabra de una adición no impacta la velocidad a la que esta se ejecuta. Nuevamente, el tiempo de ejecución permite desarrollar sumas en un sólo ciclo de reloj(20[ns]), por lo que pueden realizarse a máxima velocidad del dispositivo.

Por último, una capacidad que puede resultar interesante de implementar en el FPGA, dada la flexibilidad que permite para el diseño, es realizar sumadores de más de dos operandos, puesto que algunas estructuras de filtros pueden beneficarse de ello. Se realizarón pruebas con diferentes operandos de 16 bits y se obtuvieron los siguientes resultados:

Operandos (16 bits)	Tiempo[ns]
3	6.18
4	6.18
6	26.18
8	26.18

Tabla 3T4 tiempo de ejecución , adición con varios sumandos.

[1] se considera el tiempo de ejecución a partir del flanco de subida del reloj cuando se realiza la descripción de hardware : `if(clk'event and clk='1')then s<=x+y end if;`

La siguiente figura muestra el caso de 4 sumandos (a,b,c,d).

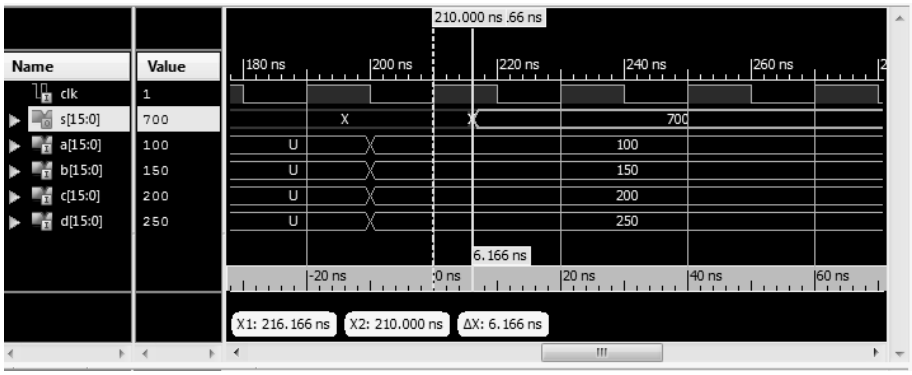

Fig. 3.1J tiempo de ejecución, adición de 4 sumandos.

Esta capacidad del FPGA brinda una flexibilidad aún mayor, ya que, en casos donde asi se requiera, pueden realizarse sumas de 4 operandos simultaneamente en un ciclo de reloj o hasta de 8 en dos ciclos. La cantidad de LUTs que se requieren para este tipo de operaciones es de un poco más del doble, pero ofrece a cambio una mejora significativa en el desempeño. La cantidad de entradas por cada slice determina el límite en el cual se requiere un ciclo más, ya que a partir de 5 operandos, la suma requiere dos ciclos de reloj.

Una vez conocidas las condiciones para el diseño, es posible comenzar a realizarlo; sin embargo, antes de entrar de lleno a dicho proceso, es importante mencionar que aunque los bloques que conforman el sistema fueron concebidos para realizar muestreo en FI, es posible adaptarlos para operar haciendo uso de muestreo paso-banda. No obstante, esto conduce a perdida de generalidad, ya que casi todos los parámetros del sistema dependen de la señal de trabajo y no siempre es posible modificarlos, como ocurre en el caso de la frecuencia de muestreo del CAD. Por ello, para comprender los alcances de cualquier diseño realizado y para permitir al diseñador hacer ajustes según la aplicación, se hace un análisis de las características de cada bloque que permite plantear criterios para elegir para sus parámetros.

3.2| Diseño del receptor.

Conocidas las limitantes asociadas al hardware con el que se cuenta, puede discutirse el proceso de diseño y así poder plantear un esquema lo mas general posible de tal manera que su funcionamiento pueda ser observado en la tarjeta de desarrollo Xilinx spartan 3E.

De acuerdo a lo presentado previamente, para el diseño del receptor, será necesario:

*Elegir los parametros de operación para el convertidor analogico digital, por ejemplo: velocidad de muestreo, tipo de cuantización y formato numerico[1] para representar las muestras que se suministrarán al FPGA.

*Generar una interfaz de comunicación del FPGA con la computadora, tanto emisión como para recepción.

*De acuerdo a la longitud de palabra elegida, construir la memoria para almacenar los datos provenientes del puerto serie. El tamaño de la memoria dependerá de la cantidad de muestras que se deseen almacenar.

*El uso y la configuración del núcleo de propiedad intelectual *DSS compiler* para la generación de las señales seno y coseno necesarias para el demodulador I/Q; asi como, elegir la cantidad de bits con la que se generarán estas señales y la representación numérica que se empleará para dicho fin, la cual deberá ser consistente con las muestras que provienen del puerto serie.

[1]Debido a la interfaz de comunicación FPGA-Computadora que ha sido elegida, la RS232 , la cual envia los bits en grupos de 8, conviene elegir longitudes de palabra que sean múltiplo de este número.

*Un circuito de control que permita coordinar la recepción de las muestras por parte de la interfaz RS232, su escritura en la memoria y, enseguida, la extracción de los datos de la misma. Asimismo, debe iniciar la generación de muestras por parte del SDD oportunamente, de tal manera que esté sincronizada con la extracción de las muestras de la memoria. Posteriormente debe activar los multiplicadores teniendo sus operandos listos e iniciar el proceso de filtrado justo en el momento en que la multiplicación haya terminado. Este aspecto es muy importante, ya que si no logran coordinarse correctamente todos los aspectos mencionados, no se obtendran resultados válidos.

*El diseño de las etapas de filtrado que trabajan con la señal en banda base y que permiten eliminar el ruido y reducir la cantidad de muestras al final del proceso. En este punto debe elegirse el tipo de filtros en cada etapa, el factor de diezmado y la estructura de filtrado para la implementación del los mismos, considerando que tanto las sumas como las multiplicaciones pueden realizarse en un ciclo de reloj y que existe una limitante importante en el número de multiplicadores disponibles: 9 para cada canal, si se descartan los dos que son necesarios para el demodulador I/Q.

*Un segundo circuito de control que permita insertar las muestras procesadas en memoria, en la medida en que se vayan generando, para que una vez que ésta cuente con todas las muestras esperadas, se coordine el envio de las mismas a través del puerto serie hacia la computadora.

*Finalmente, se deben construir los programas en matlab que permitan la comunicación con el puerto serie, enviar los datos en la representación deseada e interpretar los datos recibidos para ser mostrados.

Como puede verse, la cantidad de parametros a elegir es grande, por lo que una justificación detallada de todos ellos seria demasiado eleaborada y de poca utilidad. De entre todos los parametros mencionados, se eligen los factores que se consideraron importantes para su discusión por el impacto que tienen en el desempeño del sistema, mientras que para algunos otros se menciona únicamente alguna razón por la cual han sido elegidos, sin afirmar con ello que dicha elección sea óptima.

3.2.1 Elección de parametros del CAD en muestreo pasobanda.

3.2.1.1 Frecuencia de muestro.

Además de todas las ventajas presentadas previamente sobre muestreo paso-banda, existen tambien restricciones importantes asociadas con su uso, de entre ellas destaca que los intervalos validos para la frecuencia de muestreo dependen de la frecuencia máxima y mínima presentes en la señal de trabajo, por lo que la elección de una frecuencia de muestreo limita de manera importante las señales con las que puede trabajar un receptor que haga uso de este tipo de muestreo, o bien, obliga a incorporar componentes adicionales para ajustar la frecuencia de muestreo según la señal de interés. Lamentablemente, esto puede resultar muy complejo, ya que como se verá a continuación, la elección de la frecuencia de muestreo no es una tarea sencilla, puesto que existen multiples factores involucrados. Lo más adecuado puede ser elegir un grupo de señales, estudiar sus caracteristicas y generar parámetros para procesar cada una de ellas.

Para ilustrar los aspectos prácticos del muestreo paso banda-banda, considere una señal que exhiba una frecuencia central de 20 [Mhz] y un ancho de banda B de 200[Khz]. De acuerdo al teorema de Nyquist requiere ser muesttreada a una frecuencia F_m superior a los 40[Mhz]. En contraste, el teorema para muestreo de señales paso-banda, permite realizar dicho muestreo en alguno de los intervalos que se muestran en la tabla 3T5[1]:

[1] En el anexo A se ofrece codigo de matlab (fm_pb.m) que permite obtener estos intervalos fácilmente.

k	Fmax[Hz]	Fmin[Hz]	diferencia[Hz]
2	$3.98x10^7$	$2.01x10^7$	$1.97x10^7$
3	$1.99x10^7$	$1.34x10^7$	$6.5x10^6$
7	$6.6333x10^6$	$5.7429x10^6$	$8.9048x10^5$
8	$5.6857x10^6$	$5.025x10^6$	$6.6071x10^5$
10	$4.4222x10^6$	$4.02x10^6$	$4.0222x10^5$
50	$8.1224x10^5$	$8.04x10^5$	8244.9
70	$5.7681x10^5$	$5.7429x10^5$	2525.9
100	$4.0202x10^5$	$4.02x10^5$	20.202

Tabla 3.T5 Algunas Frecuencias de muestreo validas para muestreo pasobanda. F_c=20[Mhz] B=200[Khz].

En este caso, la frecuencia de muestreo se puede reducir hasta una velocidad que ronda los 400[Khz], casi 100 veces menos que lo que impone el teorema de Nyquist. Sin embargo, aparecen dos desventajas con esta elección. La primera es que el ruido se incrementa por un factor k, que para un muestreo a una velocidad en el ultimo intervalo de la tabla 3T5 exhibiría 100 veces más ruido que un muestreo superior a los 40[Mhz]. En segundo lugar, la variación en la velocidad de muestreo (jittering del CAD) es un factor importante a considerar en casos donde el valor de k sea elevado, ya que el intervalo válido para muestreo se vuelve pequeño y una variación mínima de la velocidad de muestreo puede desencadenar solapamiento espectral y, por ende, perdida de información.

Con respecto al primer problema asociado al muestreo paso-banda, es util comparar la diminución efectiva en la frecuencia de muestreo y el incremento del ruido. La figura (3.2A) muestra el comportamiento tipico de la relación entre estos dos factores[1]; nótese que para valores de k grandes, la reducción en la frecuencia de muestreo generada con incrementos sucesivos de k, es poco significativa con respecto a la cantidad de ruido que se produce, por lo cual , en este sentido, un factor k elevado resulta poco conveniente.

[1] En el anexo A se ofrece codigo de matlab (fm_rdo.m) que permite obtener esta grafica y evaluar hasta donde conviene reducir la velocidad de muestreo.

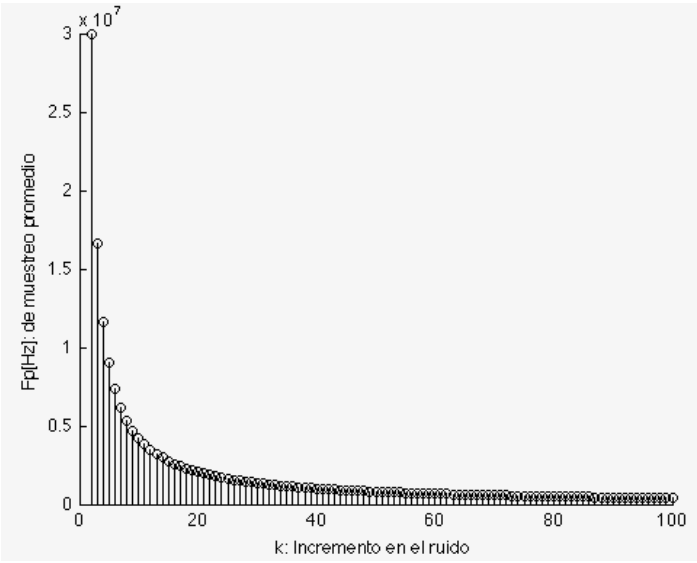

Fig 3.2A Relación entre la frecuencia de muestreo promedio y el ruido generado.

En sintesis, la elección de la frecuencia de muestreo depende de: las frecuencias máxima y mínima presentes en la señal, la variación que se pueda presentar en la velocidad de muestreo del CAD y de la cantidad de ruido que convenga incorporar al proceso respecto a la reducción obtenida en la frecuencia de muestreo, es fundamenteal que el ruido agregado por muestreo paso-banda pueda ser eliminado con el hardware digital disponible.

Aunque la bibliografia, [Proakis 06] [Vaughan 91], presenta un análisis detallado sobre la obtención de los intervalos de muestreo validos, no ofrecen información sobre la frecuencia en la que quedarán situadas las bandas de interés. Conocer esta ubicación es fundamental en receptores digitales, ya que dicha posición determina la frecuencia a la que deben operar los osciladores, de manera que permitan desplazar el espectro a banda base. A continuación se realiza un análisis que permite encontrar la ubicación exacta de la banda de interes indpendientemente de la señal de trabajo.

En primer lugar, se ha visto que al realizar muestreo de una señal a una frecuencia F_m, se producirán replicas del espectro de la señal original centradas en nF_m; $n = \pm 0,1,2,3...$etc. Por lo regular se trabaja con señales reales, cuyo espectro en magnitud es simétrico respecto a F=0. Por lo tanto, si la señal presenta una banda de interés centrada en $F = F_c$, debera exhibir también una banda igual en $F = -F_c$. Con el muestreo se consigue que las copias de la banda de interés que se ubicaban en frecuencias negativas queden centradas en $nF_m - F_c$ y las de frecuencias positivas en $nF_m + F_c$, en ambos casos para $n = \pm 0,1,2,3$ ...etc.

El espectro en magnitud de las señales discretas es periodico con periodo T=$1/F_m$, por ello, cualquier intervalo del espectro de tamaño F_m contiene la misma información. Por simplicidad conviene trabajar con el intervalo $[-\frac{F_m}{2}, \frac{F_m}{2}]$, además, dada la simetria que exhibe el espectro de magnitud en señales reales , basta considerar únicamente las frecuecuencias ubicadas en la región $[0, \frac{F_m}{2}]$ para conocer lo que ocurre en cualquier parte del espectro. Por lo tanto, interesa conocer la frecuencia F_i, en dicho intervalo, en la cual quedará centrada la banda de interés del espectro. Según la elección del factor k para muestreo paso-banda, se presentará uno de los siguientes casos (Vease fig. 3.2B):

i) Que una replica del espectro centrada en una frecuencia negativa coloque la sección positiva del espectro en el intervalo $[0, \frac{F_m}{2}]$. En este caso, la banda de interés quedará centrada en $F_i = -mF_m + F_c$, para algún m, entero y mayor que cero.

ii) Que una replica del espectro centrada en una frecuencia positiva coloque la sección negativa del espectro en el intervalo $[0, \frac{F_m}{2}]$. La banda de interes tendra como centro a: $F_i = mF_m - F_c$, donde m es entero y mayor que cero.

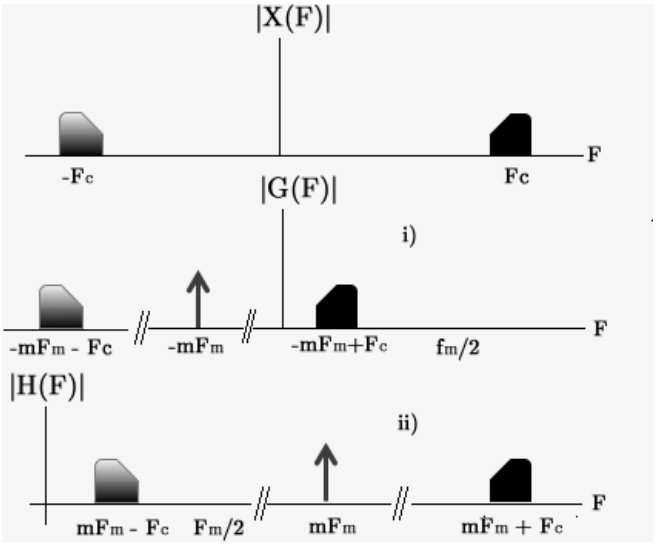

Fig 3.2B Frecuencia de interes

En la bibliografía [Lyons 10] se presenta que si en la ecuación (2.1.5) k toma valores pares, entonces existe una inversión en el espectro. Esto mismo ocurre en el caso ii) donde la banda de interés, situada en el intervalo $[0, \frac{F_m}{2}]$, corresponde a frecuencias negativas. Por tanto, al seleccionar un factor k par, para muestreo paso-banda, la frecuencia F_i estara dada como:

$$F_i = mF_m - F_c \tag{3.2.1}$$

mientras que la elección de un factor k impar, produce una frecuencia F_i igual a:

$$F_i = -mF_m + F_c \tag{3.2.2}$$

Ahora bien, para que la banda de interés se encuentre en en el intervalo de trabajo: $0 < F_i < \frac{F_m}{2}$.

