



INAOE

**Métodos eficientes para Mejorar
Decimadores Comb para
Convertidores A-D Sigma-Delta
Basados en Filtros Coseno y
Compensadores**

por

Angel Garcia Robles

Tesis sometida como requisito parcial
para obtener el grado de

**MAESTRO EN CIENCIAS EN LA
ESPECIALIDAD DE ELECTRÓNICA**

en el

**Instituto Nacional de Astrofísica,
Óptica y Electrónica (INAOE)**

Septiembre de 2015

Santa María Tonantzintla, Puebla

Supervisada por:

Dra. Gordana Jovanovic Dolecek, INAOE

©INAOE 2015

Derechos reservados

El autor otorga al INAOE el permiso de reproducir y
distribuir copias de esta tesis en su totalidad o en partes.



Resumen

En esta tesis se presenta una propuesta para el diseño de decimadores de baja complejidad basados filtros comb y coseno. Asimismo, se proponen arquitecturas eficientes en consumo de potencia y área utilizada, adecuadas para convertidores analógico-digital sigma delta.

Inicialmente se presentan los fundamentos del proceso de decimación, haciendo énfasis en la importancia del filtro antialiasing. Después se describe el filtro comb, el cual es considerado como el filtro antialiasing más simple. Antes de dar paso a la propuesta de este trabajo, se hace una revisión de algunos métodos enfocados a mejorar la respuesta de magnitud del filtro comb.

Posteriormente, se aborda la contribución de este trabajo, la cual consiste en mejorar la respuesta de magnitud del filtro comb, pero manteniendo su complejidad en niveles bajos. Ello se consigue mediante el uso de una estructura de dos etapas, la inserción eficiente de filtros coseno, y la conexión de un compensador muy simple.

Finalmente, se evalúa el desempeño de los decimadores propuestos a través de comparaciones con algunos métodos existentes. Del mismo modo, los decimadores propuestos se implementan en un FPGA para verificar su correcto funcionamiento y determinar el área utilizada y la potencia consumida.

Abstract

This thesis presents a methodology to design low complexity comb-cosine based decimators. Additionally, they are proposed efficient architectures in both power consumption and used area, which are suitable for sigma-delta analog-to-digital converters.

Initially, it is presented the fundamental theory of decimation process. Emphasis is placed on anti-aliasing filter. Then, the comb filter is described, which is the simplest antialiasing filter. Afterward, it is given a brief review of some methods focused on improving the magnitude response of the comb filter.

Subsequently, the contribution of this work is addressed. The goal is to improve the magnitude response of the comb filter, but keeping its low complexity. This is achieved by using a two-stage structure, the effective insertion of cosine filters, and the connection of a simple compensator.

Finally, the performance of proposed decimators is evaluated. Several comparisons with some existing methods are provided. Similarly, the FPGA implementation of the proposed decimators is described, to verify its proper operation and to determine the used area and power consumption.

Agradecimientos

A la Dra. Gordana Jovanovic Dolecek, mi asesora de tesis, por darme la oportunidad de trabajar en su grupo. Agradezco su invaluable apoyo, su enorme paciencia, y todas las enseñanzas que me brindó durante el desarrollo de este trabajo.

A mis padres, por sus valiosos consejos. Porque su buen ejemplo me motiva a persistir en todo momento. A mis hermanos, quienes me impulsan día con día a ser una mejor persona.

A mis sinodales, por el tiempo que ocuparon en la revisión de este trabajo.

Al grupo DSP-INAOE, Ricardo, Gerardo, Miriam, en quienes siempre encontré disposición y confianza para inquirir en la búsqueda del conocimiento. También al Dr. David E. Troncoso, por la sabiduría que compartió conmigo y por enseñarme que la ciencia es realmente maravillosa.

Al INAOE y a CONACyT, por las facilidades y el soporte proporcionados durante mis estudios.

A mis amigos y compañeros, por esos ratos de buen humor que le sumaron a mi estancia en el INAOE.

Índice general

Resumen.....	III
Abstract.....	V
Agradecimientos.....	VII
Índice general	IX
Prefacio	XIII
1. Introducción	1
1.1 Decimación.....	1
1.1.1 Submuestreo	2
1.1.2 Filtro antialiasing	3
1.1.3 Decimación multietapa	4
1.2 Convertidor Analógico-Digital Sigma Delta.....	5
1.2.1 Modulador Sigma-Delta	5
1.2.2 Decimador.....	8
2. Filtro Comb	9
2.1 Función de transferencia.....	9
2.2 Respuesta en frecuencia	10
2.3 Uso del filtro comb en el proceso de decimación.....	12
2.4 Estructura CIC.....	14
2.5 Estructura no recursiva	16
2.5.1 Estructura multi-etapas.....	16
2.5.2 Decimación polifásica.....	18
3. Métodos para Mejorar la Respuesta de Magnitud del Filtro Comb	21
3.1 Métodos basados en el filtro coseno para el incremento de atenuación entre las bandas de rechazo	21
3.1.1 Filtro Coseno	21
3.1.2 Pre-filtro Coseno [17].....	24

3.1.3 Filtro coseno rotado [20].....	27
3.1.4 Inserción de dos filtro coseno [21].....	29
3.2 Métodos para la disminución de la caída en banda de paso	31
3.2.1 Filtro compensador de segundo orden basado en un enfoque trigonométrico [22].....	31
3.2.2 Filtro compensador basado en el criterio de error minimax [23].....	33
3.2.2 Filtro compensador basado en la transformación de amplitud del filtro coseno cuadrado [24]	34
4. Propuesta para Mejorar el Decimador Comb Basada en Filtros Coseno y Compensadores	39
4.1 Incremento de la atenuación entre las bandas de rechazo	39
4.1.1 Estructura comb de dos etapas.....	39
4.1.2 Inserción de dos filtros coseno en la segunda etapa de decimación.....	42
4.1.3 Filtro propuesto para el filtrado entre las bandas de rechazo.....	43
4.1.4 Arquitecturas eficientes	45
4.2 Disminución de la caída en la banda de paso	56
4.3 Decimador propuesto	58
4.3.1 Comparación con otros métodos.....	60
5. Implementación y resultados	63
5.1 Descripción en VHDL del decimador propuesto	63
5.1.1 Arquitectura para M es el producto de dos enteros.....	64
5.1.2 Arquitectura para M es un número potencia de dos.....	69
5.1.3 Descripción del filtro compensador	70
5.2 Simulación y resultados de implementación	70
Conclusiones	75
Trabajo a futuro.....	76
Anexos	77
1. Códigos en VHDL	77
a) Descripción en VHDL de la arquitectura del demultiplexor registrado	77

b) Descripción en VHDL de la arquitectura de los bloques generadores de multiplicaciones para los filtros polifásicos del ejemplo 5.1	77
c) Descripción e VHDL de sumadores y retrasos del filtro del ejemplo 5.1	78
d) Descripción e VHDL de la arquitectura de la sección de integradores	79
e) Descripción en VHDL de la arquitectura del demultiplexor correspondiente a la segunda etapa de decimación	80
f) Descripción en VHDL de la arquitectura de los sumadores y registros de los filtros coseno.....	81
g) Descripción en VHDL de la arquitectura del filtro compensador.....	82
2. Artículos Adjuntos.....	83
a) “Using Cosine Filters to Improve Alias Rejection in Comb Decimation Filter”	83
b) “Polyphase Decomposition of Non-Recursive Comb-Cosine Decimation Filter for SDR Applications”	87
c) “FPGA Implementation of Comb-Based Decimation Filter with Improved Frequency Characteristic for SD A/D Converters Application”	91
Lista de figuras.....	97
Lista de tablas.....	99
Referencias	100

Prefacio

Actualmente, la decimación juega un papel muy importante en diversos sistemas electrónicos. Se denomina decimación al proceso mediante el cual se reduce la frecuencia de muestreo de una señal por un número entero. La decimación es ampliamente utilizada en aplicaciones como: Radio Definido por Software, convertidores analógico digital sigma delta, dispositivos médicos, entre otras.

En esencia, se necesitan dos bloques digitales para llevar a cabo la decimación: un filtro pasabajas y un submuestreador. Cuando una señal se submuestra aparecen réplicas de su espectro original. Tales réplicas se pueden superponer con el espectro original y ocasionar una irreparable distorsión en la señal, denominada aliasing. Por consiguiente, es necesario utilizar un filtro pasabajas antes efectuar el submuestreo. Este filtro también es conocido como filtro antialiasing, y tiene la finalidad de limitar el ancho de banda de la señal, previniendo con ello el efecto de aliasing.

En general, las especificaciones de un filtro antialiasing son muy severas, ya que se requiere una mínima distorsión en la señal que se decima. En muchos casos, tales especificaciones conducen a un filtro digital muy complejo. Una práctica común para disminuir la complejidad del filtro antialiasing es dividir el proceso de decimación en múltiples etapas, con lo que se pueden satisfacer las especificaciones mediante filtros más sencillos. De este modo, en la primera etapa cuya frecuencia de operación es elevada, resulta conveniente utilizar un filtro muy simple conocido como filtro comb. Lo anterior implica que el filtro de la última etapa debe ser muy selectivo, no obstante es posible realizarlo en un filtro de complejidad relativamente baja, y además su frecuencia de operación es la menor.

La principal ventaja que ofrece el filtro comb es su estructura muy simple que no requiere multiplicadores. Sin embargo, su respuesta de magnitud exhibe algunos inconvenientes que pueden deteriorar a la señal que se decima. Específicamente, presenta un reducido rechazo antialiasing, el cual se puede mejorar al conectar en cascada K filtros comb. Desafortunadamente, al hacer esto la desviación en la banda de paso sufre un aumento considerable.

Se han propuesto diversas metodologías para mejorar la respuesta de magnitud del filtro comb. Sin embargo, algunas requieren la adición de multiplicadores. En cambio, otras incrementan desmedidamente el uso de sumadores y registros. Esto no es muy conveniente ya que deriva en un aumento del área de chip y de la potencia consumida.

El objetivo de esta tesis es mejorar los decimadores basados en el filtro comb como se describe a continuación. En primer lugar se busca incrementar la atenuación entre las bandas de rechazo y disminuir la caída en banda de paso con un mínimo incremento de recursos utilizados. De igual forma, se averigua la manera de disminuir su potencia consumida y área utilizada. Una vez logrado esto, se presenta una metodología para el diseño de decimadores de baja complejidad y eficientes en área utilizada y consumo de potencia.