De (3.2.1) y (3.2.2) se obtiene que:

$$0 < -mF_m + F_c < \frac{F_m}{2};\ k\ impar \tag{3.2.3}$$

$$0 < mF_m - F_c < \frac{F_m}{2};\ k\ par \tag{3.2.4}$$

De donde:

$$\frac{F_c}{F_m} - \frac{1}{2} < m < \frac{F_c}{F_m}\ ; k\ impar \tag{3.2.5}$$

$$\frac{F_c}{F_m} < m < \frac{F_c}{F_m} + \frac{1}{2}; k\ par \tag{3.2.6}$$

Puesto que m identifica a la replica que situará la banda de interés en el intervalo $[0, \frac{F_m}{2}]$, dicho valor debe ser necesariamente un valor entero. Si se observan las desigualdades (3.2.5) y (3.2.6), se encuentra que los límites superior e inferior solo varian en 0.5. por lo cual, puede existir como máximo un valor entero que las satisfaga.

Para resolver (3.2.5) y (3.2.6) es necesario elegir una frecuencia de muestreo. Pero para cada valor de k en la ecuación (2.1.5), existe un intervalo de frecuencias de muestreo válidas, por lo cual es necesario tomar algún criterio que permita esta elección. Con dicho objetivo, la figura 3.2C presenta el efecto de aumentar o disminuir la frecuencia de muestreo: al variar la frecuencia en ambos sentidos lo suficiente las replicas espectrales, se solapan.

Por lo tanto, no es posible que la copia de la banda de interes que se encuentra en el intervalo intervalo $[0, \frac{F_m}{2}]$, pueda abandonarlo, como consecuencia de cambiar la frecuencia de muestreo dentro de un mismo intervalo válido, sin que se produzca solapamiento. Luego entonces, puede deducirse que el valor de m, que identifica a la replica de la banda de interés ubicada en $[0, \frac{F_m}{2}]$, es la misma para todos los valores de F_m asociados a un mismo valor de k en (2.1.5).

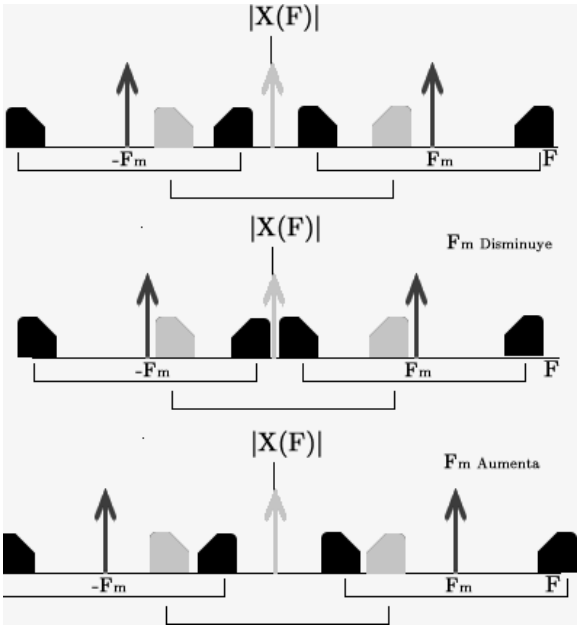

Fig 3.2C Variación de Fm .

De acuerdo con lo anterior, sin importar la frecuencia de muestreo que se elija para resolver las desigualdades (3.2.5) o (3.2.16), se obtendra el mismo valor de m siempre y cuando la frecuencia de muestreo no abandone el intervalo valido.

Conocer el valor de m que satisface la desigualdad correspondiente, para cada valor de k, permite conocer la posición F_i que ocupará la banda de interés, a medida que se modifique la frecuencia de muestreo F_m, lo que permite situarla en donde resulte mas conveniente[1].

[1] En el anexo A se ofrece codigo de matlab (mpb_posv.m) que permite obtener las graficas de Fi en función de Fm para uno o mas valores de k (ver fig 3.1D).

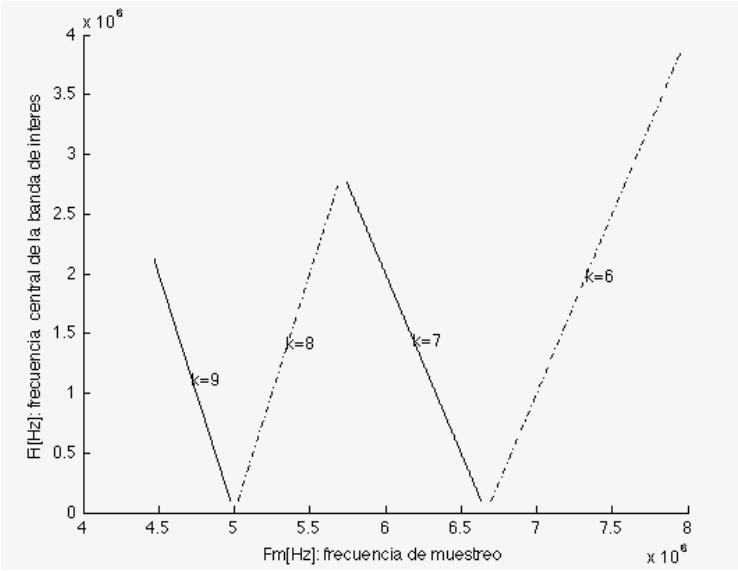

Fig 3.2D Ubicación de la banda de interes Fi.

La figura (3.2D) presenta las gráficas para algunos valores de k al muestrear la señal considerada previamente, Fc=20[Mhz] y B=200[Khz], con muestreo paso-banda. Observe que para valores de k pares, un aumento en la frecuencia de muestreo situa la banda de interés a una frecuencia mayor, mientras que para valores impares de k, un aumento en la frecuencia de muestreo ubica a la banda de interes a una frecuencia mas baja. Además, note como una misma ubicación para F_i puede ser conseguida por multiples valores de k. En la figura (3.2D), cualquier valor en el intervalo (0.25, 2) [Mhz] se puede conseguir con valores de k=6,7,8 y 9.

Sin embargo, aunque varios valores de k situen la banda de interés en una misma frecuencia, Fi, el diseño de los filtros y el analisis del impacto del diezmado, se realizan en el el domino discreto, en el cual el ancho de banda b de la señal, estará dado como:

$$b = \frac{2B}{F_m} \tag{3.2.7}$$

Así pues, frecuencias de muestreo elevadas producen anchos de banda reducidos en el dominio discreto, mientras que frecuencias de muestreo pequeñas producen anchos de banda mayores. Por ende, para una misma frecuencia Fi y diferentes velocidades de muestreo, el ancho de banda en el dominio discreto puede ser un factor que determine la elección de k, ya que anchos de banda reducidos, requerirán de filtros digitales más elaborados y de menor rendimiento. Asímismo, no debe olvidarse que uno de los bloques que se plantean para constituir el receptor es de reducción de la tasa de muestreo, por lo que debe garantizarse que el ancho de banda de la información, sea menor al ancho de banda que se presentará después de la reducción de la tasa de muestreo, de lo contrario tambien existirá solapamiento espectral y pérdida de información.

3.2.1.2 Cuantización y formato numérico.

Como se ha presentado previamente, la cuantización uniforme es la más simple ,y la que cuenta con mayor aceptación . Además, puede implementarse fácilmente, y el ruido por cuantización se puede calcular de manera simple, por lo tanto es el que se considera en este trabajo de tesis. En este tipo de cuantización, el numero de niveles, es de 2^N donde N es el número de bits empleados para representar los numeros. Como se ha visto previamente, la interrfaz RS232 sugiere el uso de multiplos de 8 bits para representar las señal. Para esta elección se debe considerar que al efectuar una multipliación de dos palabras binarias el producto duplica su longitud y que al interior del dispositivo existen 2 etapas de multiplicación de las muestras que ingresan al dispositivo, la primera en la demodulación y la segunda en el proceso de filtrado. Por lo tanto, si se eligen 16 bits de inicio, la etapa de filtrado debera multiplicar palabras binarias de 32 bits lo cual es superior a la capacidad de los multiplicadores embebidos de 18 bits, por lo que sería necesario describir el hardware para hacier uso de logica de proposito general, por simplicidad se opta por elegir 8 bits, lo que permite usar los multiplicadores embebidos, lo que conduce a 2^8 niveles de cuantización.

En lo que se refiere al formato numérico, es necesario contar con una representación que permita trabajar numeros negativos y positivos. A pesar de su exactitud, una representación en punto flotante consume una cantidad considerable de recursos y es más elaborada su implementación en hardware de arquitectura programable. Por ello, los fabricantes de FPGAs toman como standard la representación en complemento a dos para los multiplicadores embebidos e incluso para el *nucleo DSS compiler.* Asimismo, se ha mencionado que los filtros CIC, propuestos como una opción atractiva las etapas de filtrado, exhiben desbordamiento en sus acumuladores, misma que suele trabajarse sin problema al hacer uso de representación en complemento a dos. Por todo lo anterior, se elige dicho formato numérico en este trabajo de tesis.

Como último aspecto, es importante mencionar que una vez elegido el formato y la cantidad de bits de la representación es necesario decidir la cantidad de bits que corresponderan a las partes entera y decimal. En este punto conviene considerar que el nucleo DDS genera señales en el rango de -1 a 1 en complemento a dos, por lo que solamente requiere de dos bits para la parte entera. Si se asumen las mismas condiciones para la señal de entrada, se puede efectuar la multiplicación de los valores directamente, sin que esto implique alguna restricción importante sobre las señales de entrada. Así, resulta conveniente emplear 2 bits para la parte entera y 6 para la parte decimal.

3.2.2 Núcleo DDS

La descripción detallada del diseño de este bloque esta fuera de los alcances que se pretenden en este trabajo, por lo cual, la discusión sobre este bloque se limita a la metodologia que permite elegir sus parámetros e implementarlos con las librerias ofrecidas por los fabricantes para dicho fin.

A pesar de tratarse de un componente que va a operar en el dominio digital, el diseño del mismo se realiza considerando la frecuencia en el dominio analógico, por lo cual el análisis realizado previamente sobre la ubicación de las replicas en el dominio analogico es adecuado.

La frecuencia a la que las señales deben generarse es igual a la frecuencia central de la banda de interés después de haber realizado el muestreo paso-banda.

Para determinar la frecuencia de oscilación del sistema, el sistema DDS ofrece los siguientes parámetros.

La frecuencia del reloj F_{clk} configurable desde 0.01 hasta 50[Mhz]
El numero de bits B para el acumulador de fase desde 3 hasta 48 bits
Y el incremento en la fase por cada ciclo de reloj. $\Delta\theta$

La dependencia que tiene la frecuencia de oscilación en los parámetros de configuración es la siguiente:

$$F_o = \frac{F_{clk}\Delta\theta}{2^B} \tag{3.2.8}$$

Para una frecuencia de oscilación deseada, el incremento de fase está dado como:

$$\Delta\theta = \frac{F_o 2^B}{F_{clk}}$$

Un aspecto que no debe perderse de vista en el diseño que ha sido planeado es que la rapidez a la que se extraen de la memoria las muestras de la señal a procesar debe ser igual a la frecuencia de muestreo que se simuló en la computadora, de otra manera el producto de las señales generadas por el oscilador y las muestras de la señal generaría resultados erroneos.

Asimismo, dadas las fuertes restricciones asociadas al uso de puerto serie en este trabajo y a la incapacidad de contar con procesamiento en tiempo real, se desea hacer procesesamiento a máxima velocidad del sistema y asi observar el FPGA funcionando al máximo de sus capacidades.

Como se expresó anteriomente, extraer las muestras a 50[Mhz] conduciria a errores en la sincronización de ambas señales, por tal motivo sería necesario realizar algun ajuste para que esto pueda llevarse a cabo. Extraer las muestras a 50[Mhz], en vez de hacerlo a la frecuencia de muestreo del proceso de CAD Fm modifica la frecuencia de muestreo, por lo tanto si se modifica también la frecuencia de oscilación de las señales, pueden obtenerse resultados válidos.

Para explicar este cambio, se disntinguen:

F_o Frecuencia de oscilación requerida.
F_m Frecuecnia de muestreo.
F_o' Frecuencia de oscilación nueva.

Así pues, la relación existente entre la frecuencia de oscilación y la frecuencia de muestreo debe mantenerse al variar la frecuencia de muestreo, esto es.

$$\frac{F_o}{F_m} = \frac{F_o\prime}{50[Mhz]} \tag{3.2.9}$$

Por lo tanto, si se desea procesar a 50[Mhz] una señal que ha sido muestreada a una frecuencia F_m y que cuenta con una frecuencia central de la banda de interes de F_o, requerirá hacer uso de osciladores que funcionen a una frecuencia F_o' dada por la siguiente expresión:

$$F_o' = \frac{F_o(50[Mhz])}{F_m}$$

Ambos osciladores alimentan un multiplicador digital que debe obtener los productos de las muestras que corresponden tanto a la señal original como a las señales generadas localmente, por lo tanto, la longitud de palabra de todas ellas y la representación numérica deben ser las mismas para poder obtener resultados válidos a la salida de los multiplicadores.

3.2.3 Filtrado y decimación.

Como ha sido mencionado, los multiplicadores ofrecerán en sus salidas la representación en cuadratura de la señal, en la cual, tanto la componente en fase como en cuadratura exhibiran espectros en banda base. En este punto resta únicamente eliminar todo el ruido y las componentes no deseadas que presenta la señal a través del filtrado. Este proceso de filtrado opera de manera conjunta con un proceso de decimación, que tiene como objetivo obtener una menor cantidad de muestras al final del proceso.

El umbral que determina los límites para la reducción de la tasa de muestreo es el ancho de banda de la señal de interés, ya que la frecuencia de muestreo que se obtendrá al final de todo el proceso debe ser mayor que la frecuencia más grande presente en la señal(b/2, donde b es el ancho de banda de la señal).

Una vez estudiadas las caracteristicas de la tarjeta de desarrollo y sus capacidades, se puede afirmar que es posible construir filtros tanto CIC como FIR para operar con las muestras a la salida de los multiplicadores, ya que la multiplicación y la suma presentan tiempos de procesamiento, para 16 bits, correspondientes a 10 y 6 [ns], respectivamente, y pueden ser ejecutados a máxima velocidad de reloj.

A pesár de que la velocidad de operación del *hardware* permita implementar tanto filtros FIR como CIC, el ancho de banda de la señal impone una restricción importante, ya que al ser angosto con respecto al ancho de banda de trabajo, es necesario hacer uso de filtros con una banda de transición pequeña, misma que en el caso de los filtros FIR conduce a una gran cantidad de coeficientes por filtro. Por ende, al contar con una cantidad limitada de multiplicadores al interior de la tarjeta resultaria mucho más elaborado implementar filtros de longitud muy grande, así que resulta más conveniente hacer uso de filtros Hogenauer (CIC+decimación), para el filtrado. En estos filtros, la banda de transición del filtro no depende del número de multiplicadores disponibles.

3.2.3.1 Filtrado Hogenauer y sus parametros.

El uso de filtros CIC es compatible con procesos de diezmado, ya que conduce a estructuras eficientes de filtros con fase lineal que no presetan distorsión por retardo de grupo. La transformada Z de la respuesta impulso de un filtro CIC de varias etapas es la siguiente:

$$H(Z) = \left[\frac{1}{N}\left(\frac{1-z^{-N}}{1-Z^{-1}}\right)\right]^{K} \qquad (3.2.10)$$

En ella pueden visualizarse los dos parámetros para el diseño: el número de etapas K , y el numero de retardos del filtro peine N. Al usarse para procesos de diezmado (filtrado Hogenauer) suele elegirse N igual al factor de decimación. Dichos parámetros determinan la respuesta en frecuencia del filtro, en general, la forma que se tiene para su respuesta en magnitud es varios lobulos del mismo ancho,(un lobulo principal y varios lobulos laterales). A cada lado de los lobulos aparece un punto nulo, todos estos puntos se encuentran igualmente espaciados en la respuesta en frecuencia. El valor de N determinará la posición de estos puntos nulos (ceros del filtro) y por ende fijara la anchura del lobulo principal (banda de paso), a mayor N mayor cantidad de ceros y menor anchura del lobulo principal. Para la elección de este valor, es fundamental garantizar que todo el ancho de banda de la señal de interés este dentro del lobulo principal del filtro, esto es:

$$\omega_{max} < \frac{1}{N}$$

Por su parte, el valor de K en (3.2.10), el numero de etapas, determinará la atenuación ofrecida por los lobulos laterales. A mayor cantidad de etapas, se tendrá una mayor atenuación que eliminará más ruido y ofreceré mayor calidad en la señal obtenida. La bibliografia [Lyons 10] [Harris 04] reporta que cada etapa de filtrado incrementa en 16 dB la atenuación que existira en el primer lobulo lateral del filtro.

No obstante, incrementar el número de etapas tiene un costo, el cual corresponde a que en la estructura planteada existirán una mayor cantidad de integradores operando en cascada [Lyons10]. Lo anterior obliga a incrementar de manera importante la cantidad de bits para los acumuladores. Si no se considerara este aspecto, entonces la saturación de los mismos puede conducir a errores de gran magnitud.

Asimismo, los incrementos de k conducen a una deformación en la banda de paso, puesto que con cada etapa adicional, la banda de paso presenta una caida más pronunciada, tal y como se ilustra en la figura 3.2E.

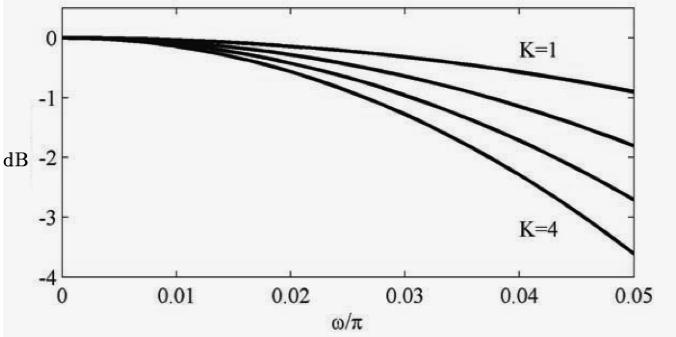

Fig 3.2E Caida en la banda de paso.

Aunque no se sacrifique el rendimiento de un filtro CIC al usar varias etapas CIC y se logre con ello una mejor atenuación en la banda de supresión, un valor de K elevado no resulta conveniente, ya que la banda de paso sufre una caida importante, tal y como se ilustra en la figura 3.2E.

En [Milic07] se señala que la caida en la banda de paso para una frecuencia f_m, con k etapas, puede cuantificarse de acuerdo a la siguiente expresión:

$$C_k = \left| \frac{\sin\left(\frac{\pi N f_m}{2}\right)}{N\sin\left(\frac{\pi f_m}{2}\right)} \right|^k \qquad (3.2.11)$$

Con ella, se suele medir el impacto que aparecerá en el peor caso de atenuación, que ocurrirá en la frecuencia máxima de la señal de interés, que corresponde a un medio del ancho de banda b, con lo cual (3.2.11) queda como:

$$C_k = \left| \frac{\sin\left(\frac{\pi N b}{4}\right)}{N\sin\left(\frac{\pi b}{4}\right)} \right|^k \qquad (3.2.11a)$$

En sintesis, se propone elegir primero un valor valor de k, tán grande que permita lograr la atenuación deseada en la banda de supresión, recordando que se añaden 16 [dB] por cada etapa, pero no tanto como para generar un impacto fuerte en la caida de la banda de paso para las frecuencias de interés. Y posteriormente seleccionar el valor de N lo suficientemente grande para que el primer lobulo del filtro permita contener únicamente las componentes deseadas de de la señal y lo suficientemente pequeño como para que la disminución en la frecuencia al final del proceso permita trabajar la señal de interés. De manera general, esta eleccion depende unicamente del ancho de banda de la señal, sin importar la frecuencia en la que la banda de interes se haya encontrado orginalmente, ya que son los componentes previos quienes se encargan de transladar la señal paso banda a banda base.

Las caracteristicas que presenta este tipo de filtro no son óptimas, por lo cual es necesario realizar un segundo proceso de filtrado que permita mejorar la calidad de la señal obtenida al final del proceso. Por otra parte, a diferencia de lo que ocurre en receptores de frecuencia intermedia, al usar muestreo paso-banda el factor de diezmado requerido no es tan grande, debido a que el tipo de muestreo es en si mismo un proceso que reduce la frecuencia de la señal.

Así pues, forzar a que la segunda etapa de filtrado realice tambien una etapa un proceso de diezmado puede limitar de manera importante el diseño del filtro Hogenauer. Por lo tanto, se propone realizar todo el diezmado a través del filtro CIC y dedicar la segunda etapa de filtrado solamente a mejorar la calidad de la señal obtenida.

3.2.3.1.1 Implementación del filtrado Hogenauer.

Tomando en consideración los textos disponibles, como [Meyer07], que hablan sobre la implementación de estos filtros y contienen algunos ejemplos de descripción de hardware, se procede a desarrollar este trabajo de tesis; sin embargo hay diversos aspectos que no son discutidos, en ellos se pretende hacer énfasis a continuación:

Las muestras con las que se alimentara el par de filtros que estaran presentes en el sistema, contaran con 16 bits, ya que tienen como origen los multiplicadores que trabajan con muestras de 8 bits.

Una multiplicación de dos palabras binarias en punto fijo, que presenten Z bits para la parte entera y Q bits de parte decimal, genera una palabra de 2Z bits enteros y 2Q decimales. En este diseño de receptor, las muestras que ingresan al sistema presentan Z=2, Q=6. Por lo cual, las muestras que ingresarán a los filtros tendran 4 bits de parte entera y 12 bits de parte decimal. La estructura interna del filtro contiene varios integradores operando en cascada, por lo que estos 4 bits que se tienen para la parte entera pueden ser insuficientes. Para resolver este inconveniente, los filtros deben realizar un proceso conocido como *bit extention,* que permite convertir una palabra binaria en otra de mayor cantidad de bits. Por tratarse de números en complemento a dos no puede simplemente agregarse bits iguales a cero a la izquierda de la palabra binaria. En cambio, debe obtenerse el valor del bit más significativo y asignarse a todos nuevos bits que se agregan a la palabra, con lo cual se obtiene una representación del mismo número para la nueva longitud de palabra.

Ahora bien, puesto que se requiere realizar un proceso de filtrado posterior en este y en casi todos los casos, el cual impone el uso de multiplicaciones, es importante mantener baja la cantidad de bits con los que se representa la señal, ya que los resultados obtenidos tendran una longitud del doble de la palabra de entrada. Por ello, la salida de los filtros CIC no puede conservar la cantidad de bits impuesta previamente para evitar desbordamiento. En este caso se cuenta con la limitante de tener multiplicadores de 18 bits, por lo que se decide conservar unicamente 16 bits.

La bibliografia no considera el manejo de datos con parte decimal, por lo que propone un truncamiento directo para hacer el ajuste a la longitud deseada, eliminando bits de la parte menos significativa, lo cual conduce a resultados aceptables en caso de trabajar con numeros enteros. En contraste, el uso de parte decimal , misma que se encontrará siempre en bits menos significativos, impedirá aplicar este mismo proceso, ya que se generarían palabras decimales con una cantidad considerable de bits para la parte entera y pocos o ningun bit para la parte decimal lo cual puede, en muchos casos, conducir a errores importantes.

Para justificar la solución que se propone al respecto, es importante considerar lo que ocurre en el proceso de *bit extention.* En dicho proceso se cumple lo siguiente: si se tiene una palabra de P bits y se desea representarla con una longitud mayor, R, bastará con repetir el bit mas significativo de P las veces que sea necesario hasta llenar los R bits, por lo que los ultimos bits de R (P+1, P+2, ... R) seran idénticos al bit P siempre y cuando la palabra binaria pueda ser representada dentro de P bits. Por lo cual, en caso de tener una palabra de R bits en complemento a dos y requerir hacer un truncamiento para obtener unicamente P bits , si la palabra binaria puede ser presentada en P bits, descartar los bits P+1, P+2, ... R no producirá perdida de información alguna.

Con base en lo anterior, se propone descartar la mayor cantidad de bits más significativos posible, de acuerdo a la magnitud máxima que se pueda presentar a la salida del filtro, y posteriormente eliminar bits de la parte menos significativa, para asi ajustar a los 16 bits deseados y minimizar el impacto del error por truncamiento. La bibliografia [Meyer07] muestra que el incremento más grande en el rango dinámico se presenta en $\omega = 0$; en este punto, el aumento en magnitud es de:

$$N^K$$

Donde N es el retardo factor de diezmado y K la cantidad de etapas, este cambio requerirá:

$$\log_2(N^k) = k \log_2(N) \tag{3.2.12}$$

bits adicionales.

Tomando como base lo anterior, si se toman $4+ k \log_2(N)$ bits para la parte entera y 16-$(4+ k \log_2(N))$ bits para la parte decimal, se obtendrá el menor impacto posible por error de truncamiento.

Por último, un aspecto que no puede dejarse de lado y que conviene mencionar, es que al trabajar en VHDL se manejan señales STD_LOGIC, que es un tipo de dato capaz de exhibir 6 valores diferentes. Si las operaciones aritméticas se efectuan sin que alguno de los operandos este disponible se obtendra un valor desconocido('X'), por ejemplo, una suma de '1' +'X' genera como resultado 'X'.

Lo anterior resulta indeseable, puesto que la realimentacion presente en los integradores hace que este error se acumule y genere el mismo resultado permanentemente. Por lo anterior, es imperativo añadir mecanismos al filtro que le permitan efectuar las operaciones hasta el momento en que sus operandos esten listos, de lo contrario el sistema estara incapacitado para ofrecer resultados validos.

3.2.3.2 Filtrado FIR.

Como ultima parte del procesamiento, es conveniente e incluso necesaria una etapa adicional de filtrado que permita completar el trabajo iniciado por los filtros CIC. Previamente se presentó la diferencia que existe con respecto al esquema de receptor digital de frecuencia intermedia: la reducción en la tasa de muestreo durante el fitrado no es de gran magnitud y por tanto conviene concentrar el diezmado en el filtrado CIC, y así maximizar la flexibilidad en su diseño. Desafortunadamente, algunos de los tipos de filtro que han sido discutidos en el capitulo 2, como los polifase, pueden resultar no muy convenientes ya que sus mayores ventajas se alcanzan cuando el filtro esta acompañado de procesos de decimación. De igual forma, en el caso de los filtros de media banda que minimizan el uso de multiplicadores, requiere, además de un proceso de diezmado que le acompañe, condiciones particulares en la respuesta en frecuencia, tales como simetría y rizos en en ambas bandas de la misma magnitud, las cuales no necesariamente son factibles.

Por ejemplo, en el primer caso se debe garantizar que la operación en conjunto, del muestreo paso-banda y la compresión de la tasa de muestreo, realizada durante el filtrado CIC, den como resultado que la información ocupe gran parte del intervalo de las frecuencias discretas [0,pi/2] , conservando la información de interés, y minimizando la cantidad de ruido presente. Este conjunto de condiciones, pueden complicar de manera muy significativa el diseño del receptor, por lo cual, no se consideran para este diseño, sin afirmar con ello que no puedan ser considerados como una opción viable.

Ante estos inconvenientes se debe plantear un proceso de filtrado con la menor cantidad de coeficientes posible, que no requiera incluir el diezmado, para explotar sus caracteristicas. Un filtro FIR convencional puede cumplir con estas caracteristicas si se garantiza que la cantidad de coeficientes es baja. El algoritmo de Parks-McClellan permite obtener el filtro con la mejor calidad, a partir de un conjunto de especificaciones dadas, y con fase lineal. Ante este escenario, puede emplearse una estructura de fase lineal para su implementación, permitiendo contruir filtros de hasta 18 coeficientes con los 9 multiplicadores disponibles para cada canal.

Como se ha presentado previamente, un diseño con el algoritmo de Parks-McClellan permite partir de una cantidad de coeficientes deseada para encontrar el filtro que mejor se ajuste a los requerimientos de bandas de paso y supresión. En el capitulo 2 se presentó una formula que permite aproximar la cantidad de coeficientes requerida , la cual se retoma a continuación.

$$N = \frac{-10\log_{10}(\delta_1\delta_2) - 13}{2.34(\omega_s - \omega_p)}$$

De dicha expresión, puede verse que, rizos en bandas de transición cercanos a uno o bandas de transición amplia, conducen a filtros de orden bajo; mientras que rizos cercanos a cero o bandas de transición angosta producen filtros de orden elevado. Con ayuda de MATLAB se puede hacer un estudio empiríco que permita, de acuerdo a las condiciones requeridas por cada señal, relajar las condiciones en forma iterativa hasta hayar un filtro que pueda ser construido con el hardware disponible , considerando siempre que las condiciones ofrecidas por el mismo sean aceptables.

3.2.3.2.1 Implementación del filtrado FIR.

Como se ha comentado anteriormente, una estructura de fase lineal permite aprovechar la simetría de los coeficientes de un filtro FIR y así utilizar únicamente la mitad de los multiplicadores que se requerirían para una implementación directa, lo que permite maximizar las capacidades del hardware.

Se puede referir al lector interesado a la bibiliografia [Meyer07] que presenta una gran gran cantidad de ejemplos sobre implementación de filtros FIR con VHLD. En lo que correponde a este trabajo, se pretende, tal y como se hizo para el caso de filtros CIC, exponer algunos de los aspectos que no se consideran o no se discuten y resultan de importancia en la implementación.

El primero de ellos es la longitud de palabra. El proceso de filtrado FIR cuenta con multiplicadores, por lo que las muestras duplicaran la cantidad de bits a su paso por el mismo. Si se tienen Z bits enteros y Q decimales, su salida exhibira 2Z bits de parte enterea y 2Q como parte decimal. En este punto existen dos alternativas, realizar un proceso de truncamiento como el que ha sido descrito para la salida de los filtros CIC o bien hacer uso de todos los bits que se obtengan y asi maximizar la precisión; en el caso particular de este trabajo, se opta por conservar los 32 bits, puesto que en este punto ya se habra realizado un proceso de truncamiento durante el fitltrado CIC que introducirá un error en la salida. Además, la interfaz de comunicaciòn con el puerto RS 232 facilita el manejo para el envio de multiplos de 8 bits.

Otro aspecto fundamental, que debe ser considerado al momento de trabajar con numeros positivos y negativos es que resulta necesario especificar en el código vhdl la naturaleza de los datos, ya que de lo contrario, las multiplicaciones que son inferidas conducen a resultados erroneos. Se sugiere que la comunicación del filtro con los demas bloques sea a través de señales STD_LOGIC_VECTOR y se indique de manera explicita la interpretacion de los bits como SIGNED al interior del mismo, de este modo se garantiza que las operaciones se realizan correctamente.

La necesidad de identificar si se trata de una multiplicación signada o no signada surge debido a que, en busca de optimizar el proceso, Xilinx modifica el algoritmo de multiplicación para no requerir la extensión del número de bits (de N a 2N bits). Dicho algoritmo es distinto en el caso de la multiplicación signada y no signada por lo cual resulta fundamental indicar, a través del tipo de dato, cual es el tipo de multiplicación que se desee realizar

Finalmente, con respecto a la estructura que se ha elegido para la implementación del sistema, la versatilidad asociada al FPGA puede ser aprovechada para que en lugar de implementarse una estructura iterativa, que sume uno a uno los productos obtenidos y retarde la salida de la señal N ciclos de reloj, se construyan sumadores de varios operandos, los cuales como se ha presentado previamente permiten ofrecer una salida de menos ciclos de reloj.

3.2.4 Almacenamiento de las muestras.

Una vez compleatado el análisis para el diseño del sistema, resta únicamente elegir el medio de almacenamiento de las muestras, tanto para las señales de entrada, que provienen del puerto serie, como las que se obtendran a la salida de los filtros. Por simplicidad y para evitar el uso de recursos para manejar memorias robustas como las que se tienen al exterior de la tarjeta, se hace uso de las memorias de bloque, que se encuentran al interior del FPGA, buscando realizar un diseño simple y lo suficientemente útil.

Como se ha planteado previamente, el receptor contará a su entrada con muestras de 8 bits, y a su salida con muestras de 32, por lo que las salidas seran 4 veces mas grandes que las entradas; sin embargo, el proceso de diezmado eliminará una cantidad importante de muestras, por lo que este hecho puede ser compensado.

Si se cuenta con P palabras de 8 bits para realizar el procesamiento, se obtendran a su salida $\frac{4M}{N}$ palabras, donde N es el factor de diezmado elegido, por cada canal. Esto se debe a a que la salida del sistema presentará palabras de 32 bits en su salida que son 4 veces mas grandes, y a que el numero de muestras totales, se vera reducido por un factor N durante el diezmado. Así, el número total de localidades de memoria requeridas sera de $M + 2 * \frac{4M}{N}$. Este número debera ser inferior a 40,000, que corresponde a la cantidad de palabras binarias de 8 bits que pueden ser almacenadas con las memorias de bloque al interior del XC3S500E.

Una vez concluido este analisis, el siguiente capitulo muestra ejemplos para varias condiciones de señal, haciendo uso de los parámetros que han quedado determinados a lo largo de este capitulo y calculando aquellos que dependen de la señal de interés.

Capítulo 4

Ajustes, pruebas y resultados experimentales.