Con el fin de lograr el objetivo planteado, en primer lugar se describe el proceso de decimación. Después, en el capítulo 2 se presenta el filtro antialiasing más simple, conocido como filtro comb. En el capítulo 3 se hace una revisión de los métodos enfocados en mejorar la respuesta de magnitud del filtro comb. Particularmente, se estudian los métodos basados en el filtro coseno.

A partir de los conceptos presentados en los tres primeros capítulos, en el capítulo 4 se presenta la propuesta para mejorar el decimador basado en el filtro comb.

Finalmente, en el capítulo 5 se presentan los resultados obtenidos de las implementaciones realizadas en FPGA.

Capítulo 1

Introducción

Durante los últimos años, la tendencia en el procesamiento de señales se ha dirigido hacia el empleo de técnicas digitales, ya que ofrecen múltiples ventajas en comparación con el procesamiento analógico. Por un lado, los circuitos digitales son robustos y se pueden realizar mediante estructuras simples y extremadamente pequeñas, que a su vez pueden combinarse fácilmente para obtener sistemas complejos, precisos y rápidos. Además, ya que los dispositivos digitales dominan el mercado de la electrónica, el desarrollo de la tecnología de integración se optimiza para satisfacer los requerimientos de la electrónica digital.

En este capítulo se aborda un proceso que se utiliza frecuentemente para disminuir la tasa de muestreo en forma digital. Se describe mediante sus dos bloques constitutivos. Posteriormente se aborda el esquema multietapas como una forma eficiente de llevar a cabo tal proceso. Finalmente, se describe el convertidor analógico digital sigma delta, haciendo énfasis en las características del decimador que requiere.

1.1 Decimación

La decimación es un proceso que consiste en disminuir la frecuencia de muestreo de una señal por un número entero, el cual es conocido como factor de submuestreo y es denotado por M . El proceso de decimación se efectúa por un sistema denominado decimador. Fundamentalmente, el decimador está constituido por dos bloques: un filtro pasabajas y un submuestreador. Su diagrama se ilustra en la Figura 1.1.

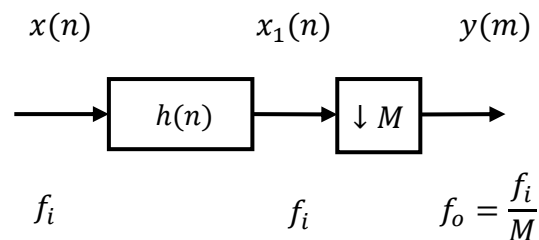


Figura 1.1 Diagrama típico de un decimador.

1.1.1 Submuestreo

El bloque submuestreador tiene la finalidad de disminuir la frecuencia de muestreo de la señal. Esto se lleva a cabo conservando las muestras que son múltiplos enteros del factor de submuestreo, y descartando las otras muestras. En el dominio del tiempo discreto, la expresión que relaciona la salida $y(m)$ con la entrada $x(n)$ del submuestreador es la siguiente:

$$y(m) = x(mM). \quad (1.1)$$

Ya que algunas muestras de la señal original se eliminan, la operación de submuestreo es irreversible. Es decir, no se puede recuperar $x(n)$ de $y(m)$ de manera exacta, solamente se puede obtener una aproximación de $x(n)$.

Por su parte, en el dominio de la frecuencia, el espectro de la señal de salida se relaciona con el espectro de la señal de entrada por medio de la siguiente expresión [1]:

$$Y(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{j\frac{(\omega-2\pi k)}{M}}\right), \quad (1.2)$$

A partir de la expresión anterior, se deduce que al submuestrear una señal su espectro se escala en amplitud por un factor $\frac{1}{M}$, se expande en frecuencia M -veces y aparecen réplicas de este espectro centradas en múltiplos de 2π . Debido a que la señal que se submuestra es de carácter digital, su espectro no está completamente limitado en banda. De este modo, algunos intervalos de las réplicas se superponen con el espectro original. Si la magnitud de estas porciones de espectro no es lo suficientemente pequeña, entonces la señal original sufrirá una distorsión irreversible, denominada aliasing. El efecto de aliasing debe ser evitado a toda costa, ya que degrada la información que lleva la señal. En la Figura 1.2 se ilustra el efecto de aliasing. Supóngase que $x(n)$ es una señal cuyo espectro es representado en la Figura 1.2(a). Después de submuestrear la señal, su espectro se ve afectado de tres maneras: se comprime en amplitud, se expande en frecuencia y aparecen réplicas, como se observa en la Figura 1.2(b). Es fácil advertir que algunas componentes de las réplicas se traslapan con el espectro original y lo distorsionan.

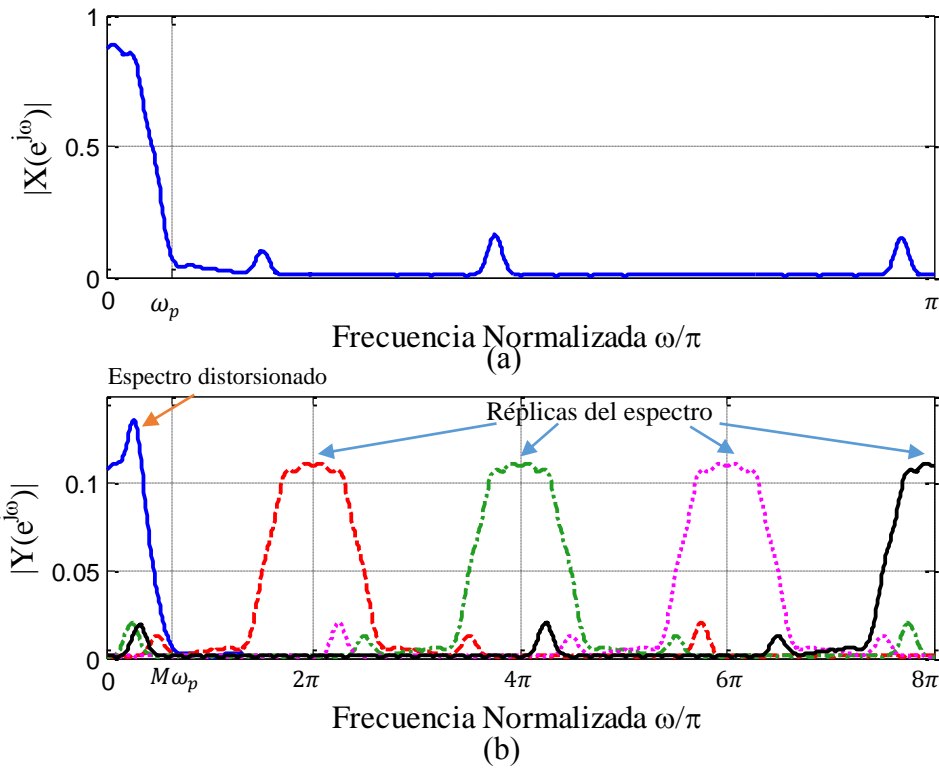


Figura 1.2 Ilustración en el dominio de la frecuencia de los efectos del submuestreo. a) Espectro original de la señal. b) Espectro de la señal submuestreada.

1.1.2 Filtro antialiasing

Para prevenir el efecto de aliasing es necesario utilizar un filtro pasabajas antes de llevar a cabo el submuestreo. A menudo tal filtro es llamado filtro antialiasing. Su propósito es limitar el espectro de la señal, de tal manera que al submuestrearla sus réplicas no causen distorsión. Idealmente, las especificaciones de este filtro están dadas por [2]:

$$|H(e^{j\omega})| = \begin{cases} 1, & \omega \leq \omega_c \\ 0, & \frac{\pi}{M} \leq \omega \leq \pi \end{cases}, \quad (1.3)$$

donde ω_c representa la frecuencia más alta que se debe preservar de la señal que se decima. Dado que las especificaciones ideales son muy difíciles de conseguir mediante filtros realizables, en la práctica se proporcionan ciertas tolerancias aceptables. Generalmente, se dan en términos de una desviación máxima en la banda de paso y una atenuación mínima en la banda supresora. Éstas dependen del grado de distorsión que se

puede tolerar en determinada aplicación. Cabe recordar que a mayor rigurosidad de las especificaciones, mayor es la complejidad de un filtro digital.

1.1.3 Decimación multietapa

El decimador de la Figura 1.1 es una estructura de una sola etapa, pues utiliza un solo filtro antialiasing y un solo submuestreador. En términos generales, la complejidad de un filtro digital mantiene relación inversa con el ancho de su banda de transición. Al aumentar el factor de submuestreo se requieren filtros con banda de transición cada vez más estrecha, implicando un aumento considerable en su complejidad. Una opción para disminuir la complejidad del filtro antialiasing es mediante el esquema multietapa [3], siendo posible cuando el factor de submuestreo se puede factorizar como el producto de números enteros, esto es $M = M_1 \cdot M_2 \cdot M_k$. A partir de la factorización de M , el decimador se puede realizar en una conexión en cascada de k etapas, tal como se aprecia en la Figura 1.3. Al hacer esto, las especificaciones de los filtros cada etapa son menos rigurosas, y con ello la complejidad total se reduce significativamente.

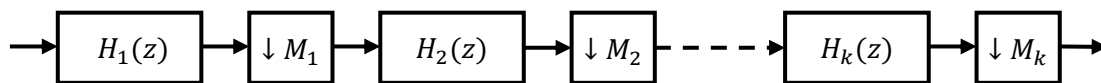


Figura 1.3 Decimador multietapa.

En muchos casos es muy práctico considerar un esquema de dos etapas. Bajo tal consideración, los filtros antialiasing en cada una presentan propiedades diferentes. En cuanto al filtro de la primera etapa, éste debe exhibir una gran atenuación dentro de determinados intervalos frecuenciales conocidos como bandas de supresión. Estas bandas se encuentran centradas en múltiplos de la frecuencia $\frac{2\pi}{M}$ y tienen un espesor que es igual al doble del ancho de banda de la señal que se desea preservar, ω_c . Fuera de estos intervalos, la atenuación exhibida por el filtro es irrelevante, ya que en la segunda etapa de decimación, el filtro antialiasing es el encargado de proporcionar la atenuación deseada.

Un filtro que se adapta de manera adecuada a las especificaciones requeridas en la primera etapa de decimación es el filtro comb, ya que sus que sus ceros están localizados justo en el centro de las bandas de supresión. Adicionalmente, este filtro es muy simple y no requiere multiplicadores. Sin embargo, su respuesta de magnitud dista de ser la

ideal. A pesar de esto, las ventajas que ofrece el filtro son mayores en relación con sus desventajas, por lo que en la mayoría de las aplicaciones el filtro comb es el más conveniente para la primera etapa de decimación. Por tales razones, vale la pena el desarrollo de metodologías enfocadas a mejorar cualquier aspecto del filtro, ya sea su respuesta de magnitud o su arquitectura.

1.2 Convertidor Analógico-Digital Sigma Delta

El convertidor analógico-digital sigma delta (SD-ADC, por sus siglas en inglés) es un dispositivo que utiliza un decimador en su estructura. En la Figura 1.4 se presenta su diagrama a bloques. Básicamente está compuesto de dos partes: la primera es de carácter analógico y se denomina modulador Sigma Delta (MSD), mientras que la segunda es un decimador.

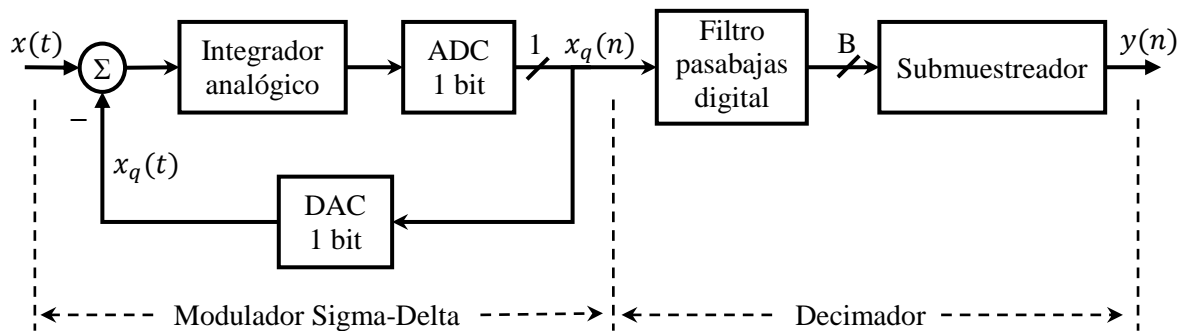


Figura 1.4 Diagrama a bloques del SD-ADC.

1.2.1 Modulador Sigma-Delta

El MSD es la interfaz entre el dominio analógico y el dominio digital. Fue desarrollado como una extensión del modulador delta [4]. El MSD aprovecha las técnicas de sobremuestreo y modelado de ruido para disminuir el ruido de cuantización dentro de la banda de interés de la señal.

La operación de muestrear una señal con una frecuencia mayor que la tasa de Nyquist, f_N , se denomina sobremuestreo. Siendo la razón de sobremuestreo igual a:

$$OSR = \frac{f_s}{f_N}, \quad (1.4)$$

donde f_s es la frecuencia de muestreo.

Las ventajas que ofrece el sobremuestreo se enuncian a continuación [5]:

- Disminución del ruido de cuantización en la banda de la señal. El efecto del ruido de cuantización sobre la señal muestreada se evalúa a través de la razón señal a ruido de cuantización ($SNRQ$), definida como la relación entre la potencia de una señal senoidal y el ruido de cuantización en la banda de interés.

$$SNRQ = 6.02 \cdot ENOB + 10 \log_{10}(OSR) + 1.76, \quad (1.5)$$

donde $ENOB$ es el número efectivo de bits del ADC. Se observa que al doblar la OSR , la $SNRQ$ aumenta 3 dB, equivalente a incrementar la resolución del convertidor en medio bit.

- Reducción de la complejidad del filtro antialiasing analógico. Por razones prácticas, todos los convertidores analógico digital emplean un filtro antes de muestrear la señal. Comúnmente, dicho filtro recibe el nombre de filtro antialiasing analógico. Su complejidad depende de manera inversa con la proporción entre la banda de rechazo y la banda de paso [6]. Al aumentar la frecuencia de muestreo, esta proporción también incrementa, y consecuentemente disminuye la complejidad de filtro. De modo que si la OSR es muy grande, el filtro antialiasing analógico se puede realizar con componentes analógicos de baja precisión y de menor costo.
- Al incrementar la frecuencia de muestreo, el número de niveles de cuantización puede ser menor, sin que exista pérdida de información. Generalmente, un ADC de sobremuestreo efectúa la cuantización en dos pasos. Primero la señal se sobremuestra, posibilitando su cuantización con un número reducido de bits (frecuentemente mediante un comparador de 1 bit). Posteriormente, dicha versión toscamente cuantizada se decima a la frecuencia de Nyquist consiguiendo mayor precisión en la cuantización.

El modelado de ruido permite un mejoramiento al desempeño del ADC. Este principio se ilustra en el dominio de tiempo discreto, dicho análisis es apropiado ya que generalmente los SDM se implementan con tecnología de capacitores conmutados. El diagrama equivalente en tiempo discreto del SDM se muestra en la Figura 1.5. El sistema tiene dos entradas, $X(z)$ representa la señal de entrada y $E(z)$ representa el ruido de cuantización, y una salida $Y(z)$.

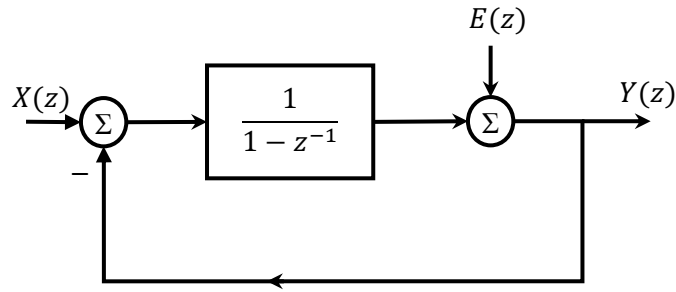


Figura 1.5 Modelo del Modulador Sigma Delta en tiempo discreto.

De manera que se pueden hallar dos funciones de transferencia, una con respecto a la entrada $STF(z)$, y la otra con respecto al ruido $NTF(z)$, dadas por:

$$STF(z) = z^{-1}. \quad (1.6)$$

$$NTF(z) = 1 - z^{-1}. \quad (1.7)$$

La NTF del modulador sigma delta representa un filtro pasaaltas, y por lo tanto el ruido de cuantización se mapea hacia altas frecuencias. En la Figura 1.6 se ilustran las funciones de transferencia del ruido no modulado y con modulación de primer y segundo orden. El orden se refiere al número de acumuladores en el cuantizador. Se observa que al incrementar el orden, la potencia en la banda de interés disminuye y la potencia del ruido fuera de la banda aumenta. A pesar de que la potencia del ruido de cuantización sufre un incremento en el intervalo fuera de banda, no impacta a la señal ya que el ruido se puede remover por medio de un filtro digital pasabajas, sin que afecte a la señal de entrada. Tal filtro digital es parte del proceso de decimación que sigue al SDM.

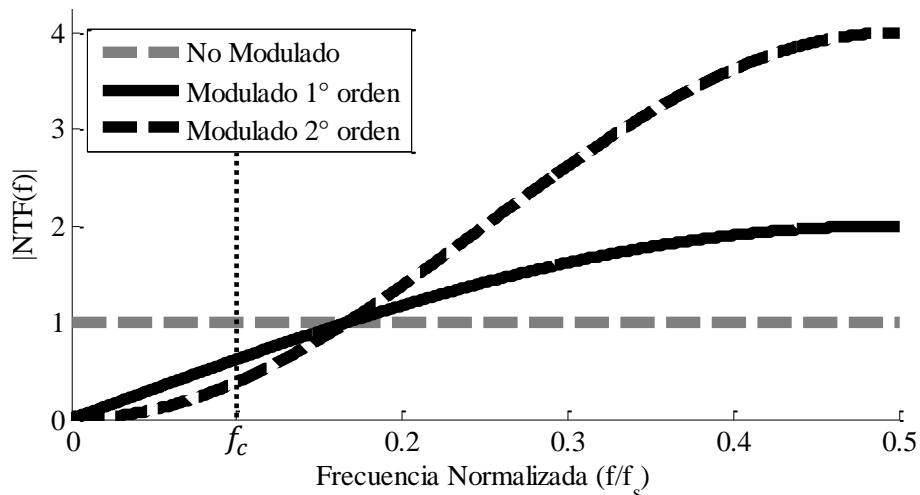


Figura 1.6 Gráficas de la función de transferencia de ruido de un SDM.

1.2.2 Decimador

La segunda parte que conforma a un SD-ADC es un decimador.

La finalidad del bloque decimador utilizado en el SD-ADC se resume en los siguientes tres puntos:

- Remover el ruido de cuantización. Como ya se puntualizó en la sección 1.2.1, el SDM traslada la mayor parte del ruido de cuantización hacia altas frecuencias, y solo una pequeña porción se queda dentro de la banda de la señal. Así, el principal objetivo del filtro pasabajas que compone al decimador es eliminar el ruido de cuantización que se encuentra fuera de la banda de la señal. Ya que de lo contrario, al submuestrear la señal, dicho ruido ocasionaría aliasing.
- Incrementar la resolución efectiva de la señal digital. La salida del SDM es una cadena de muestras toscamente cuantizadas de la señal analógica, generalmente de un bit de resolución. Una mayor resolución se logra al promediar un gran número de muestras. El filtrado pasabajas es equivalente a la operación de promediado. Así, la reducción del ruido de cuantización es equivalente a un incremento en la resolución efectiva de la señal digital.
- Reducción de la frecuencia de muestreo. La salida del SDM se encuentra a muy alta frecuencia de muestreo. Una vez que el ruido de cuantización de alta frecuencia ha sido atenuado, es posible reducir la frecuencia de muestreo. Es deseable que la frecuencia de muestreo se reduzca a la tasa de Nyquist, ya que con ello se minimiza la cantidad de información para llevar a cabo alguna de las siguientes tareas: transmisión, almacenamiento o procesamiento.