4.1 Señales a procesar y ajustes.

Con el objetivo de evaluar el desempeño del receptor digital, es necesario elegir algunas señales que permitan ajustar el esquema genérico que ha sido planteado en el capitulo 3, a la frecuencia central y ancho de banda de cada una de ellas. En el capitulo 1 de este trabajo se presentó que las señales de resonancia magnética nuclear y las de acelerómetros MEMS son algunas áreas en donde este trabajo puede encontrar su campo de aplicación. Por tal motivo, se eligen valores para frecuencias máxima, mínima y anchos de banda que pueden ser encontrados comunmente en este tipo de señales.

4.1.1 Señales de RMN.

Existen reportes [Perez 99] que presentan algunos valores comunes para ancho de banda de las señales de resonancia magnetica: 12.5, 25 y 50[Khz], asi como para la frecuencia central de las mismas, conocida como frecuencia de larmor. Según el equipo con el que se trabaje, este valor puede ir desde 10[Mhz] hasta 1[Ghz]. Como primer ejemplo, se considera un equipo de 0.47[T], el cual presenta una frecuencia de Larmor de 20[Mhz]; como segundo caso, se considera un equipo como el que se estudia en [Perez 99] de 4.7 [T] que genera una frecuencia central de 200[Mhz]; en ambos casos, considerando señales con anchos de banda de 50[Khz].

4.1.2 Señales de un acelerómetro MEMS.

Un propósito adicional que motivó el desarrollo de este trabajo fue el procesamiento de señales de aceleración, que pueden ser generadas por acelerometros piezoeléctricos que funcionen con ondas acústicas superficiales.Por lo tanto, como tercer ejemplo, se considera una señal con una frecuencia central 200[Mhz] y un ancho de banda más amplio que el que aparece en RMN, en este caso se requieren 700[Khz].

4.1.3 Ajustes al diseño.

Para permitir la operación del esquema presentado previamente, con las señales de interés, sera necesario elegir:

*Un intervalo de muestreo de entre los que se obtienen a partir de la ecuación (2.1.5), así como una frecuencia de muestreo en ese intervalo.

*Una frecuencia de oscilación para la generación de señales requeridas por el demodulador I/Q, según la ubicación de la banda de interes producida por la frecuencia de muestreo elegida.

*Los parametros K y N para la etapa de filtrado CIC que permitan realizar un proceso de filtrado eficiente y comprimir la tasa de muestreo de la señal, de manera que se obtenga un ancho de banda de trabajo tal que pueda contener al ancho de banda de la señal al final del proceso.

*Finalmente, escoger los coeficientes para el filtrado FIR de fase lineal.

4.1.3.1 RMN 20[Mhz] y ancho de banda 50[Khz].

Una señal con estas caracteristicas, permite que la frecuencia de muestreo paso-banda se pueda elegir de entre 400 intervalos, la relación entre frecuencia de muestreo promedio en el intervalo valido y el ruido introducido, se obtiene a partir de:

```
> mpb_rdo(20000000-25000,20000000+25000);
```

La figura 4.1A presenta una comparativa entre los 65 valores de frecuencia promedio más grandes y el factor de incremento en el ruido asociado a dicha frecuencia; se observa que en caso elegir valores de k mayores a 10, el incremento en el ruido, puede no ser conveniente, ya que el decremento que aparece en la frecuencia de muestreo promedio se hace cada vez mas pequeño.

Por lo anterior al considerar valores de k mayores a 10, debe evaluarse si conviene seguir aumentando el nivel de ruido presente en la señal, con respecto la reduccion en la frecuencia de muestreo que pueda conseguise.

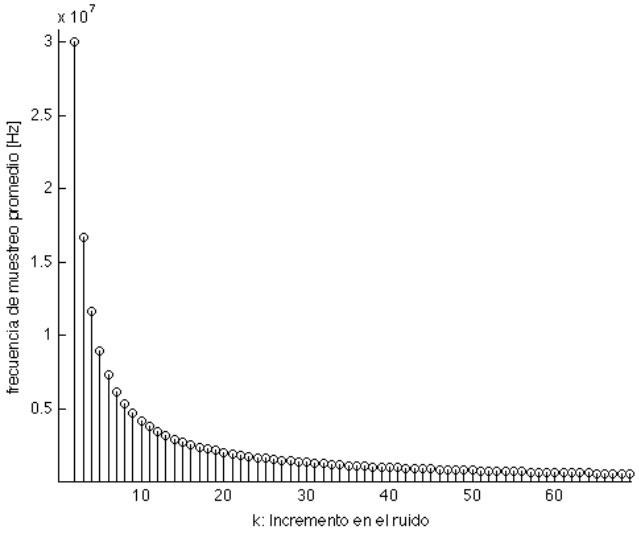

Fig 4.1A Relación entre ruido y frecuencia de muestreo promedio $F_c = 20[Mhz]$, B=50[Khz].

Los intervalos validos para la frecuencia de muestreo paso-banda se pueden obtener a partir de:

```
> fm_pb(20000000-25000, 20000000+25000);
```

En la tabla 4.T1 se presentan algunos de ellos para ilustrar las diferentes condiciones que pueden ser encontradas, destacandose aquellos que se encuentran proximos a k=10 de tal manera que puedan ser comparados.

Observe para valores de k elevados, se obtienen intervalos de muestreo demasiado angostos, que solo permiten variaciones mínimas en la frecuencia de muestreo, lo cual es poco conveniente.

K	Fmax	Fmin	ΔF[Hz]
2	3.995e+07	2.0025e+07	1.9925e+07
3	1.9975e+07	1.335e+07	6.625e+06
7	6.6583e+06	5.7214e+06	9.369e+05
8	5.7071e+06	5.0062e+06	7.0089e+05
9	4.9938e+06	4.45e+06	5.4375e+05
10	4.4389e+06	4.005e+06	4.3389e+05
11	3.995e+06	3.6409e+06	3.5409e+05
12	3.6318e+06	3.3375e+06	2.9432e+05
20	2.1026e+06	2.0025e+06	1.0013e+05
50	8.1531e+05	8.01e+05	14306
100	4.0354e+05	4.005e+05	3035.4
200	2.0075e+05	2.0025e+05	503.77
300	1.3361e+05	1.335e+05	112.04
400	1.0013e+05	1.0012e+05	0.31328

Tabla 4T1 Intervalos validos de muestreo , $F_c = 20[Mhz]$, B=50[Khz].

Por otro lado, la ejecución de:

```
> mpb_posv(20000000-25000,20000000+25000,[8 9 10 11 12]);
```

Permite obtener las curvas que corresponden a la ubicación de la banda de interés en el intervalo de [0, Fm/2], para los valores de k elegidos, como función de la frecuencia de muestreo. Con el objetivo de mantener baja la cantidad de ruido y no reducir la frecuencia de muestreo por un factor muy grande, que afecte la flexibilidad del diseño de los filtros CIC, se opta por un factor k=8 para trabajar, ya que inicrementa la cantidad de ruido en forma moderada y el rango de operación de 5.7-5.02[Mhz] puede ser manejado por el hardware disponible.

Pese a tratarse de un factor par, el efecto de inversión del espectro asociado no tiene consecuencias, ya que se asume simetria en el espectro. En general, los factores pares tiene la ventaja de que al decrementar la frecuencia de muestreo, también se acerca la banda de interes hacia banda base.

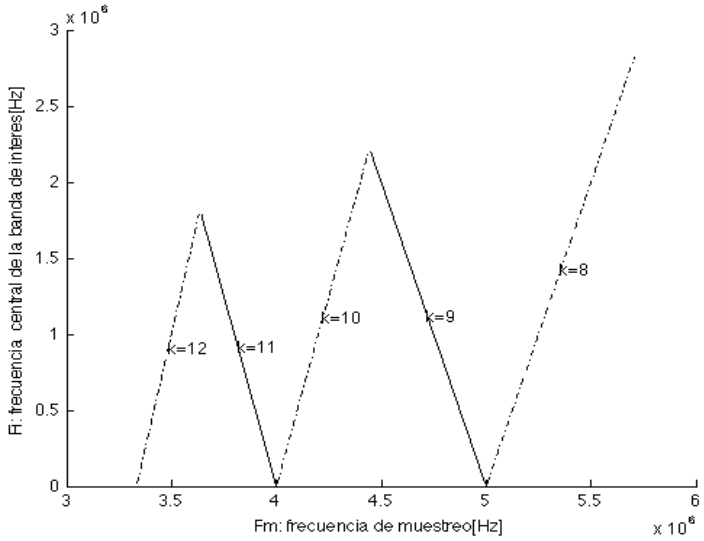

Fig 4.1B Ubicación de la frecuencia de interes para algunos valores de k, $F_c = 20[Mhz]$, B=50[Khz].

En ese intervalo, se elige una frecuencia de 5.2[Mhz] para el muestreo, con el objetivo de trabajar con una frecuencia baja en dicho intervalo, sin ser lo suficientemente cercana al limite inferior, como para que la variación en la frecuencia de muestreo pueda ser un problema. Esta elección situa la banda de interes en 800[khz] y produce un ancho de banda b, en el dominio discreto, de 50[khz]/5.2[Mhz]*2=0.0192[rad/muestra].

Según lo anterior, los osciladores deberan operar a 800[Khz] para realizar el traslado a banda base,;no obstante para trabajar con el hardware diseñado para maxima frecuencia del reloj, de acuerdo con (3.2.9) el oscilador deberá operar a:

$$\frac{50{,}000{,}000(800000)}{5{,}200{,}000} \approx 7{,}692{,}307.69[Hz]$$

Si se elige un acumulador de 29 bits y se trabaja con el reloj de 50[Mhz], el $\Delta\theta$ que permite configurar los osciladores numericos es, de acuerdo con (3.2.8):

$$\frac{7{,}692{,}307.69(2^{29})}{50{,}000{,}000} \approx 82{,}595\ 525\ pasos$$

Elegidos estos valores, se requiere seleccionar los parámetros para los filtros Hogenauer, que corresponden a elegir el numero de etapas K y el retrazo del filtro comb N que debe ser igual al factor de diezmado. En el capitulo 3 se mencionó que cada etapa CIC añade una atenuación en el lobulo lateral de -16[dB], en terminos más precisos esto significa:

$$-16 = 20log(\frac{x}{100})$$

Donde:

$$x = 10^{-\frac{16}{20}} * 100 = 15.8\%$$

Es decir, aquellas componentes que se ubiquen mas alla del primer lobulo lateral, conservaran unicamente el 15.8% de su valor original. De manera similar, una atenuación de -32[dB], ofrecida por un arreglo de 2 etapas permite reducir hasta el 2.5% del valor original el valor de las componentes no deseadas; si bien la elección de valores mas elevados de k ofrece mejores atenuaciones, estos incrementan el numero de bits que requieren los acumuladores y producen una caida en la banda de paso mas pronunciada, por lo tanto se elige k=2 para éste y los demas ejemplos considerados en este trabajo de tesis.

Para el diseño del filtrado Hogenauer, resta únicamente la elección del parametro N. En este punto se debe recordar que este valor debe producir una reducción en el ancho de banda de trabajo, tal que el ancho de banda de la señal pueda ser contenido en el ancho de banda de trabajo al final del proceso. La tabla 4T2 presenta un resumen de las caracteristicas de filtros CIC para distintos valores de N, que resultan convenientes para tipo de señal considerada en este caso.

N	ω_c [rad/m]	Bits extra	C_{max}[dB]
5	0.13	5	-0.02
10	0.0625	7	-0.09
20	0.033	9	-0.3
30	0.021	10	-0.7
40	0.016	11	-1.4
50	0.011	12	-2
100	0.006	14	-10

Tabla 4T2 Variación de N en filtros CIC con K=2.

Puede observarse que una elección de N=100, o mayor no permite el paso de la señal, ya que la frecuencia de corte es de 0.006[rad/m] mientras que la frecuencia máxima que exhibe la señal es de 0.0096[rad/m]. Se ha planteado que debe haber una segunda etapa de filtrado FIR para mejorar la calidad de la señal, por lo que es conveniente mantener en ancho de banda relativo de valor bajo, que permita efectuar un filtrado con una cantidad de coeficientes reducida.

Una elección de **N=20,** conduce a buenos resultados, ya que presenta un numero razonable de bits en los acumuladores y ofrece una atenuación máxima de solo 0.3 [dB]. Es importante mantener bajo este valor, ya que el filtro FIR que se diseñará enseguida, tambien presentara tolerancia en la banda de paso; por ello si se elige un valor alto en esta etapa, se puede comprometer el rendimiento del siguiente filtro. La respuesta que ofrecerá el filtro CIC propuesto se ilustra en la figura 4.1C. El uso de un factor 20 produce un ancho de banda en el dominio discreto de la señal de 0.0192*20=0.384[rad/m] y una frecuencia angular máxima de 0.192[rad/m], misma que debera servir como parametro para la construcción del filtro FIR.

Como parte final del proceso de ajuste, resta la elección de los coeficientes del filtro FIR mediante el algoritmo de Parks-McClellan, considerando que se cuenta con 8 multiplicadores disponibles por canal, lo cual permite que se tengan filtros con un maximo de 16 coeficientes.

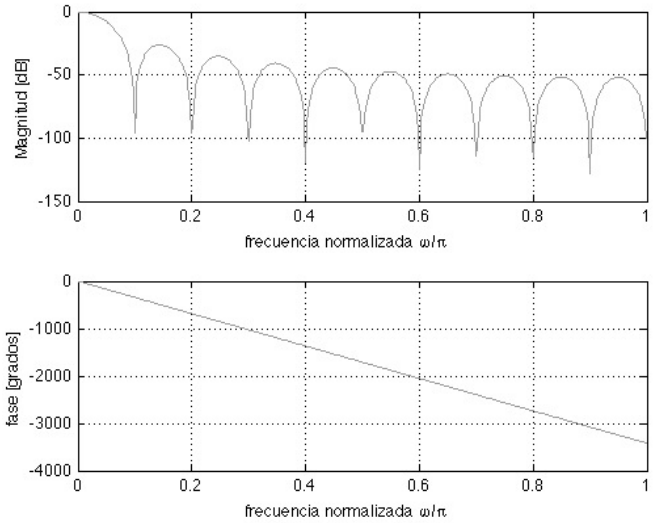

Fig 4.1C Respuesta del filtro CIC, K=2 N=20.

Para llevar a cabo este proceso se ofrece un programa en matlab[1], que parte de las especificaciones en decibeles para las atenuaciones y genera un filtro a través del algoritmo de Parks Mc-Clellan, empiricamente, se han visto buenos resultados con atenuaciones que rondan los 30[dB] para el lobulo lateral y 1[dB] para la banda de paso, variando iterativamente los valores para las frecuencias de paso y supresión hasta que logre generarse un filtro con los 16 coeficientes deseados.

En este caso, la ejecución de:

```
> fopt(0.18,0.31,30,1)
```

Genera un filtro de 16 coeficientes con las caracteriscicas necesarias para la señal elegida, el cual exhibe la respuesta en frecuencia presentada en la figura 4.1D.

[1] La ejecución de fopt(ω_s, ω_p, A, d) (fopt.m), ofrecido en los anexos, genera los coeficientes del filtro FIR requeridos para contar con: frecuencia de paso ω_s, frecuencia de supresión ω_s, y atenuaciones en las bandas de paso y supresión, d y A respectivamente. Estas ultimas, expresadas en [dB].

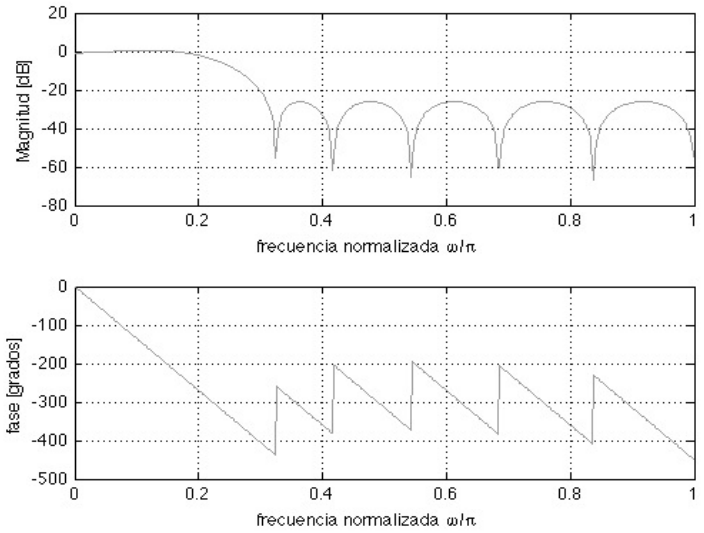

Fig 4.1D Respuesta del filtro FIR, M=16 $\omega_c = 0.21[rad/m]$

4.1.3.2 RMN 200[Mhz] y ancho de banda 50[Khz].

Este ejemplo permite ilustrar las bondades del uso de muestreo paso-banda, ya que la frecuencia central de la señal, obligaría a muestrear a frecuencias superiores a los 400[Mhz], según el teorema de muestreo de señales paso baja. Sin embargo, en este caso se pretende emplear el hardware para recibir muestras a una tasa cercana a las 50[Mm/s]. El muestreo de esta señal, se puede relajar con una frecuencia a seleccionar de entre 4000 intervalos posibles, la figura 4.1E presenta los primeros 80, observandose que valores de k próximos a 10, son aquellos que permiten comenzar a trabajar con frecuencias de muestreo cercanas a los 50[Mhz]; además, a partir de dicho valor, los incrementos en la frecuencia de muestreo promedio son poco significativos respecto al incremento en el ruido asociado a valores de k mas elevados, por ende se analiza lo que ocurre entorno a k=10 en la tabla 4T3.

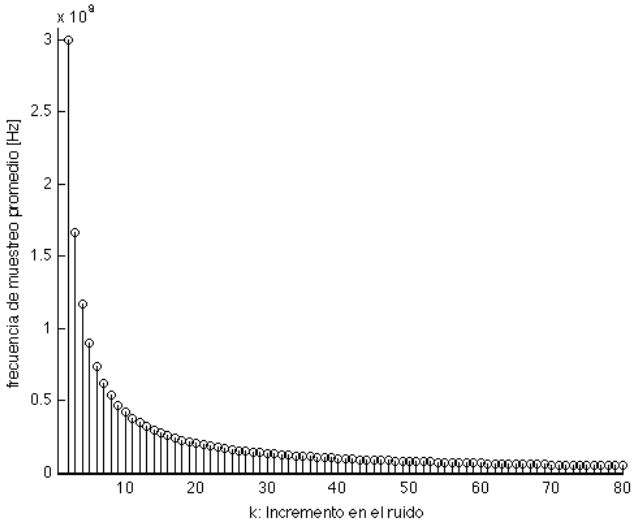

Fig 4.1E Relación entre ruido y frecuencia de muestreo promedio $F_c = 200[Mhz]$, B=50[Khz].

k	Fmax[Hz]	Fmin[Hz]	**Δ [Hz]**
2	**3.9995e+08**	**2.0002e+08**	**1.9992e+08**
7	**6.6658e+07**	**5.715e+07**	**9.5083e+06**
8	**5.7136e+07**	**5.0006e+07**	**7.1295e+06**
9	**4.9994e+07**	**4.445e+07**	**5.5438e+06**
10	**4.4439e+07**	**4.0005e+07**	**4.4339e+06**
11	**3.9995e+07**	**3.6368e+07**	**3.6268e+06**
12	**3.6359e+07**	**3.3338e+07**	**3.0216e+06**
13	**3.3329e+07**	**3.0773e+07**	**2.5561e+06**
14	**3.0765e+07**	**2.8575e+07**	**2.1904e+06**

Tabla 4T3 Intervalos validos de muestreo , $F_c = 200[Mhz]$, B=50[Khz].

De la tabla puede verse con que valores de k mayores a 8 se puede trabajar con los 50[Mhz] de la tarjeta. El análisis de la posición de la banda de interés según la frecuencia de muestreo se muestra en la figura 4.1F.

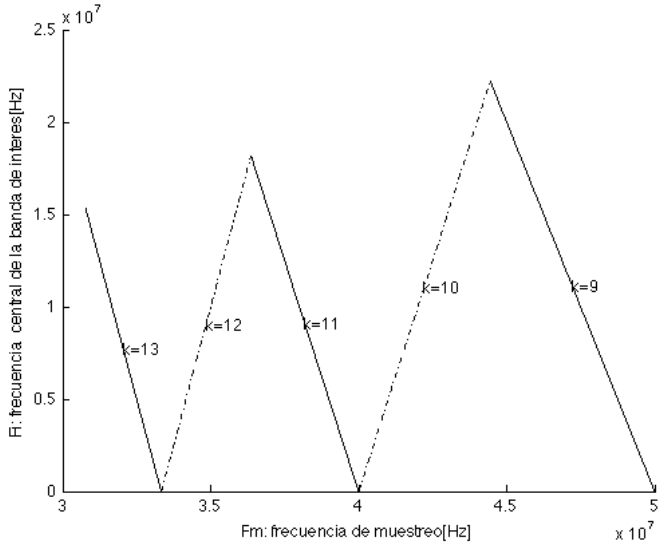

Fig 4.1F Ubicación de la frecuencia de interes para valores de k, $F_c = 200[Mhz]$, B=50[Khz].

De lo anterior, se opta entonces por elegir el valor de k=10, de tal manera que se pueda trabajar lo suficientemente cerca de la frecuencia máxima de 50[Mhz], y mantener el nivel de ruido bajo. Para la elección de la frecuencia de muestreo a lo largo de dicho intervalo, se buscó elegir una frecuencia en medio de intervalo válido, lo que eventualmente permitiria trabajar con señales de un ancho de banda mayor y, al mismo tiempo dejar replica espectral no deseada, que se genera al desplazar el espectro, en el limite el ancho de banda de trabajo, lo que permite eliminarla fácilmente, puesto que la atenuación ofrecida por los filtros en esa region es la de mayor magnitud. Se elige entonces una frecuencia de muestreo de 42[Mhz], lo que situa la replica espectral en 10[Mhz] según la grafica 4.1F.

Para que el sistema funcione a máxima frecuencia de reloj, de acuerdo con (3.2.9), se deberan generar señales para los osciladores de:

$$\frac{50{,}000{,}000(10{,}000{,}000)}{42{,}000{,}000} \approx 11{,}904{,}761.9[Hz]$$

Con lo anterior, si se elige un acumulador de 29 bits y se trabaja con el reloj de 50[Mhz], entonces de acuerdo con (3.2.8) se requiere que el parametro de conguración $\Delta\theta$ sea igual a:

$$\frac{11{,}904{,}761.9(2^{29})}{50{,}000{,}000} \approx 127\,826\,408\,pasos$$

A continuación deben elegirse los parámetros para el filtro Hogenauer. Previamente se ha discutido que un factor de K=2 resulta conveniente y ofrece simplicidad importante, por lo que resta elegir el valor de N. Para ello debe considerarse el ancho de banda de la señal, esto es: 50[khz]/42[Mhz]*2= 0.00238[rad/m].

N	ω_c [rad/m]	Bits extra	C_{max}[dB]
5	0.13	5	0
10	0.0625	7	0
20	0.033	9	-0.01
30	0.021	10	-0.02
40	0.016	11	-0.04
50	0.011	12	-0.06
100	0.006	14	-0.2

Tabla 4T4 Valores para distintos N en filtros CIC

Para mantener bajo el número de bits extras y eliminar una cantidad importante de muestras, se elige un factor N=30, que ofrece una atenuación máxima en la banda de paso muy pequeña. Con esta elección, el ancho de banda relativo de la señal a la entrada del filtro FIR sera de 0.0022*30=0.071[rad/m], con una frecuencia máxima de 0.0357[rad/m]. La respuesta del filtro CIC obtenida para k=2 y N=30 se ilustra en la figura 4.1G.

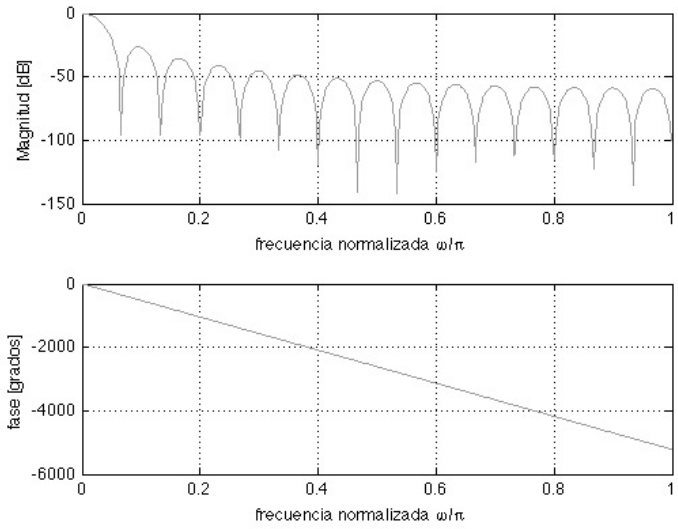

Fig 4.1G Respuesta del filtro CIC, K=2 N=30.

Al tener una frecuencia maxima de 0.35[rad/s] se plantea un filtro FIR con las siguientes caracteristicas:

```
> fopt(0.03,0.16,30,1)
```

Esto da como resultado, un filtro de 16 coeficientes cuya respuesta en frecuencia se presenta en la figura 4.1H

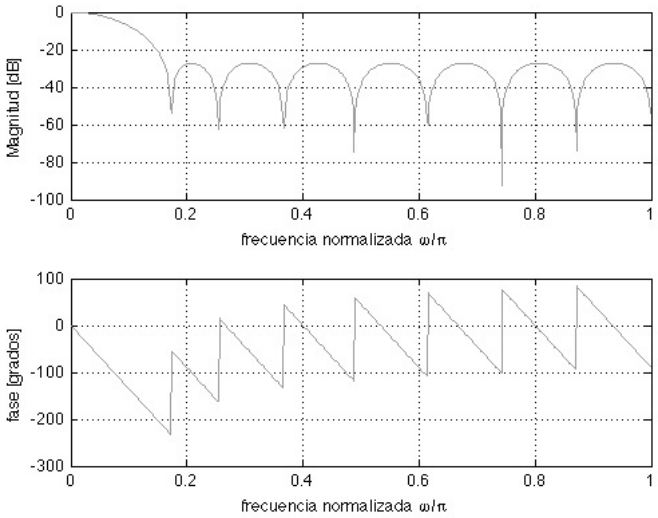

Fig 4.1H Respuesta del filtro FIR, M=16 $\omega_c = 0.05[rad/m]$

4.1.3.3 MEMS 200[Mhz] y ancho de banda 700[Khz]

El objetivo de esta última prueba es observar la capacidad del receptor para trabajar con señales de ancho de banda mayores. Para esto se parte de una señal de la misma frecuencia central que en el caso anterior pero con ancho de banda mucho mayor, como el que puede encontrarse en señales de acelerómetros MEMS. En el diseño anterior, se eligió deliveradamente, un intervalo y una frecuencia de muestreo que permitieran trabajar con un ancho de banda mayor, por lo cual es posible utilizar las mismas condiciones para muestreo que en el ejemplo de RMN. Esto es, frecuencia de muestreo de 42[Mhz], que ubica la replica espectral en 10[Mhz], y permite que los parametros calculados para el oscilador sean identicos.

Enseguida, se debe hacer la elección del factor N del filtro CIC que permita trabajar con el nuevo ancho de banda, en el dominio discreto este estará dado como: 700[Khz]/42[Mhz]*2=0.033[rad/m], con una frecuencia máxima de 0.0166[rad/m].

N	ω_c [rad/m]	Bits extra	C_{max}[dB]
5	0.13	5	-0.05
10	0.0625	7	-0.173
20	0.033	9	-0.7
30	0.021	10	-1.6
40	0.016	11	-2.9
50	0.011	12	-6

Tabla 4T5 Valores para distintos N en filtros CIC

La tabla muestra que N=20 resulta conveniente para el ancho de banda de la señal. Con este factor, el ancho de banda de salida será de 0.033*20=0.66[rad/m], con una frecuencia máxima de 0.33[rad/m]. La respuesta del filtro CIC elegido es idéntica a la que se presentó en la figura 4.1C.

La ejecución de

```
> fopt(0.3,0.43,30,1)
```

Produce un filtro FIR con las caracteristicas que se requiere para permitir filtrar las señales con el nuevo ancho de banda de 0.33[rad/m], la cual se ilustra en la figura 4.1I.

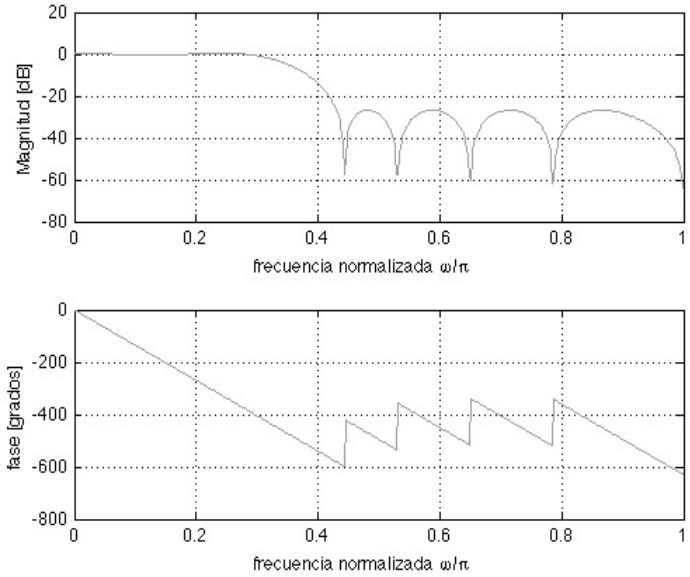

Fig 4.1I Respuesta del filtro FIR, M=16 $\omega_c = 0.33[rad/m]$

4.2 Resultados.

Con el objetivo de demostrar la operación del sistema, se realizan pruebas en señales que tienen las caracteristicas mencionadas previamente. Para esto se efectua una modulación AM que produzca señales con la frecuencia central deseada, partiendo de señales en banda base y con ancho de banda conocido. Una vez en alta frecuencia, se añade ruido gausiano para reproducir el efecto de un canal de transmisión sobre la misma y posteriormente, se realiza un proceso de muestreo y cuantización bajo las condiciones elegidas para cada caso, con ello se generan todos los datos con los que se alimentará el FPGA. Los resultados que fueron obtenidos se presentan a continuación.

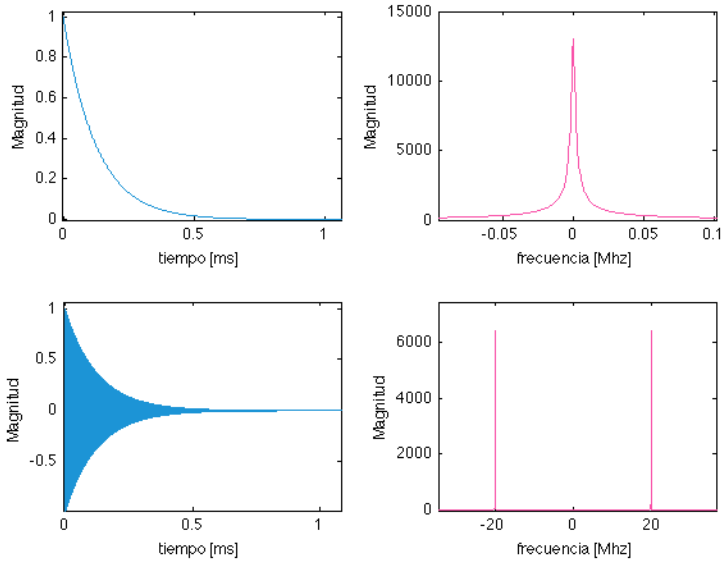

Fig 4.2A Señal de RMN en banda base y en 20[Mhz] , B=50[Khz]

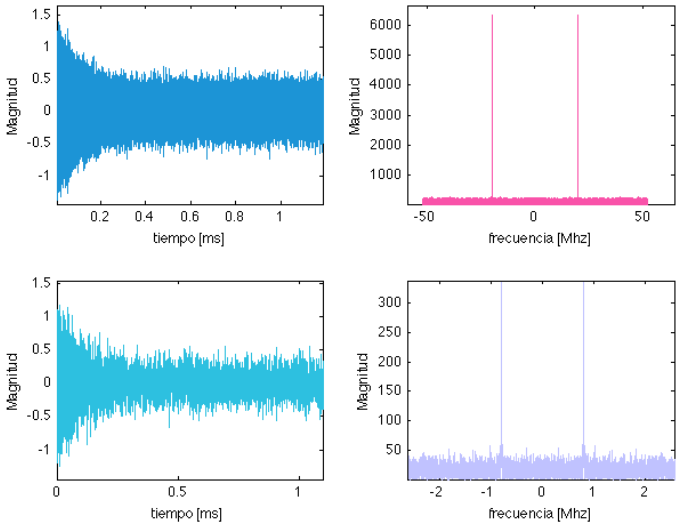

Fig 4.2B Señal de RMN con ruido SNR=15[dB] y muestreada Fm=5.2[khz]

Si bien las señales digitales deben ser interpretadas en el dominio de las frecuencias discretas, se presentan, en las figuras 4.2A y 4.2B, resultados en terminos de la frecuencia analógica para constatar la posición de la replica espectral; en este caso, la figura 4.2B presenta la localización de la replica en 800[khz], tal y como habia sido predecido. Observese tambien el efecto del muestreo paso-banda. En esta señal se habia considerado un k=8, que produciría un aumento de esa misma magnitud en el ruido, el cambio puede observarse en la misma figura.