Capítulo 2

Filtro Comb

El filtro comb posee características que lo posicionan como el filtro antialiasing más popular y simple. Tiene respuesta de magnitud pasabajas y fase lineal, que se consiguen mediante una estructura de pocos recursos y que no requiere multiplicadores. En consecuencia, este filtro puede operar a frecuencias altas. Adicionalmente, tanto su diseño como su implementación se realizan de manera sencilla.

En este capítulo se presenta el filtro comb a través de su función de transferencia y de sus características en frecuencia. Se discuten sus propiedades al ser utilizado como filtro antialiasing en el proceso de decimación. Asimismo, se analiza la implementación de sus dos realizaciones: recursiva y no recursiva.

2.1 Función de transferencia

El filtro comb es el filtro pasabajas más simple [7]. Tiene respuesta al impulso finita (FIR) y se caracteriza por la siguiente función de transferencia:

$$H(z) = \left[\frac{1}{M} \frac{1 - z^{-M}}{1 - z^{-1}} \right]^K = \left[\frac{1}{M} \sum_{i=0}^{M-1} z^{-i} \right]^K, \quad (2.1)$$

donde el parámetro K se conoce como orden del filtro e indica el número de filtros comb conectados en cascada, y el parámetro M es la longitud del filtro o el factor de decimación. En las funciones de sistema dadas por 2.1, la forma recursiva del filtro comb corresponde a la expresión que es el cociente de dos binomios. Alternativamente, la expresión equivalente que contiene la sumatoria de términos z^{-i} se conoce como forma no recursiva. Estas dos formas de la función de transferencia exhiben propiedades diferentes al ser implementadas, las cuales se abordarán más adelante en las Secciones 2.4 y 2.5. De acuerdo con la forma recursiva, y teniendo en cuenta que el orden representa la multiplicidad de los polos y ceros, se deduce que la función de transferencia tiene KM ceros en el círculo unitario, localizados en $z = e^{j2\pi k/M}$, $k = 0, 1, \dots, M - 1$. También existen $K(M-1)$ polos en el origen, y un solo polo de

multiplicidad K en $z = 1$ que se cancela con el cero ubicado exactamente en el mismo lugar, dando como resultado un filtro estable, como se aprecia en su forma no recursiva. Con fines de ejemplificación, en la Figura 2.1 se muestra el plano z con la ubicación de los polos y ceros de un filtro comb cuyos parámetros son $M=9$ y $K=3$.

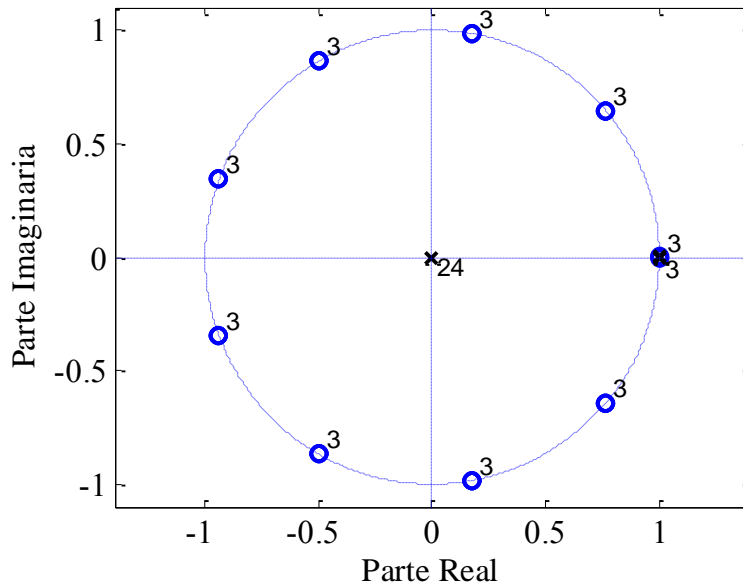


Figura 2.1 Localización de polos y ceros del filtro comb.

2.2 Respuesta en frecuencia

La respuesta en frecuencia del filtro comb se obtiene al evaluar la función de transferencia en el círculo unitario, mediante la sustitución $z = e^{j\omega}$ en (2.1), dando como resultado la siguiente expresión:

$$H(e^{j\omega}) = \left[\frac{1}{M} \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \cdot e^{-\frac{j\omega(M-1)}{2}} \right]^K. \quad (2.2)$$

La respuesta de magnitud del filtro comb se deduce fácilmente de la expresión anterior, siendo definida por:

$$|H(e^{j\omega})| = \frac{1}{M^K} \left| \frac{\sin^K\left(\frac{\omega M}{2}\right)}{\sin^K\left(\frac{\omega}{2}\right)} \right|. \quad (2.3)$$

En la Figura 2.2 se muestran las gráficas de las respuestas de magnitud de tres filtros comb que tienen el mismo factor de decimación $M=9$, pero que tienen orden diferente. Nótese que en el intervalo $[0, \pi]$ existen $\left\lfloor \frac{M}{2} \right\rfloor$ ceros ubicados en las frecuencias que son múltiplos enteros de $\frac{2\pi}{M}$, cada uno de ellos tiene multiplicidad K . Al observar las gráficas de los filtros comb de orden superior, se advierte que un incremento en el orden del filtro resulta en una mayor atenuación. Sin embargo, al hacer esto, la desviación en la banda de paso también aumenta.

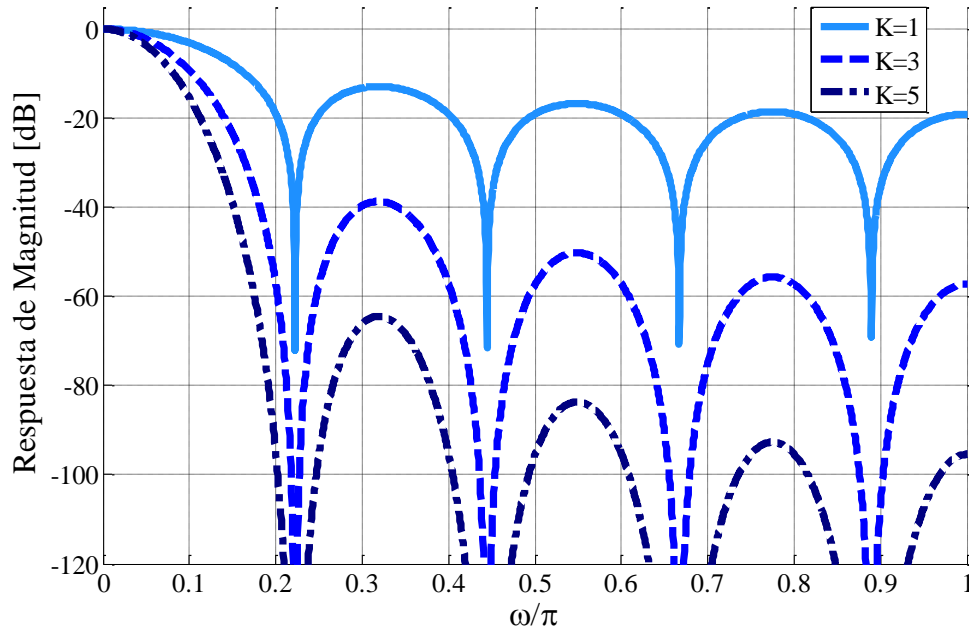


Figura 2.2 Respuestas de magnitud de filtros Comb de diferente orden.

La respuesta de fase del filtro comb se encuentra de (2.2) a partir del exponente del factor exponencial, siendo igual a:

$$\theta(\omega) = -\frac{K(M-1)}{2} \cdot \omega. \quad (2.4)$$

De la expresión anterior se observa que el filtro comb tiene fase lineal con retraso de grupo constante dado por la pendiente: $-K(M-1)/2$.

2.3 Uso del filtro comb en el proceso de decimación

Las características que presenta el filtro comb en su respuesta de magnitud, lo hacen apropiado para ser utilizado como filtro antialiasing en la primera etapa de un proceso de decimación. La propiedad que resulta importante para tal aplicación se enuncia como sigue: sus ceros se localizan justo en el centro de las bandas que se pliegan al submuestrear la señal. Cuando se usa en la primera etapa de decimación, el filtro presenta múltiples bandas de rechazo y una banda de paso. Éstas se definen a continuación:

- Banda de paso (BP) corresponde al intervalo de frecuencias donde el filtro debe dejar pasar íntegramente las componentes frecuenciales de la señal que se submuestra. Es dependiente del factor de decimación M y del factor de decimación residual, v . Tal intervalo está dado por:

$$BP = [0, \omega_p]; \text{ con } \omega_p = \frac{\pi}{Mv}, \quad (2.5)$$

donde ω_p se conoce como frecuencia de borde de banda de paso.

- Bandas de rechazo (BR) son intervalos de frecuencias que se mapean hacia la banda de paso como consecuencia del submuestreo. Por ello, la atenuación que el filtro debe proporcionar dentro de tales bandas es muy estricta. Estas bandas también son conocidas como folding bands y se ubican alrededor de los ceros del filtro comb, siendo definidas como sigue:

$$BR_i = \left[\frac{2\pi \cdot i}{M} - \omega_p, \frac{2\pi \cdot i}{M} + \omega_p \right]; \quad i = 1, 2, \dots, \left\lfloor \frac{M}{2} \right\rfloor. \quad (2.6)$$

Se han especificado algunas medidas para evaluar el desempeño del filtro comb [7] que se determinan a partir de su respuesta de magnitud. Éstas se consideran con fines de comparación y se enuncian a continuación:

- Desviación en la banda de paso (DBP) idealmente la respuesta de magnitud del filtro debe exhibir una característica plana en todo el intervalo de la banda de paso. Sin embargo, esto no es posible mediante filtros reales. La DBP indica la

máxima atenuación que sufre la respuesta de magnitud en la banda de paso. Se obtiene al evaluar la respuesta de magnitud en el borde de la banda de paso:

$$WCP = |H(e^{j\omega})|_{\omega=\omega_p} \quad (2.7)$$

- Peor caso de atenuación (PCA) representa la atenuación mínima conseguida por el filtro. Se obtiene al evaluar la respuesta de magnitud en el borde inferior de la primera folding band:

$$WCA = |H(e^{j\omega})|_{\omega=\frac{2\pi}{M}-\omega_p} \quad (2.8)$$

Las características descritas anteriormente se ilustran en la Figura 2.3. Se toma como ejemplo la magnitud de un filtro cuyos parámetros son $M=8$, $K=4$ y se considera el mínimo factor de decimación residual, esto es $v=2$.