Se efectua un procedimiento similar para generar la señal de RMN, pero ahora ubicada en los 200[Mhz], el cual se presenta en la figura 4.2C.

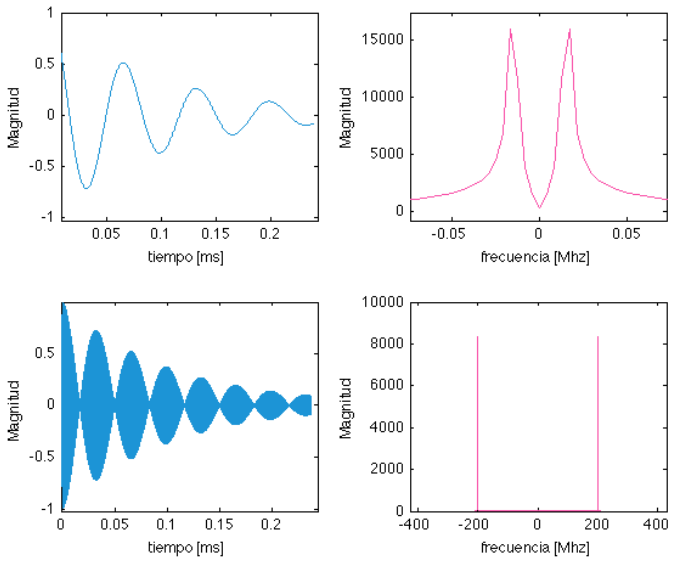

Fig 4.2C Señal de RMN en banda base y en 200[Mhz], B=50[khz]

En este caso también se añade ruido para observar el incremento al efectuar el proceso de muestreo, esta señal permite realizarlo a una frecuencia de 42[Mhz], lo cual, según se predijo, produce una replica ubicada en los 10[Mhz]. Estas dos condiciones se pueden corroborar a través de la figura 4.2D.

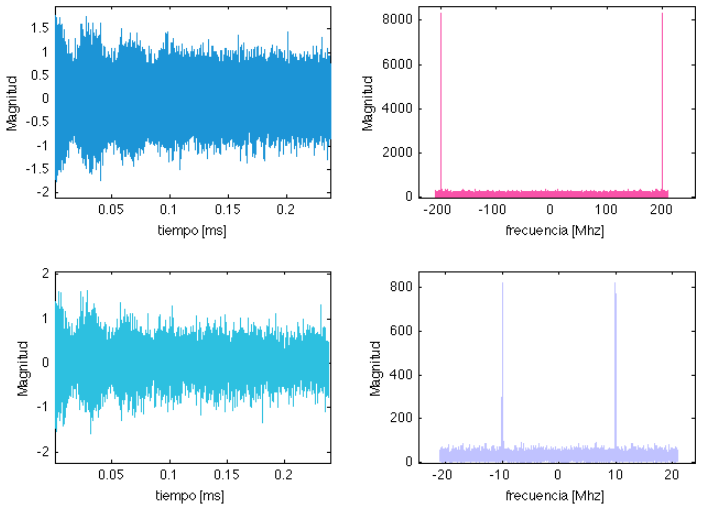

Fig 4.2D Señal de RMN con ruido SRN=10[dB] y muestreada Fm=42[Mhz]

Por último, se presenta en la figura 4.2E señal de trabajo que posée las caracteristicas que aparecen en señales producidas por acelerómetros MEMS, la ilustración del efecto del ruido y la ubicación espectral han sido demostrados en la figura anterior, por lo cual no son mostradas. Las señales producidas por el muestreo y cuantización simulados en MATLAB se envian directamente al FPGA a través del puerto serie. Es importante destacar que se hace uso de 10,000 muestras, por lo que el tiempo de procesamiento es de 0.238[ms] para las señales de 200[Mhz] y de 2.38[ms] para la señal de 20[Mhz]

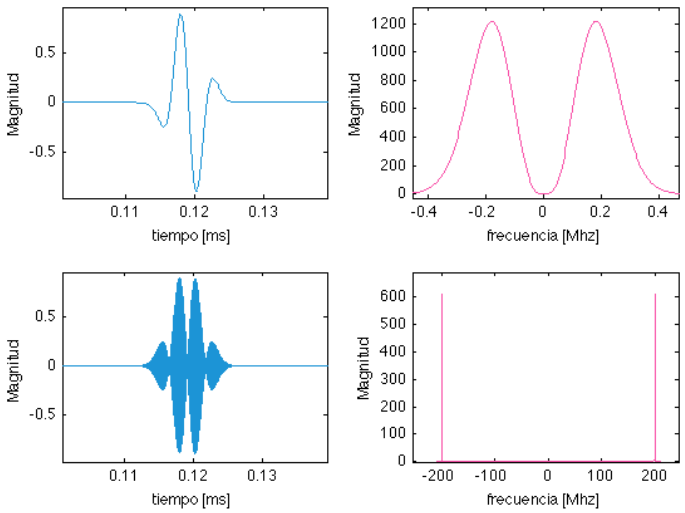

Fig 4.2 E Señal MEMS en banda base y en 200[Mhz], B=700[khz]

En los ejemplos previos se eligió una SRN baja, para que fuese claro el efecto del muestreo paso-banda sobre el ruido; Sin embargo, para las primeras pruebas directamente en el FPGA, se toman las formas de onda de la figura 4.2E y 4.2C asumiendo un nivel de ruido mucho menor (SNR=40[dB]), que permita visualizar claramente las formas de onda y apreciar la operación del mismo. La primera prueba realizada, tuvo como objetivo mostrar el efecto del angulo de fase en la recepción de señal, para lo cual se realizaron pruebas variando el ángulo entre las portadoras del receptor y del emisor; es decir, modificando el ángulo con el que se genera la portadora de AM. El resumen de dichas pruebas se presenta en las figuras 4.2F y 4.2G

Tomando como base la señal que aparece en la figura 4.2E, se realizo 5 veces el procesamiento de 10,000 Muestras en el FPGA, variando el angulo de defasamiento en el primer cuadrante, es decir: $0, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3} y \frac{\pi}{2}$ [rad].

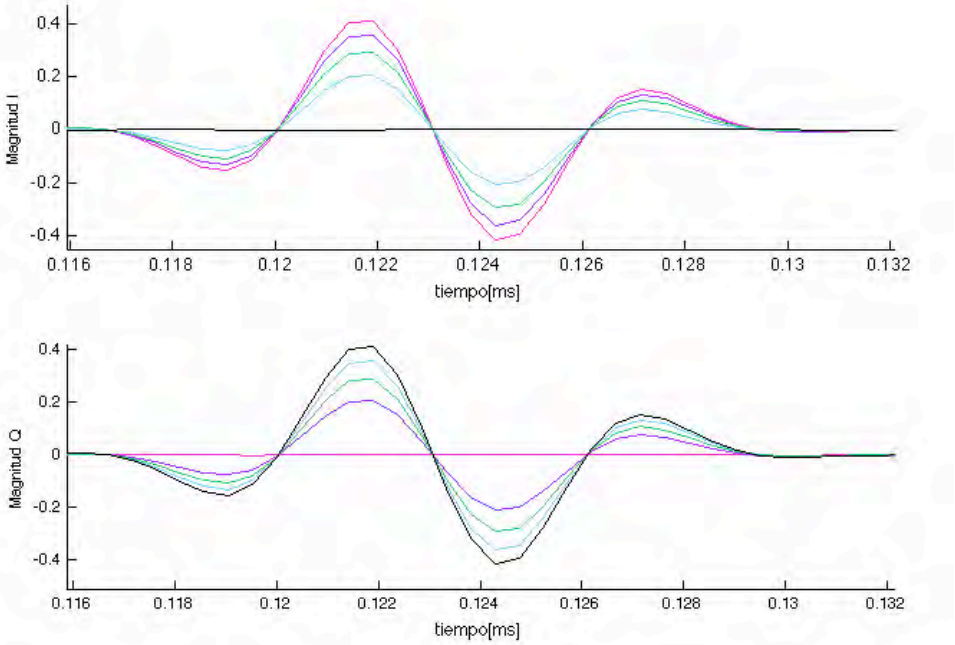

Fig 4.4F Efecto de la fase en las componentes recibidas para $\phi = 0\ (rosa), \frac{\pi}{6}(morado), \frac{\pi}{4}(verde), \frac{\pi}{3}(azul), \frac{\pi}{2}(negro)$. B=700[khz],Fc=200[Mhz],SNR=40[dB]

Observe que cuando el angulo es cero (rosa), se tiene demodulación coherente, la componente en fase (I) tiene toda la información de la señal , mientras que el canal en cuadratura (Q) no cuenta con información alguna. Asimismo, en la medida en que el ángulo de defasamiento se incrementa, la componente en fase decrece en magnitud. Cuando el ángulo de defasamiento se acerca a 90 grados (negro), toda la infomación estará contenida en la componente en cuadratura.

Note la similitud que este comportamiento tiene con las componentes de un vector que se ubique en el primer cuadrante, cuyo ángulo medido desde el eje positivo de las x es ϕ. La componente en fase corresponde a la proyección sobre el eje x del vector, mientras que la componente en cuadratura a su proyección sobre el eje y.

Como segunda prueba se considera la forma de onda que aparece en la figura 4.2C, pero ahora se varía el angulo de defasamiento con valores que se ubiquen en los 4 cuadrantes.

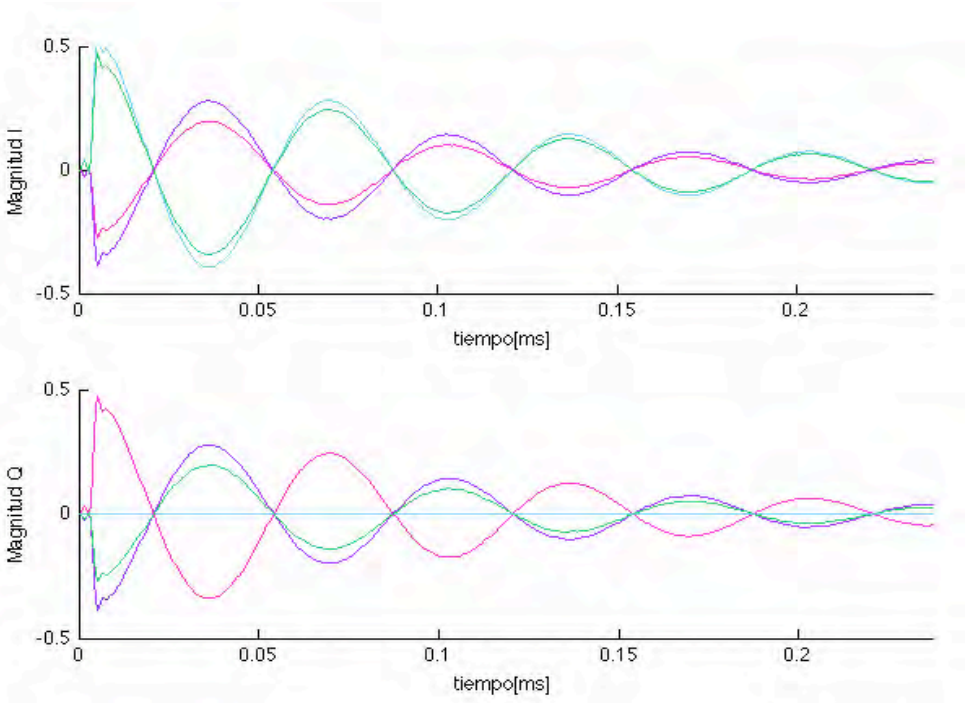

Fig 4.2G Efecto de la fase en las componentes recibidas para $\phi = 0\ (azul), \frac{4\pi}{6}(rosa), \frac{5\pi}{4}(morado), \frac{11\pi}{6}(verde)$, B=50[Khz], Fc=200[Mhz], SNR=30[dB].

De esta segunda prueba puede observarse que el comportamiento es tambien analogo a la proyección de un vector sobre los ejes x y y, ya que en los cuadrantes donde la funcion coseno toma valores positivos, la componente en fase toma amplitud positiva, mientras que ésta se torna negativa (inversión en la forma de onda) en los valores de ϕ donde la función coseno toma valores negativos. El mismo fenómeno relaciona la componente en cuadratura y la función seno, observe que en todos los casos la información queda contenida en alguno de los canales, por lo que se valida la inmunidad que la demodulación I/Q presenta a la diferencia entre fases.

El analisis anterior permite simplificar el procedimiento para evaluar el rendimiento del sistema, ya que si se asume un angulo de defasamiento de 0[rad] es posible concentrarse en un solo canal para analizar la vulnerabilidad al ruido del sistema propuesto. En el primer caso se estudió, la señal de RMN centrada en 20[Mhz] y con un ancho de banda B de 50[Khz]. El procesamiento de la señal en el FPGA para distintos niveles de ruido presentó los resultados que se muestran en las figuras 4.2H y 4.2I.

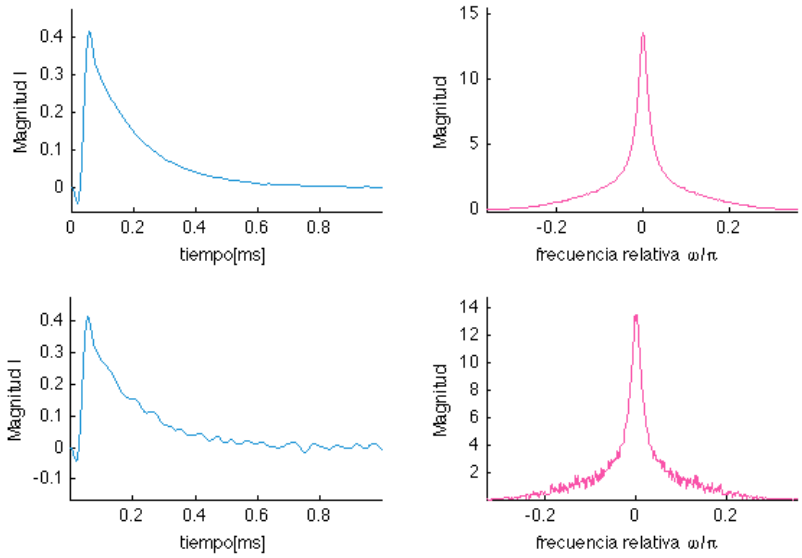

Fig 4.2H Señal procesada, componente (I) , B=50[Khz], Fc=20[Mhz], SNR=40[dB](arriba) y SNR=20[dB](abajo).

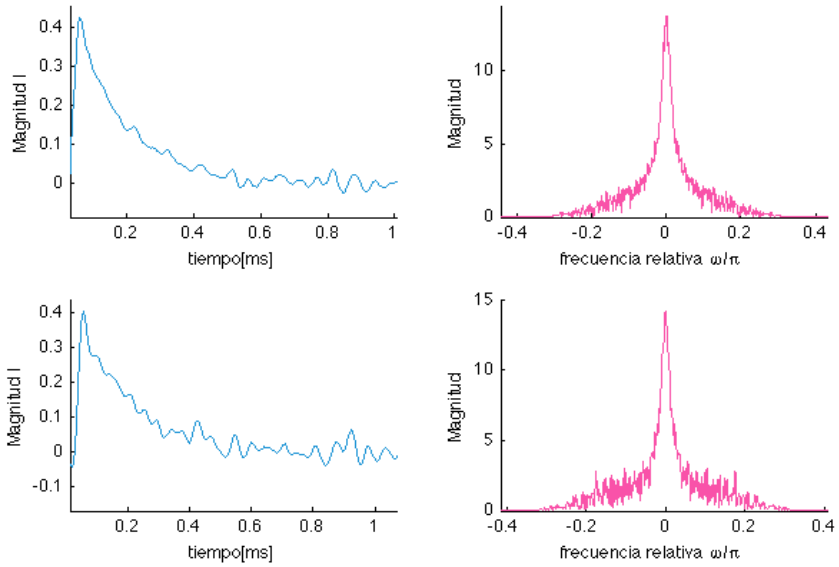

Fig 4.21 Señal procesada, componente (I) , B=50[Khz], Fc=20[Mhz], SNR=15[dB](arriba) y SNR=10[dB](abajo).

Dependiendo de la aplicación, podran resultar convenientes los niveles de ruido que se obtienen, el problema que aparece aquí es la extensión que presenta el espectro de una exponencial decreciente, el cual se extiende en la medida en que la señal exponencial decrece mas rapidamente, añadiento componentes de mayor frecuencia de muy poca magnitud. En este caso se consideraron muchas de las componentes que presentan amplitud muy pequeña, por lo cual el nivel de ruido que puede soportarse es medio; por ejemplo, si se considera a 20[dB] como un nivel de ruido aceptable, la amplitud de la señal con respecto al ruido debera de ser 10 veces mayor.

La herramienta ISE de xilinx presenta un resumen de los recursos utilizados para la implementación en el FPGA, para este primer caso se tiene:

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,435	9,312	15%	
Number of 4 input LUTs	1,308	9,312	14%	
Number of occupied Slices	1,106	4,656	23%	
Number of Slices containing only related logic	1,106	1,106	100%	
Number of Slices containing unrelated logic	0	1,106	0%	
Total Number of 4 input LUTs	1,395	9,312	14%	
Number used as logic	1,276			
Number used as a route-thru	87			
Number used as Shift registers	32			
Number of bonded IOBs	3	232	1%	
Number of RAMB16s	10	20	50%	
Number of BUFGMUXs	4	24	16%	
Number of MULT18X18SIOs	12	20	60%	
Average Fanout of Non-Clock Nets	2.04			

Fig 4.2J Consumo de hardware para la señal de RMN , B=50[Khz], Fc=50[Mhz].

Notese que unicamente se requiere 11.5% del total de recursos lógicos del dipositivo ya que del 23% de los *slices* empleados no se hace uso los SLICEM, sino unicamente de los SLICEL. Las herramientas de sintesis que xilinx permiten optimizar el consumo de recursos, de tal manera que solo se consumen aquellos slices que sirven para funciones logicas. Se observa ademas que se hace uso de solo 12 de los 20 multiplicadores, debido a que algunos de los coeficientes tienen el mismo valor y a que las herramientas de optmización reducen la cantidad de multiplicadores requeridos.

El segundo ejemplo que se considera, tiene como finalidad observar la operación del dispositivo con una señal de frecuencia mayor.

Las señales que se observaron condicionando el sistema a operar en demodulación coherente, con diferentes niveles de ruido, se presentan en las figuras 4.2K y 4.2L.

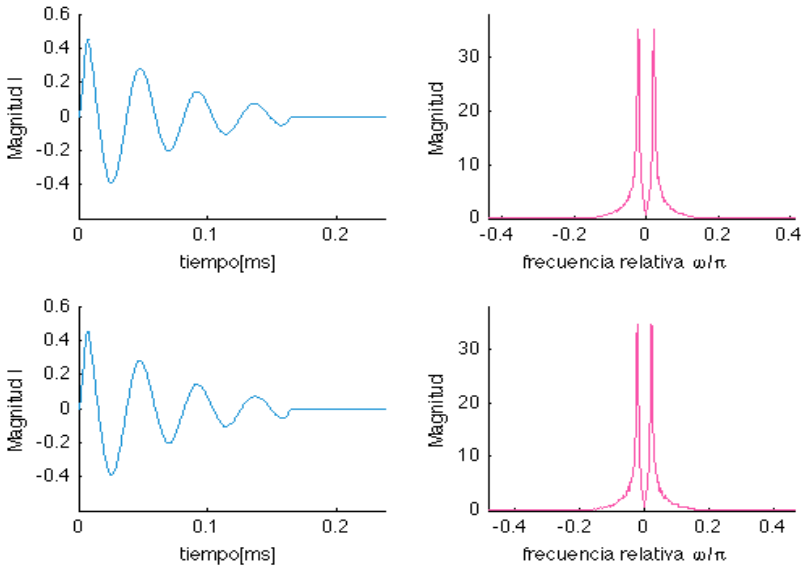

Fig 4.2K Señal procesada, componente (I) , B=50[Khz], Fc=200[Mhz], SNR=40[dB](arriba) y SNR=20[dB](abajo).

Debido a que se observarón bajos niveles en la señal resultante ruido se realizaron pruebas de hasta 10 y 5 [dB], las cuales se presentan enseguida.

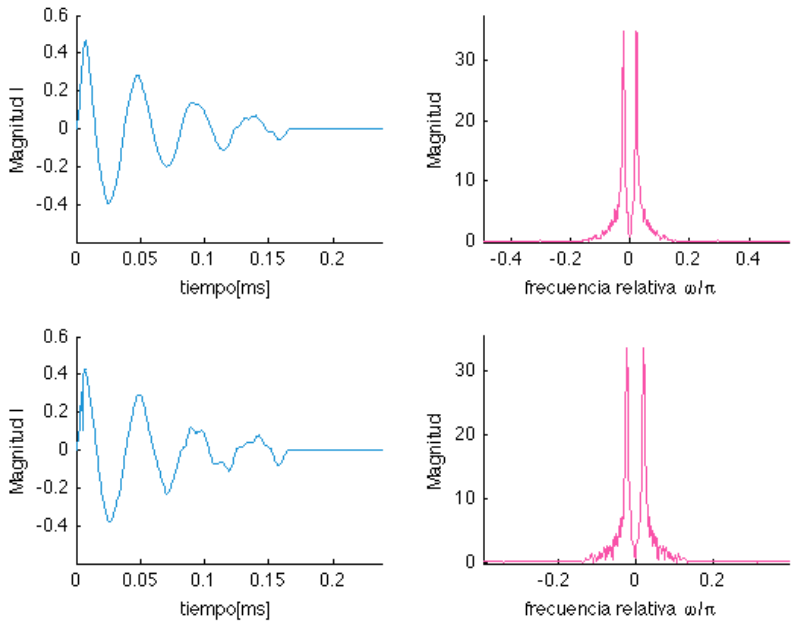

Fig 4.2L Señal procesada, componente (I) , B=50[Khz], Fc=200[Mhz], SNR=10[dB](arriba) y SNR=5[dB](abajo).

Se observa una inmunidad al ruido importante, ya que la selectividad del filtro Hogenauer es mayor que en el diseño para 20[Mhz], lo cual se manifiesta en una señal de mayor calidad, incluso para niveles de 10[dB]. Se observa por ejemplo que para el caso de 5[dB] aunque el ruido que puede verse en el espectro pudiese parecer limitado, hay un deterioro en la forma de onda original, lo cual puede resultar poco viable, por ejemplo en el caso de imágenes de resonancia magnetica nuclear.

La herramienta ISE de xilinx presenta un resumen de los recursos utilizados para la implementación en el FPGA de este segundo diseño:

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,461	9,312	15%	
Number of 4 input LUTs	1,205	9,312	12%	
Number of occupied Slices	1,110	4,656	23%	
Number of Slices containing only related logic	1,110	1,110	100%	
Number of Slices containing unrelated logic	0	1,110	0%	
Total Number of 4 input LUTs	1,285	9,312	13%	
Number used as logic	1,173			
Number used as a route-thru	80			
Number used as Shift registers	32			
Number of bonded IOBs	3	232	1%	
Number of RAMB16s	10	20	50%	
Number of BUFGMUXs	4	24	16%	
Number of MULT18X18SIOs	10	20	50%	
Average Fanout of Non-Clock Nets	2.07			

Fig 4.2M Consumo de hardware para la señal de RMN , B=50[Khz], Fc=200[Mhz].

En este caso, se requiere una cantidad similar de recursos logicos, sin embargo la herramienta de sintesis realiza una simplificación importante en cuanto al numero de multiplicadores requeridos ya que a pesar de contar con 16 multiplicadores para este propósito, se obtiene una implementación completa del diseño que solo hace uso de 10.

Para concluir las pruebas, se realiza un trabajo de recepción de señal con un ancho de banda mayor, que corresponde a las especificaciones requeridas para una señal MEMS, con ancho de banda de 700[Khz]. Haciendo uso del mismo proceso de modulación y asumiendo las mismas condiciones que para las pruebas anteriores se obtuvieron los siguientes resultados:

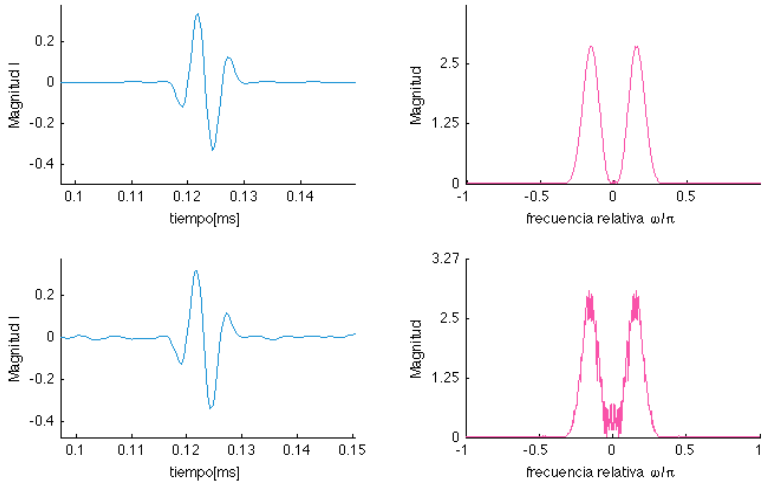

Fig 4.2N Señal procesada, componente (I) , B=700[Khz], Fc=200[Mhz], SNR=40[dB](arriba) y SNR=20[dB](abajo).

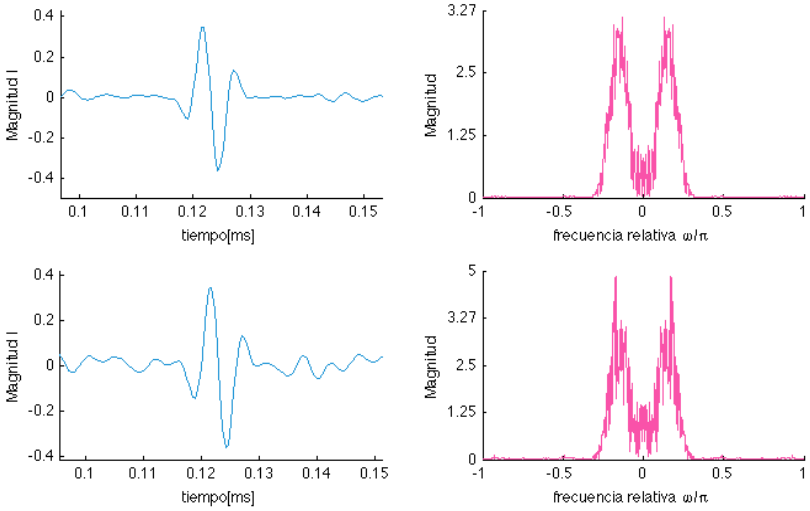

Fig 4.2O Señal procesada, componente (I) , B=700[Khz], Fc=200[Mhz], SNR=15[dB](arriba) y SNR=10[dB](abajo).

En este ultimo ejemplo se corrobora nuevamente la operación del dispositivo, observandose una vulnerabilidad al ruido media, con niveles aceptables de hasta SNR de 15[dB]; a diferencia de los demas casos, este ejempo permite apreciar la cantidad importante de ruido que aparece en la banda que la señal, si bien el filtrado que se ha planteado ofrece buena atenuación fuera de la banda de paso, las componentes del ruido que aparecen en la banda de la señal pueden ofrecer una distorsión significativa, afectando la forma de onda. Para la eliminación de las mismas hara falta hacer uso de tecnicas mas avanzadas de procesamiento de señal. Por último se presenta el consumo de hardware que se requiere para la implementación del sistema, que nuevamente ofrece valores similares a los diseños anteriores.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,423	9,312	15%	
Number of 4 input LUTs	1,303	9,312	13%	
Number of occupied Slices	1,092	4,656	23%	
Number of Slices containing only related logic	1,092	1,092	100%	
Number of Slices containing unrelated logic	0	1,092	0%	
Total Number of 4 input LUTs	1,387	9,312	14%	
Number used as logic	1,271			
Number used as a route-thru	84			
Number used as Shift registers	32			
Number of bonded IOBs	3	232	1%	
Number of RAMB16s	10	20	50%	
Number of BUFGMUXs	4	24	16%	
Number of MULT18X18SIOs	12	20	60%	
Average Fanout of Non-Clock Nets	2.04			

Fig 4.2P Consumo de hardware para la señal de RMN , B=700[Khz], Fc=200[Mhz]

Capítulo5

Conclusiones.

5.1 Metas alcanzadas.

Los primeros tres objetivos del presente trabajo pueden resumirse como la presentación de alternativas y herramientas teoricas ofrecidas por el procesamiento digital de señales con el fin de seleccionar las más adecuadas para el diseño de un receptor digital, capaz de operar en el rango de radio frecuencias. Durante el desarrollo de este analisis se observaron las ventajas asociadas al uso del muestreo paso-banda en receptores digitales, siendo la más importante el que esta técnica permite disminuir significativamente la velocidad de muestreo por debajo de la establecida para el muestreo de señales pasobajas. Reducir la frecuencia de muestreo hace posible realizar el proceso de conversión analogica digital en señales de frecuencias mayores, y en algunos casos directamente en el rango de radio frecuencia, con lo que puede eliminarse el uso de componentes analogicos para el traslado de la señal a frecuencia intermedia. Este hecho se traduce en un sistema más preciso y con mayor estabilidad en su funcionamiento, ya que depende en menor medida de las condiciones ambientales o del deterioro natural de los componentes analogicos, al mismo tiempo, simplifica el proceso de diseño y aumenta la capacidad del sistema para ser reproducido fácilmente.

Un estudio del estado del arte en convertidores analogicos digitales reveló que existen varios dispositivos que permiten aplicar muestreo paso-banda en señales de hasta 2.8[Ghz], por lo que este enfoque cobra relevancia y un futuro prometedor.

El analisis realizado, condujo tambien al planteamiento de un esquema de procesamiento basado en el modelo usado para frecuencia intermedia, efectuando los ajustes que le permitiesen hacer uso de muestreo paso-banda. Como parte del proceso de diseño se realizó un analisis sobre las implicaciones de muestrear por debajo de la frecuencia Nyquist, el cual condujo a resultados importantes.

En primer lugar se obtuvo una comparativa entre la reducción en la frecuencia de muestreo, que puede obtenerse con muestreo paso-banda, y el deterioro que sufre la señal como consecuencia de dicho muestreo, lo cual permite tener un mejor criterio para elegir una frecuencia de muestreo.

En segundo lugar, se realizó un estudio de muestreo paso-banda que se tradujo en un algoritmo que conduce a la ubicación exacta de las frecuencias de interés, una vez que se ha efectuado el muestreo paso-banda. Anteriormente la bibliografia solo habia reportado los intervalos validos para el muestreo sin hacer mención de la localización en frecuencia de la información a procesar. Este hecho representa un avance importante en la aplicación del muestreo paso-banda en receptores digitales, en donde contar con esta información es de vital importancia, puesto que el traslado a banda base de la señal debe ser efectuado con un conocimiento preciso de la ubicación de la información.

Asimismo, se observaron algunas de las restricciones inherentes al muestreo paso-banda, siendo una de las más importantes, la reducción en la flexibilidad del sistema, ya que los parámetros de operación dependen en gran medida de las caracteristicas de cada señal a procesar. En busqueda de generalidad en el sietema se planteó un esquema genérico que pudiese ser empleado para distintas señales, acompañado con algunos programas en matlab que permiten lidiar con la reducción en flexibilidad y permiten realizar ajustes al esquema propuesto de acuerdo a las necesidades de cada diseñador.

La validez de estos resultados fue demostrada en forma practica, ya que se realizo la implementación de la parte digital del sistema haciendo uso de un FPGA. La elección de este dispositivo tuvo como motivación su versatilidad y capacidad para generar arquitecturas optimizadas para los algoritmos de procesamiento, y a que permite construir una gran cantidad de unidades de procesamiento operando de manera paralela, lo cual reduce el tiempo de espera de las operaciones para ser ejecutadas.

Para la implementación y el diseño de este trabajo, se tomaron en cuenta las capacidades del harware, su rendimiento y las limitantes asociadas al uso del mismo. El FGPA con el que se trabajó, es capaz de operar a 50[Mhz], por lo cual se diseñaron pruebas orientadas al uso de este dispositivo. Se plantearon esquemas que permiten procesar señales de 20 y 200[Mhz], En el último de los casos, seria necesario muestrear por encima de los 400[Mhz] y contar con hardware capaz de operar a esa misma velocidad bajo un esquema de sobremuestreo, en este caso se logro efectuar satisfactoriamente el procesamiento con hardware digital operando a 50[Mhz].

El enfoque elegido produjo resultados favorables en todos los casos planteados, el diseño genérico permitió adaptar los bloques a señales con diferentes anchos de banda y frecuencias centrales, los cálculos y predicciones realizadas demostraron su validez con los resultados obtenidos. A pesar de tratarse de un esquema simple y acotado por el hardware disponible, fue posible el procesamiento de señales con difrentes niveles de ruido, demostrando un grado importante de vulnerabilidad al mismo, permitiendo lidiar con relaciones señal a ruido de hasta 15 [dB] aún cuando se emplean filtros con una cantidad reducida de coeficientes para el filtrado FIR y de pocas etapas en el caso del filtrado CIC. Los esquemas de filtrado que fueron planteados lograron una eliminiación importante del ruido a pesar de su holgura, ya que se elige diseñar filtros con atenuaciones en la banda de supresión de solo -32[dB].