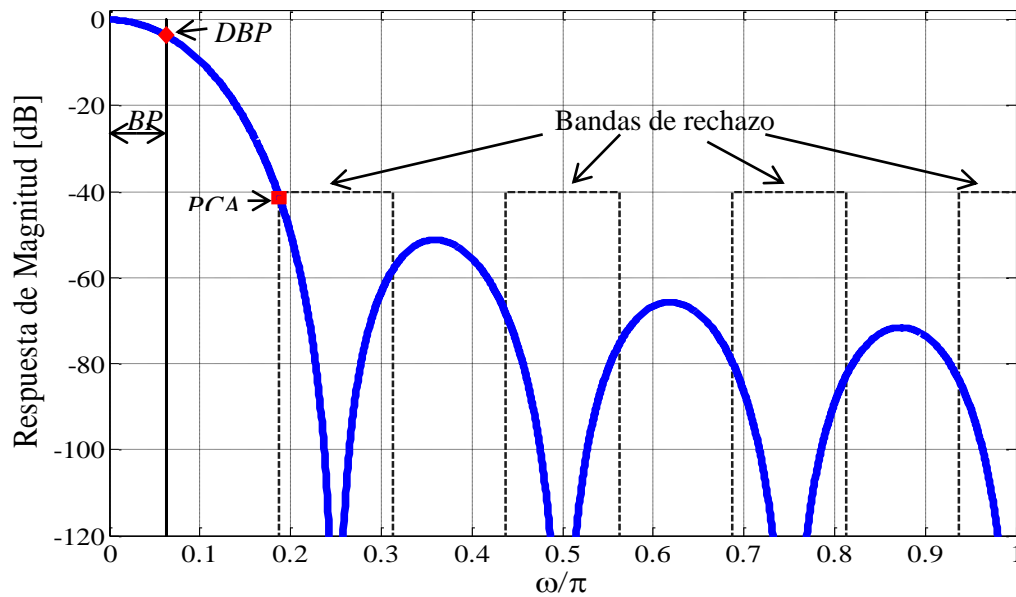


Figura 2.3 Características que exhibe el filtro comb al ser utilizado en el proceso de decimación.

De manera práctica, tanto el PCA como la DBP no dependen del factor de decimación sino solamente del orden del filtro comb. En la Tabla 2.1 se presenta esta correspondencia. Es interesante notar el impacto positivo que se presenta con el incremento del orden, cada vez que esto sucede, el PCA se incrementa poco más de 10

dB y también se incrementa la atenuación de todas las bandas de rechazo. En el lado negativo, se observa que la DBP sufre un aumento de 0.9 dB, por cada incremento del orden.

Parámetro independiente	Parámetros dependientes	
K	DBP (dB)	PCA (dB)
1	0.91	10.2
2	1.82	20.3
3	2.73	30.4
4	3.65	40.6
5	4.56	50.7
6	5.47	60.8

Tabla 2.1 Relación entre el orden del filtro comb, K , con la DBP y el PCA .

2.4 Estructura CIC

Hogenauer introdujo un decimador cuya estructura está basada en la forma recursiva del filtro comb [8]. Cuando el filtro comb es utilizado de esta manera es llamado CIC, ya que consiste de la conexión en cascada de integradores y diferenciadores. La implementación se realiza en dos secciones separadas por un bloque de reducción de frecuencia de muestreo. La primera sección funciona a la más alta frecuencia de muestreo, f_s . Esta sección se compone de K filtros integradores cuya función de transferencia es la siguiente:

$$H_I(z) = \left[\frac{1}{1 - z^{-1}} \right]^K. \quad (2.9)$$

Por otro lado, la segunda sección trabaja a la tasa de muestreo más baja, esto es f_s/M . Esta sección se conforma por K filtros diferenciadores. Su función de transferencia, referenciada a la frecuencia de muestreo más alta, está dada por:

$$H_D(z) = [1 - z^{-M}]^K. \quad (2.10)$$

En la Figura 2.4 se presenta la estructura de un decimador con filtro CIC de K etapas. Se observa que no se requieren multiplicadores y el número de registros y sumadores utilizados es muy bajo, dependiendo únicamente del orden del filtro comb. Esto conduce a una implementación reducida en recursos. En la práctica, el bloque reductor de frecuencia de muestreo se implementa a través de un interruptor que es habilitado cada M ciclos de reloj.

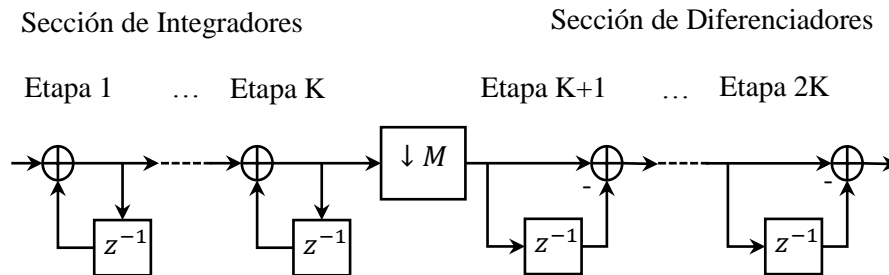


Figura 2.4 Estructura de un filtro decimador CIC de K etapas.

Existen restricciones que deben considerarse en la implementación de un filtro decimador CIC para evitar desbordamiento y asegurar su adecuado funcionamiento [9]. Lo primero que se debe tomar en cuenta es utilizar aritmética de complemento a dos en la implementación del filtro. La segunda restricción consiste en definir un ancho de bus mayor o igual que la máxima magnitud esperada a la salida del filtro. Este ancho de bus se debe aplicar en todas las etapas del filtro y se calcula mediante:

$$B_{max} = [K \log_2(M)] + B_{in}, \quad (2.11)$$

donde B_{in} es el ancho de bus en la entrada del filtro y $[x]$ representa la operación de redondeo hacia el entero inmediato menor que x .

Ahora se resumen las características de la estructura CIC, destacando sus ventajas:

- Implementación sencilla. Es muy fácil efectuar la implementación del filtro CIC de cualquier orden y factor de decimación. El orden define la estructura a realizar, es decir indica cuantos integradores y diferenciadores son necesarios. Únicamente se debe tomar en cuenta el apropiado ancho de bus, de acuerdo con 2.11, y aplicarlo a cada registro y sumador. Asimismo, el bloque submuestreador

se realiza como un interruptor controlado por un contador capaz de habilitarlo cada M ciclos de reloj.

- Reducida área de chip. En términos generales, ésta depende del número de recursos utilizados y de su ancho de bus. A pesar del gran ancho de bus que requiere cada elemento del filtro, únicamente son necesarios $2K$ sumadores y $2K$ registros, lográndose una implementación de área mínima.

Y considerando sus desventajas:

- La mayor desventaja del decimador CIC es su elevado consumo de potencia. La sección de integradores trabajando en alta frecuencia junto con el gran ancho de bus en toda la estructura son la causa de ello. Como consecuencia de esto, se evita utilizar la estructura CIC en aplicaciones que demandan un bajo consumo de potencia.

2.5 Estructura no recursiva

La forma no recursiva de la función de transferencia del filtro comb permite obtener estructuras decimadoras con un consumo de potencia menor que el disipado por la estructura CIC [10]. Además, tales estructuras no presentan problemas de desbordamiento. Enseguida se describen las dos configuraciones principales de la estructura no recursiva.

2.5.1 Estructura multi-etapas

El filtro comb no recursivo está compuesto por K filtros FIR de longitud M con coeficientes unitarios. Mitra, Tchobanov y Bryukhanov [11] han propuesto un método conocido como factorización polinomial, mediante el cual es posible factorizar un polinomio completo de coeficientes unitarios como el producto de polinomios racionales más simples. Tal método establece que si el factor de decimación del filtro comb es un número compuesto por L factores enteros, es decir $M = M_1 \cdot M_2 \cdot \dots \cdot M_L$, entonces el filtro se puede realizar en una estructura con L etapas comb más simples conectadas en cascada, teniéndose en la l -ésima etapa un factor de decimación M_l . Esta factorización se determina como sigue:

$$\left[\frac{1}{M} \sum_{i=0}^{M-1} z^{-i} \right]^K = \left[\prod_{l=1}^L \frac{1}{M_l} \left(\sum_{i=0}^{M_l-1} z^{-i \cdot \prod_{j=0}^{l-1} M_j} \right) \right]^K = \prod_{l=1}^L H_l(z^{\prod_{j=0}^{l-1} M_j}) \quad (2.12)$$

donde $M_0 = 1$. Un caso particular de esta factorización surge cuando el factor de decimación es una potencia de dos, $M = 2^p$. Se prefiere utilizar este caso, ya que evita la división por el factor de normalización. En la Figura 2.5 se ilustra el diagrama de bloques del filtro comb multietapas.

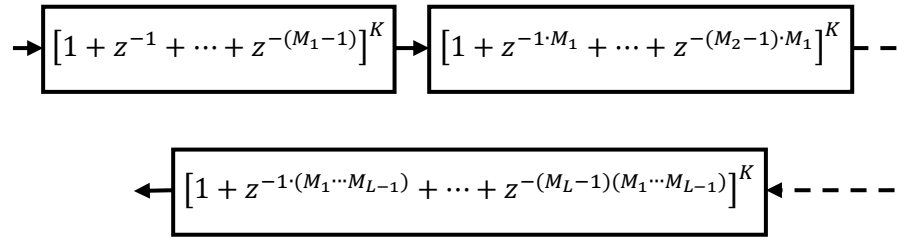


Figura 2.5 Diagrama de bloques de un filtro comb no recursivo multietapas.

La técnica anterior permite obtener decimadores multietapas, en los que existen L filtros comb trabajando a distintas frecuencias, como se observa en la Figura 2.6. Esto es posible al aplicar la tercera identidad noble. Nótese que únicamente el filtro de la primera etapa opera a la frecuencia más alta, mientras que los filtros de las etapas subsecuentes trabajan a frecuencias cada vez menores. Asimismo, el ancho de bus para cada etapa es diferente, siendo menor en la primera etapa e incrementándose gradualmente en las etapas posteriores. De este modo, en comparación con la realización no recursiva de una sola etapa, el filtrado se vuelve más eficiente en consumo de potencia y los requerimientos computacionales también se reducen.

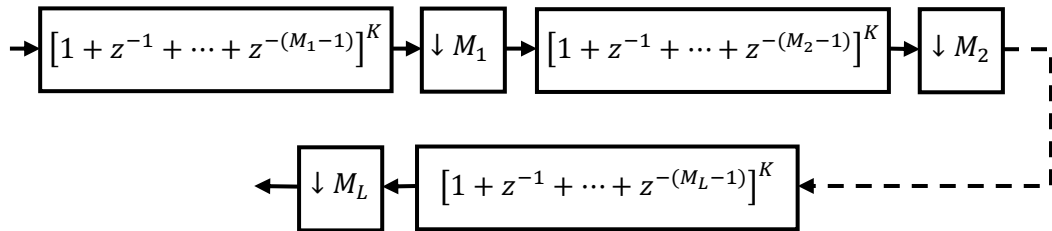


Figura 2.6 Estructura de un decimador multietapas basado en comb no recursivo.

2.5.2 Decimación polifásica

Una reducción adicional del consumo de potencia se puede lograr mediante el uso de la técnica de descomposición polifásica, que permite realizar la función de transferencia no recursiva en una realización en paralelo de M subfiltros, como se observa en las siguientes expresiones:

$$\begin{aligned} \left[\sum_{i=0}^{M-1} z^{-i} \right]^K &= h_0 + h_1 z^{-1} + \dots + h_{(K(M-1)-1)} z^{-(K(M-1)-1)} + h_{K(M-1)} z^{-(M-1)K} \\ &= H_0(z^M) + z^{-1} H_1(z^M) + \dots + z^{-(M-1)} H_{M-1}(z^M) = \sum_{m=0}^{M-1} z^{-m} \cdot H_m(z^M), \quad (2.13) \end{aligned}$$

donde:

$$H_m(z^M) = \sum_{j=0}^{\lfloor \frac{K(M-1)-m}{M} \rfloor} h_{M \cdot j + m} \cdot z^{-M \cdot j}; \quad m = 0, 1, \dots, M-1. \quad (2.14)$$

Después, al introducir los filtros polifásicos de (2.15) en el esquema decimador y haciendo uso de la tercera identidad multi-tasa, se obtiene la estructura computacionalmente eficiente de la Figura 2.7. Se observa que el filtrado se vuelve más eficiente, ya que la frecuencia de operación de cada subfiltro es M veces menor. Además, se propicia que solo se efectúen las operaciones necesarias, evitando que los filtros ejecuten cálculos innecesarios como en los esquemas decimadores no polifásicos.

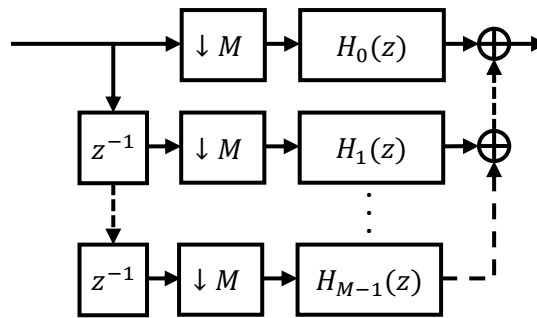


Figura 2.7 Estructura de un decimador polifásico.

Al aplicar el método de factorización polinomial seguido de la técnica de descomposición polifásica en cada etapa, se consigue un decimador con el menor consumo de potencia. No obstante, se incrementa la complejidad tanto en el diseño como en la implementación.

Capítulo 3

Métodos para Mejorar la Respuesta de Magnitud del Filtro Comb

Como se ha visto en el capítulo anterior, las características del filtro comb lo convierten en el más simple y utilizado en la primera etapa de decimación, no obstante su respuesta de magnitud no es la adecuada en determinadas aplicaciones donde se requiere mínima distorsión de la señal que se decima. Por desgracia, la respuesta de magnitud exhibe poca atenuación entre las bandas de rechazo y gran caída en banda de paso. A consecuencia de ello, se han propuesto diversos métodos para compensar la caída e incrementar la atenuación. Algunos de ellos modifican la función de transferencia del filtro comb al conectar en cascada filtros adicionales. Técnicas como los filtros correctores [12] y diversos compensadores [13]-[14] son ejemplo de ello. En cambio, otros métodos mejoran la respuesta de magnitud al modificar directamente la estructura del filtro comb. Entre estos métodos se pueden mencionar la técnica de sharpening [15], el método de rotación de ceros introducido por Presti [16] y generalizado por Laddomada [8].

En este capítulo se hace una revisión de los métodos de mejora de la respuesta de magnitud del filtro comb. Se comienza con aquellos que incrementan la atenuación en las bandas de rechazo y que están basados en el filtro coseno. Posteriormente se presentan tres métodos que reducen la caída en la banda de paso.

3.1 Métodos basados en el filtro coseno para el incremento de atenuación entre las bandas de rechazo

3.1.1 Filtro Coseno

Existen técnicas que incrementan la atenuación en las bandas de rechazo al conectar un filtro adicional en cascada con el filtro comb. Se requiere que la estructura añadida no introduzca demasiada complejidad, por lo que debería no contener multiplicadores y

realizarse con un número reducido de recursos. Bajo estas condiciones, uno de los filtros más simples que se puede utilizar para lograr tal fin es el llamado filtro coseno, que tiene función de transferencia dada por:

$$H_{cos}(z^N) = \frac{1}{2} [1 + z^{-N}], \quad (3.1)$$

donde N debe ser un número entero y es conocido como factor de expansión. La realización de esta función de transferencia requiere únicamente un sumador y N registros. Al evaluar la expresión 3.1 en el círculo unitario se encuentra la respuesta en frecuencia del filtro coseno:

$$H_{cos}(e^{j\omega}) = e^{-j\omega\frac{N}{2}} \cdot \cos\left(\frac{\omega N}{2}\right). \quad (3.2)$$

En la expresión anterior, fácilmente puede verse que la respuesta de magnitud está caracterizada por la función coseno, de donde se deriva el nombre del filtro. Además, tiene fase lineal. La Figura 3.1 muestra las gráficas de respuesta de magnitud de dos filtros coseno con factores de expansión 6 y 12. Se advierte que este filtro tiene $[N/2]$ ceros a lo largo del eje de frecuencias normalizado. También se nota que su primer cero está ubicado en la frecuencia $\frac{\pi}{N}$, mientras que los ceros restantes se hallan en múltiplos enteros impares de dicha frecuencia.

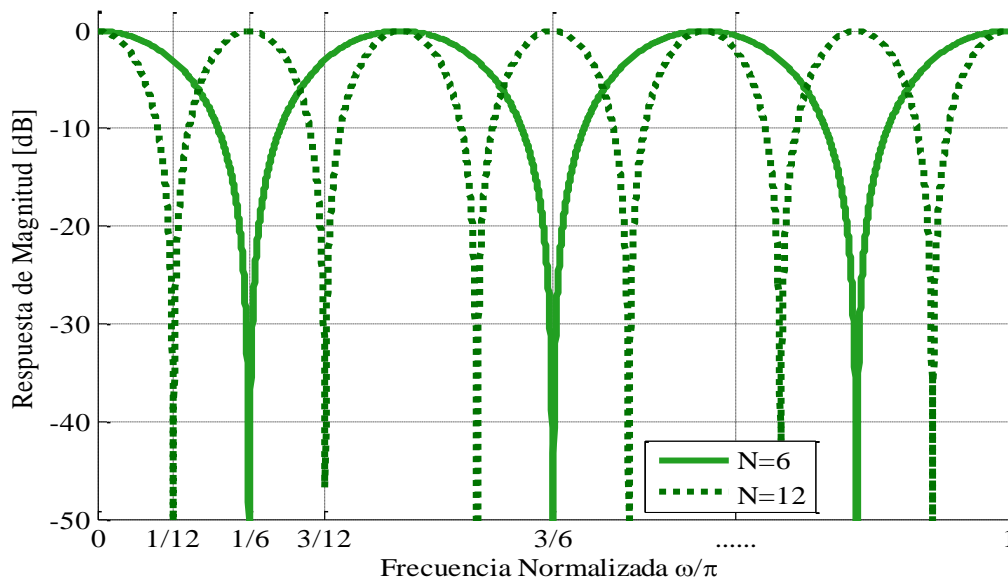


Figura 3.1 Respuesta de magnitud de filtros coseno.

A continuación se muestra una manera de incrementar la atenuación de las bandas de rechazo del filtro comb conectando en cascada filtros coseno. Se ha visto en la sección 2.3 que las folding bands del filtro comb están ubicadas al rededor de sus ceros, teniendo un espesor que es igual al doble del ancho de banda de la señal que se desea preservar. Por lo tanto, si se coloca un cero adicional dentro de una determinada banda de rechazo, entonces la atenuación total conseguida en tal banda se incrementará. La idea anterior se puede llevar a cabo conectando en cascada un filtro coseno y eligiendo apropiadamente el valor de su factor de expansión. Por ejemplo, para incrementar la atenuación en la primera banda de rechazo se puede igualar la frecuencia de ubicación del primer cero del filtro coseno, con aquella del primer cero del filtro comb, y después resolver para el valor de N . El resultado al que se llega es $N = \left\lceil \frac{M}{2} \right\rceil$, donde el redondeo hacia el entero inmediato superior asegura el mejoramiento del WCA cuando N es impar. Además, debe destacarse que al elegir el valor de N en esta forma, el filtro coseno posiciona sus ceros restantes en las frecuencias $\omega = \frac{\pi}{N} \cdot i$, con $i = 3, \dots, 2 \cdot \left\lceil \frac{N}{2} \right\rceil - 1$. Estos ceros caen en la i -ésima folding band o en su cercanía, y por consiguiente también se incrementa la atenuación de las folding bands impares. En general, si el factor de decimación es divisible entre 2^p , con p entero, una elección conveniente es establecer $N_j = \frac{M}{2^j}$, siendo $j \leq p$. Bajo estas circunstancias, es posible conectar en cascada p filtros coseno con el filtro comb, y lograr un incremento en la atenuación de las folding bands que son múltiplos enteros impares de 2^{j-1} .