De igual forma, se mostró el alcance del FPGA y la capacidad de las herramientas de sintesis realizadas por Xilinx. Los diseños realizados, conteniendo todos los bloques necesarios, pudieron contruirse al interior del dispositivo ocupando únicamente alrrededor del 11% de los recursos logicos de proposito general disponibles, y con menos de los 18 multiplicadores que se consideraban para el diseños del mismo, obteniendose sintesis de hardware que requirieron el uso unicamente de 12 y 10 multiplicadores. Con lo anterior se demostró que a pesar de tratarse de un dispositivo básico, de pruebas y que no esta orientado a la implementación de sistemas muy complejos, cuenta con la suficiente capacidad para implementar bloques de procesamiento de señal importantes.

Por todo lo anterior, es posible afirmar que se ha cumplido a cabalidad con los objetivos planteados, y se cuenta con una plataforma importante que puede ser de utilidad en desarrollos posteriores sobre esta misma línea de trabajo. Se han dejado una serie de programas que permiten ajustar el diseño a diferentes condiciones y tipos de señal. Además, la descripción de hardware que se ha realizado, permite ser sintetizada para algún otro hardware capaz de operar a mayor velocidad.

Además de ello, se ofrece una recopilación de herramientas para mejorar el diseño propuesto, mismas que son ofrecidas en textos de procesamiento multitasa, los cuales exigen un nivel importante de especialización en el área de procesamiento de señal y suelen demandar una cantidad importante de tiempo por parte del lector para extraer los conceptos fundamentales de cada herramienta. Este trabajo trata de ofrecer un enfoque cualitativo de los mismos y permite acceder más facilidad a bibliografia especializada teniendo en mente una idea más clara de los principios basicos detrás de cada herramienta.

5.2 Mejoras propuestas y trabajos futuros.

El esquema que ha sido desarrollado e implementado, dista mucho de ser óptimo o de ofrecer los mejores resultados que pueden ser obtenidos con el hardware disponible hoy en dia.

La tarjeta de desarrollo empleada, Spartan 3E de Xilinx, no esta orientada en concreto a procesamiento digital de señales y opera a una velocidad media; hoy en dia, pueden ser encontradas tarjetas de desarrollo que pueden operar a frecuencias mas elevadas, hasta 200[Mhz] y cuentan con una mayor cantidad de hardware orientado a procesamiento digital de señales que permitiría la eventual construcción de un sistema más preciso, robusto y capaz de procesar señales de un rango de frecuencias más elevado.

De considerar condiciones similares de ruido en la señal, este hardware permitiría el manejo de señales con frecuencias centrales alrrededor de los 800[Mhz]. O bien podria elevarse la cantidad de ruido en el sistema y elevar el rango de frecuencias permitido hasta los rangos que permiten los convertidores analogicos digitales actuales, es decir, hasta 2.8[Ghz].

En cualquier caso, es posible optimizar varios aspectos en el esquema planteado, puede mencionarse que el uso del puerto RS-232 y la baja velocidad con la que puede operar impiden una implementación capaz de operar en tiempo real, lo cual es fundamental para cualquier aplicación práctica. En este sentido deberá desarrollarse un esquema de comunicación basado en algun otro protocolo de comunicación que sea capaz de operar a frecuencias mayores.

El esquema de trabajo con 8 bits puede resultar ineficiente para muchas aplicaciones debido a la cantidad de ruido de cuantización asociada, por lo cual es importante aumentar la cantidad de bits de trabajo a la entrada del dispositivo y generar un analisis que permita realizar los ajustes y truncamiento de datos para que pueda lidiarse con los incrementos en la longitud de palabra que ocurren tanto en el demodulador I/Q , como en el filtrado FIR.

En lo que respecta a las etapas de filtrado, es posible mejorar el trabajo de eliminación de ruido si se hace uso de un esquema mas general para filtrado CIC. En este trabajo, se asumió que el factor de diezmado era igual al retardo del filtro peine, sin embargo, en el articulo donde se propone el trabajo con estos filtros [Hogenauer 81], puede encontrarse un esquema más general, que contempla una un factor adicional que permite reducir la longitud del lobulo principal sin necesidad de variar el factor de diezmado. Esto permitiría dividir el trabajo de compresión de la tasa de muestreo en ambas etapas de filtrado y permitir el uso de filtros orientados a procesamiento multitasa en la segunda etapa. En particular, el uso de filtros de media banda y filtros polifase puede resultar una opción conveniente.

Anexos

Codigo matlab y descripción de hardware en VHDL

A.1| Programas en Matlab.

fm_pb.m Calculo de los intervalos validos para muestreo pasobanda, requiere f_{max} y f_{min} presentes en x(t)

```
% Calculo de los intervalos para muestreo de señales pasobanda
% dados: fmax (f2) y fmin (f1) presentes en la señal.

function fm_pb(f1,f2)
  ls=floor(f2/(f2-f1));%valor entero maximo para k (2.1.6)
  for k=2:ls;%valores posibles para k
      fmx(k-1)=2*f1/(k-1);%frecuencias maximas (2.1.5)
      fmn(k-1)=2*f2/(k);%frecuencias minimas.
      df(k-1)=fmx(k-1)-fmn(k-1);%diferencia entre ellas.
      i(k-1)=k;%valor correspondiente de k
  end;

  disp('          k       Fmax[Hz]      Fmin[Hz]    delta[Hz]')
  disp('-------------------------------------------------------')
  format('shortG')%formato para mostrar los valores de f.
  disp([i' fmx' fmn' df'])%muestra valores obtenidos
```

mpb_rdo.m Comparativa velocidad de muestreo promedio e incremento en el ruido, requiere f_{max} y f_{min} presentes en x(t).

```
function mpb_rdo(f1,f2)
    %f1: Fmin, f1: Fmax.
    ls=floor(f2/(f2-f1));%valor entero maximo para k (2.1.6)
    fc=(f1+f2)/2;%frecuencia central

    figure
    xlabel('k: Incremento en el ruido');
    ylabel('frecuencia de muestreo promedio [Hz]');
    set(gcf,'Color',[1,1,1])
    hold
    for k=2:ls;
        fmx=2*f1/(k-1);%frecuencias maximas (2.1.5)
        fmn=2*f2/(k);%frecuencias minimas.
        fp(k-1)=(fmx+fmn)/2;
    end;
    stem([2:ls],fp,'black');
```

mpb_posv.m Calculo de la posición exacta de la frecuencia central para varios valores de k.

```
function mpb_posv(f1,f2,V)
     %f1: Fmin, f2: Fmax , V: vector con valores de k (e.g [1 3 4 5 9])
     ls=floor(f2/(f2-f1));%valor entero maximo para k (2.1.6)
     fc=(f1+f2)/2;%frecuencia central

     figure
     xlabel('Fm: frecuencia de muestreo[Hz]');
     ylabel('Fi: frecuencia  central de la banda de interes[Hz]');
     set(gcf,'Color',[1,1,1])

     hold
     for k=V;
      fmx=2*f1/(k-1);%frecuencia maximas (2.1.5)
      fmn=2*f2/(k);%frecuencia minimas.
      fp=(fmx+fmn)/2;
      vl=[fmn:(fmx-fmn)/50000:fmx];

        if mod(k,2)~=0;% k impar
             ls=fc/fp;
          li=(fc/fp)-1/2;
          if((floor(ls)-floor(li))==1)%ls, li  consecutivos
              m=floor(ls);% la m-sima replica se ubica en [0,Fm/2]
              fr=-m.*vl+fc
              plot(vl,fr,'black');
              text(vl(250),fr(250),strcat('k=',int2str(k)));%etiquetas
          end;

        else %k par;
            ls=fc/fmn+1/2;
            li=fc/fp;
            if((floor(ls)-floor(li))==1)%ls, li  consecutivos
                m=floor(ls)% la m-sima replica se ubica en [0,Fm/2]
                fr=m.*vl-fc;
                plot(vl,fr,'-.black');
                text(vl(250),fr(250),strcat('k=',int2str(k)));
            end;
        end;
    end;
```

comb_res.m Obtención de la respuesta en frecuencia de los filtros CIC, a partir del numero de etapas K, y el retardo del filtro comb. N.

```
function comb_res(N,k)
    n=[1 zeros(1,N-1) -1]/N;
    d=[1 -1];
    for i=1:k-1;
       n=conv(n,[1 zeros(1,N-1) -1]/N);
       d=conv(d,[1 -1]);
    end;
    freqz(n,d);
```

fopt.m Diseño de filtros FIR por medio del algoritmo Parks-McClellan, a partir de las frecuencias relativas y atenuaciones en la banda de paso y supresión.

```
function [h]=fopt(wp,ws,As,Rp)
   % wp Frecuencia de paso normalizada.
   % ws Frecuencia de paso normalizada.
   %As atenuacion en la banda de supresion
   %Rp rizo en la banda de paso.

   delta1=(10^(Rp/20)-1)/(10^(Rp/20)+1);
   delta2=(1+delta1)*(10^(-As/20));

   deltaM=max(delta1,delta2);deltamn=min(delta1,delta2);

   deltaf=(ws-wp)/2;
   M=ceil((-20*log10(sqrt(delta1*delta2))-13)/(14.6*deltaf)+1);%calculo del
   orden del filtro.


   pesos=[delta2/delta1 1];
   n=[1:1:M];
   f=[0 wp ws 1];
   m=[1 1 0 0];
   h=firpm(M-1,f,m,pesos);%especificaciones para filtro paso-bajas.
```

A.2| Descripción de hardware en VHDL.

cic.vhdl Filtro CIC en VHDL N=30, k=2.

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

        entity CIC is
        PORT(clk: IN STD_LOGIC;
                enable: IN STD_LOGIC;
                clk2: OUT STD_LOGIC;
                 xn: IN STD_LOGIC_VECTOR(15 DOWNTO 0);
                 yn: OUT STD_LOGIC_VECTOR(15 DOWNTO 0)
        );
        end CIC;


        architecture Behavioral of CIC is

        SIGNAL x: STD_LOGIC_VECTOR(15 DOWNTO 0):="0000000000000000";
        SIGNAL x26: STD_LOGIC_VECTOR(25 DOWNTO 0):=(OTHERS=>'0'); --version de la señal en 26 bits.
        SIGNAL i0,i1: STD_LOGIC_VECTOR(25 DOWNTO 0):=(OTHERS=>'0');--Integradores 0 y 1.
        --Mas etapas  de integradores (K) requerirían registros i2, i3, i4 ,i(K-1)
        SIGNAL  c0:  STD_LOGIC_VECTOR(25  DOWNTO  0):=(OTHERS=>'0');--acumulador  de  entrada  (i.e.
        entrada a la parte comb).
        SIGNAL c0r1, c1: STD_LOGIC_VECTOR(25 DOWNTO 0):=(OTHERS=>'0');--Primer filtro comb
        --Mas retardos en el comb requeririan, c0r2, c0r3, c0rM (i.e. acumulador de entrada c0 rettrazado 2, 3
        ...n muestras)
        SIGNAL c1r1, c2: STD_LOGIC_VECTOR(25 DOWNTO 0):=(OTHERS=>'0');--Segundo filtro comb.
        --Mas etapas comb (K) requeririan un registro c3, c4...c(K)  y sus correspondientes registros para
        retardos.


        TYPE estado IS(idle,espera,muestrea);--estados para la decimación.
        SIGNAL edo : estado;

        --contadores
        SIGNAL cnt : INTEGER  RANGE 0 to 29:=29;-- Contador para factor de diezmado( debe iniciar en) el maximo
        para tomar la primera muestra
        SIGNAL ci : INTEGER  RANGE 0 to 4;--contador para inicio de integradores.
        SIGNAL cd : INTEGER  RANGE 0 to 3;-- contador para salida del filtro CIC
        SIGNAL en : STD_LOGIC;
```

```
begin
process(clk)--Maquina de estados. para diezmado/ indicar existen salidas
begin
if(clk'event AND clk='1')then
   if(en='1')then
        if(ci=3)then --muestras listas para integradores
              if(cnt=29)then --factor de decimación
                  cnt<=0;
                  edo<=muestrea;
                    if(cd=3) then—muestra lista a la salida del filtro
                      clk2<='1';
                      cd<=cd;
                     else
                          cd<=cd+1
                    end if;--cd
             else
                cnt<=cnt+1;
                edo<=espera;
                 clk2<='0';
             end if;--cnt
        else
         ci<=ci+1;
        end if;--ci
   end if; --if enable
end if;--if clk
end process;

process (xn,clk)--convertir la entrada de 8 a 26 bits bit extension
begin
if(clk'event AND clk='1')THEN
   if(en='1')then
      x26(15 DOWNTO 0)<=x;--coloca la entrada en los primeros 15 bits.
      for k in  25 DOWNTO 16 loop
        x26(k)<=x(15);--repite el ultimo bit para conservar el signo.
      end loop;
   end if;
end if;
end process;

process (clk)--se enciende un ciclo despues de recibir la señal de encendido
begin
if(clk'event AND clk='1')
              if(enable='1')then
                  en<='1';
              else
                 en<='0';
              end if;
end if;
end process;
```

```
        process(clk)—parte de integradores.
        begin

                if(clk'event AND clk='1')then
          if(en='1')then
            x<=xn;--carga entrada en acumulador.
            i0<=i0+x26;--integrador 1
            i1<=i1+i0;--integrador 2
         end if;
                end if;
        end process;

        process(clk)--  parte comb
        begin
          if(clk='1'AND clk'event)then
                    if(en='1')then
                      if(edo=muestrea)then
                      c0<=i1;-- (ultimo integrador se asigna a acumulador de entrada c0)
                      --Primer comb
                      c0r1<=c0;
                      c1<=c0-c0r1
                      --Segundo comb
                      c1r1<=c1;
                      c2<=c1-c1r1;
                      end if;
                   end if;
          end if;
        end process;

        --salida del filtro
        yn<=c2(22 DOWNTO 7);
end Behavioral;
```

fir.vhdl Filtro FIR en VHDL 16 coeficientes, estructura de fase lineal.

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

entity FIR is
        PORT( clk: in STD_LOGIC;
        clk2: in STD_LOGIC;-- indicador de muestra de entrada
        clk3: out STD_LOGIC;--indicador de muestra de salida
        s: out STD_LOGIC_VECTOR(31 DOWNTO 0);
        xn: in  STD_LOGIC_VECTOR(15 DOWNTO 0)
        );
end FIR;
```

```
architecture Behavioral of FIR is

TYPE retardos IS ARRAY (1 TO 15) OF SIGNED(15 DOWNTO 0);
TYPE factores IS ARRAY (1 TO 8) OF SIGNED(15 DOWNTO 0);
TYPE productos IS ARRAY (1 TO 8) OF SIGNED(31 DOWNTO 0);



SIGNAL xnr:
retardos:=((OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),
(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),
(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'));

SIGNAL sumas:
factores:=((OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),
(OTHERS=>'0'),(OTHERS=>'0'));

--20 Mhz B=50khz.
SIGNAL coef:
factores:=("1111111111111111","1111111111111101","1111111111111110","1111111111111111","000
0000000000011","0000000000000111","0000000000001100","0000000000001111");

SIGNAL prod:
productos:=((OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(
OTHERS=>'0'),(OTHERS=>'0'));

SIGNAL di: INTEGER RANGE 0 TO 3;
SIGNAL dn: INTEGER RANGE 0 TO 3;
begin

process(clk2)--retrazos
begin
if(clk2'event and clk2='1')then
   xnr(1)<=signed(xn);
   for k IN 2 TO 15 loop
         xnr(k)<=xnr(k-1);
   end loop;
end if;
end process;

process(clk2)--sumas parciales.
begin
if(clk2'event and clk2='1')then
   sumas(1)<=signed(xn)+xnr(15);
   for k IN 2 TO 8 LOOP
         sumas(k)<=xnr(k-1)+xnr(16-k);
   end loop;
end if;
end process;
```

```
process(clk)--calculo de productos
begin
if(clk'event and clk='1')then
for k IN 1 TO 8 LOOP
        prod(k)<=sumas(k)*coef(k);
end loop;
end if;
end process;

process(clk2)--Suma de productos
begin
 if(clk2'event and clk2='1')then
        if(di=1)then
         s<=prod(1)+prod(2)+prod(3)+prod(4)+prod(5)+prod(6)+prod(7)+prod(8);
        else
          di<=di+1;
        end if;
        if(dn=2)then--delay de notificacion
           dn<=dn;
        else
        dn<=dn+1;
        end if;
end if;
end process;


process(dn,clk2,clk)
begin
  if(dn=2)then
    clk3<=clk2;
  else
    clk3<='0';
end if;
end process;
end Behavioral;
```