Ejemplo 3.1. Se mejora la atenuación entre las bandas de rechazo de un filtro comb con parámetros $M = 24$ y $K = 2$. Para ello, se han conectado en cascada tres filtros coseno eligiendo $N_j = \frac{24}{2^j}$, con $j = 1, 2, 3$. Este método se ilustra en la Figura 3.2, donde se observa que eligiendo $j = 1$, el filtro coseno resultante mejora la atenuación entre las folding bands 1, 3, 5, 7, ... etc. Del mismo modo, el filtro coseno producido con $j = 2$ aumenta el rechazo antialiasing en las folding bands 2, 6 y 10. Y al hacer $j = 3$, el respectivo filtro coseno logra un incremento de atenuación en las bandas de rechazo 4 y 12. Debe advertirse que la caída en la banda de paso incrementa ligeramente.

En la literatura se han reportado diversas propuestas en las que el filtro coseno es la base para mejorar la atenuación entre las folding bands. Enseguida se da una introducción a algunas de ellas.

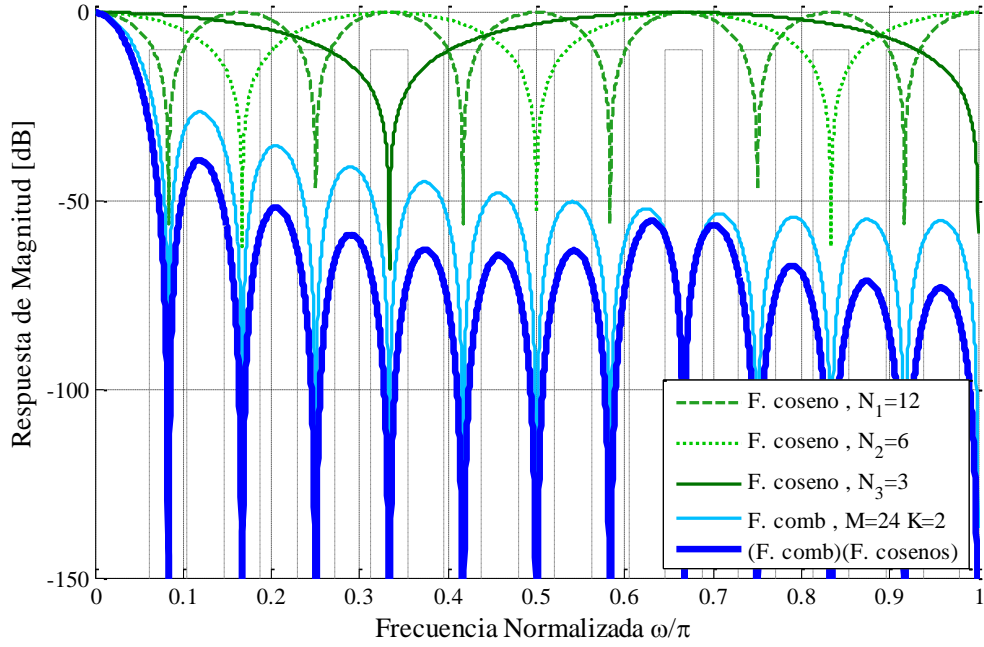


Figura 3.2 Ilustración del método para mejorar la respuesta de magnitud del filtro comb a través de la conexión en cascada de filtros coseno.

3.1.2 Pre-filtro Coseno [17]

El pre-filtro coseno fue introducido por Lian y Lim [17], se forma al conectar en cascada un filtro coseno expandido por $2N$ y dos filtros coseno expandidos por N . Tiene función de transferencia expresada por:

$$H_{PC}(z^N) = \frac{1}{8} [1 + z^{-2N}] [1 + z^{-N}]^2. \quad (3.3)$$

Y su correspondiente respuesta de magnitud es como sigue:

$$|H_{PC}(e^{j\omega})| = \frac{1}{2} |\cos(N\omega) + \cos^2(N\omega)|. \quad (3.4)$$

Como se observa en la Figura 3.3, esta respuesta de magnitud es periódica cada $\frac{2\pi}{N}$, y presenta múltiples ceros. En los puntos del eje de frecuencia que son múltiplos enteros impares de $\omega = \frac{\pi}{N}$, existen dos ceros superpuestos que ocasionan una gran atenuación.

Alrededor de estos puntos la atenuación decrece tenuemente debido a la existencia de un cero a la izquierda y otro a la derecha.

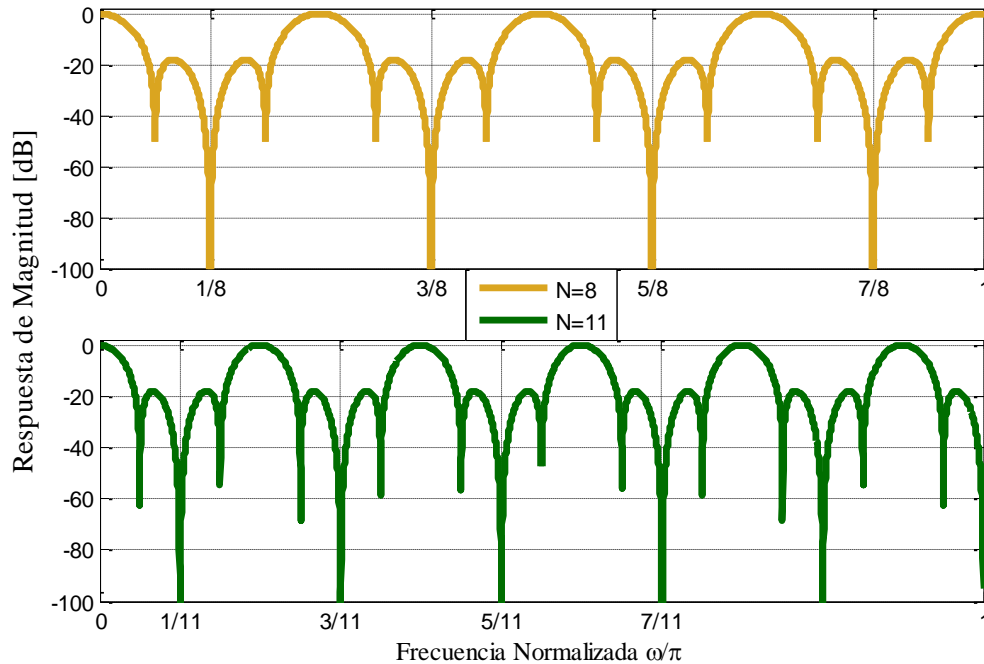


Figura 3.3 Respuesta de magnitud de pre-filtros coseno.

En varios métodos se ha sugerido utilizar al pre-filtro coseno para lograr un incremento en la atenuación del filtro comb. Jovanovic y Laddomada [18] proponen conectar en cascada L veces el mismo pre-filtro coseno y asignar a su factor de expansión un valor de $N = \frac{M}{2}$. De este modo, el segundo cero del pre-filtro coseno, cuya multiplicidad es $2L$, cae justo encima del primer cero del filtro comb. Adicionalmente, se consigue colocar L ceros a la derecha y otros L a la izquierda. Esto se repite en todas las folding bands impares, y por lo tanto se obtiene un incremento notable en su atenuación.

Ejemplo 3.2. Se requiere incrementar la atenuación de un filtro comb cuyo factor de decimación es $M = 20$ y su orden $K = 2$. Utilizando el método descrito anteriormente se elige $L = 1$ y el pre-filtro coseno tiene $N = 10$. En la Figura 3.4 se bosquejan las gráficas tanto del filtro comb como del pre-filtro coseno. Así mismo, se presenta la gráfica resultante al conectarlos en cascada. Adviértase que se consigue un considerable incremento de atenuación en las folding bands impares.

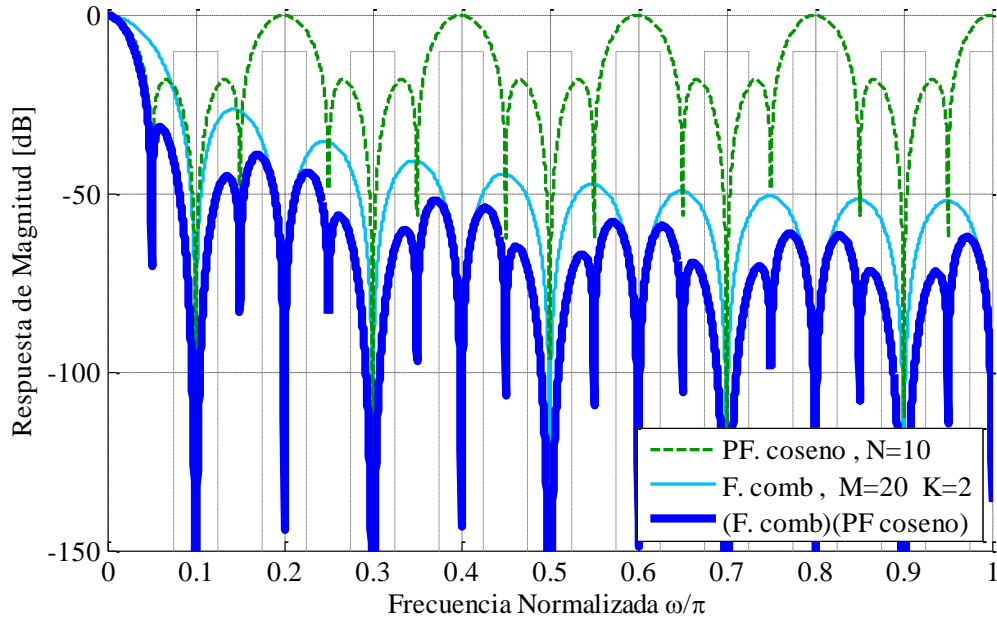


Figura 3.4 Ilustración del método para incrementar la atenuación en las folding bands utilizando un pre-filtro coseno.

Por otra parte, Jovanovick y Díaz [19] proponen conectar en cascada varios pre-filtros coseno con el comb, y elegir sus respectivos factores de expansión de acuerdo al siguiente criterio $N_i = \frac{M}{2^{i+1}}$. El filtro resultante tiene la siguiente función de transferencia:

$$H_{mCCOS}(z) = H(z) \prod_{i=1}^T H_{PC}(z^{N_i}), \quad (3.5)$$

donde T es el número de pre-filtros coseno conectados en cascada. Adicionalmente, se ha propuesto expresar la función de transferencia del filtro comb en su forma multi-etapas (ver sección 2.4.1), y escoger $N_T = M_1$, de modo que todos los pre-filtros coseno se puedan mover a menor frecuencia. Se aclara que con este método es posible incrementar la atenuación en la mayoría de las folding bands, si se elige $T > 2$.

Ejemplo 3.3. Considérese ahora un filtro comb que tiene orden $K = 2$ y $M = 24$. En este ejemplo se conectan en cascada dos prefiltros coseno con $N_1 = 6$ y $N_2 = 3$. La respuesta de magnitud conseguida se ilustra en la Figura 3.5. Nótese el significativo incremento de atenuación en todas las folding bands. También, debe observarse un inconveniente del método: la desviación en la banda de paso sufre un incremento.

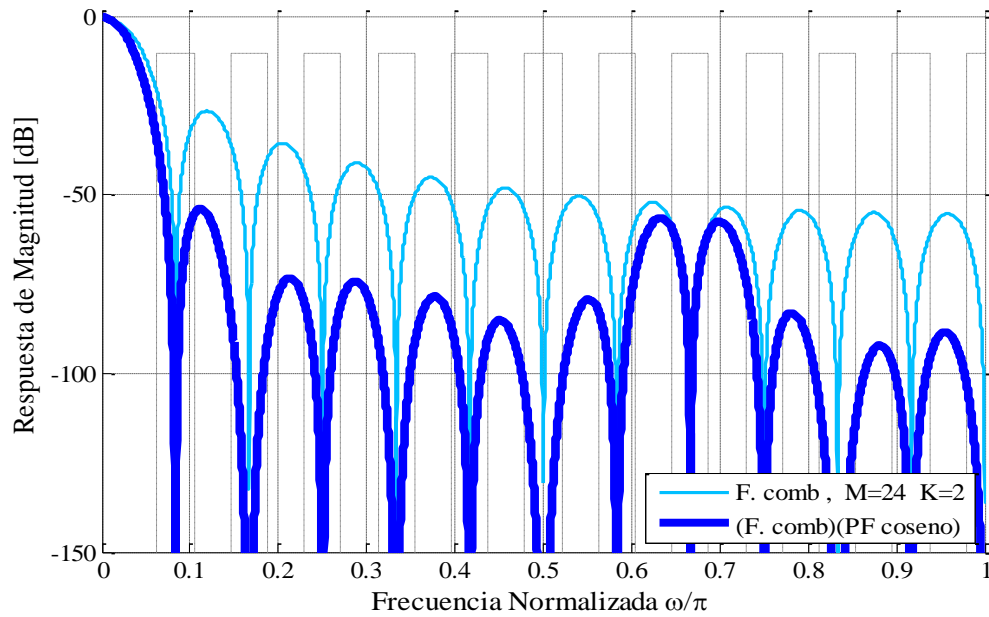


Figura 3.5 Ilustración del método para incrementar la atenuación en las folding bands utilizando dos pre-filtros coseno.

3.1.3 Filtro coseno rotado [20]

Teran y Dolecek [20], plantearon aplicar la técnica de rotación de ceros a la estructura del filtro coseno. Al efectuar una rotación por un ángulo α en sentido de las manecillas del reloj (SMR) a la expresión (3.1) se llega a:

$$H_u(z^N) = \frac{1}{2} [1 + z^{-N} e^{j\alpha N}]. \quad (3.6)$$

Si la rotación se hace en el sentido opuesto, la función de transferencia resultante es:

$$H_d(z^N) = \frac{1}{2} [1 + z^{-N} e^{-j\alpha N}]. \quad (3.7)$$

Estos filtros tienen coeficientes complejos, pero al conectarlos en cascada y combinarlos se genera un filtro con coeficientes reales, llamado coseno rotado, cuya función de transferencia es la siguiente:

$$H_r(z^N) = H_u(z)H_d(z) = \frac{1}{4} [1 + 2\cos(\alpha N)z^{-N} + z^{-2N}]. \quad (3.8)$$

Los nulos del filtro se ubican ahora en las frecuencias: $\frac{\pi i}{N} \pm \alpha$, $i = 1, 3, 5, \dots, 2 \cdot \left\lfloor \frac{N}{2} \right\rfloor - 1$.

De igual forma que en los métodos anteriores, se pueden conectar en cascada varios filtros coseno rotado con el filtro comb y con ello mejorar las bandas de rechazo al seleccionar apropiadamente los valores del ángulo de rotación y de los factores de expansión. La aplicación de esta técnica es viable en filtros comb cuyos factores de decimación son potencia de dos, ya que, por un lado permite ubicar los ceros del filtro coseno rotado justo entre las folding bands y maximizar su atenuación. Por otra parte, al realizar la decimación en un proceso de múltiples etapas de decimación por dos, los filtros coseno rotado se pueden introducir en distintas etapas, lo que conduce a una disminución tanto en su frecuencia de operación como en la energía consumida. La función de transferencia correspondiente a este método se expresa como sigue:

$$H_{Cr}(z) = \left[\prod_{l=0}^{p-1} \frac{1}{2} (1 + z^{-2^l}) \right]^K \prod_{k=0}^{R-1} H_r \left(z^{\frac{M}{2^k}} \right). \quad (3.9)$$

Se ha indicado que el número de filtros coseno rotado introducidos R , sea menor o igual tres para evitar aumentar demasiado la complejidad del filtro total. El criterio para la elección del ángulo de rotación dicta que debe ser igual al valor de la frecuencia de paso:

$$\alpha = \omega_p = \frac{\pi}{Mv}. \quad (3.10)$$

Con ello se logra un máximo ensanchamiento de las bandas de rechazo, y por lo tanto se previene el alising de mejor manera. Finalmente, con el fin de evitar multiplicadores en la realización, se ha propuesto el redondeo del coeficiente $2\cos(\alpha N)$ de la siguiente forma:

$$\beta_r = r \cdot \text{round} \left(\frac{2\cos(\alpha N)}{r} \right), \quad (3.11)$$

donde r se denomina constante de redondeo y debe ser un número potencia de dos, en este caso $r = 2^{-6}$. A partir de la expresión (3.10), el coeficiente $2\cos(\alpha N)$ ya se puede implementar como corrimientos y sumas.

Ejemplo 3.4. Se incrementa la atenuación entre las folding bands de un filtro comb que tiene $M=16$ y $K=3$. Utilizando dos filtros coseno rotado, los parámetros obtenidos son: $N_1 = 8$, $N_2 = 4$ y $\alpha = 0.0982$. En la Figura 3.6 se comparan las respuestas de

magnitud del filtro comb tradicional con la del filtro $H_{Cr}(z)$. Nótese que la atenuación en la mayoría de las folding bands se ha mejorado notablemente.

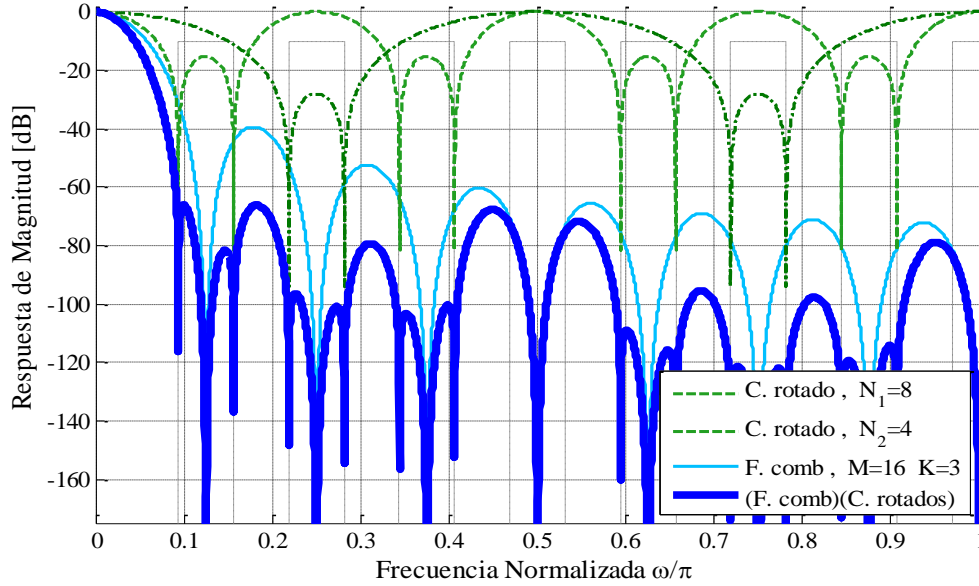


Figura 3.6 Ilustración del método para incrementar la atenuación en las folding bands utilizando filtros coseno rotado.

3.1.4 Inserción de dos filtro coseno [21]

El objetivo principal de este método es incrementar la atenuación en la primera folding band, ya que es en esta donde ocurre el menor rechazo antialiasing. Esto se logra mediante la conexión en cascada de dos filtros coseno con el filtro comb, obteniéndose:

$$H_{CS}(z) = H(z)H_S(z) = \left[\frac{1}{M} \frac{1 - z^{-M}}{1 - z^{-1}} \right]^K \cdot \left[\frac{1}{4} [1 + z^{-N_1}][1 + z^{-N_2}] \right]. \quad (3.12)$$

Los factores de expansión, N_1 y N_2 , se eligen con el fin de colocar dos ceros adicionales en la proximidad del primer cero del filtro comb sin que se encimen en este. De acuerdo con lo anterior, se busca posicionar un cero a la izquierda y otro a la derecha de la frecuencia $\omega = \frac{2\pi}{M}$, lo que resulta en dos casos que dependen del factor de decimación:

$$M \text{ par: } N_1 = \frac{M}{2} - 1, \quad N_2 = \frac{M}{2} + 1; \quad (3.13)$$

$$M \text{ impar: } N_1 = \left\lfloor \frac{M}{2} \right\rfloor, \quad N_2 = N_1 + 1. \quad (3.14)$$

Asimismo, a través de estos criterios de selección de los factores de expansión, se incrementa la atenuación en aproximadamente el 50% de las folding bands. De este modo, si M es un número par, en el intervalo definido por $\omega = \left[0, \frac{\pi}{2}\right]$, se mejoran las folding bands impares, y en el intervalo acotado por $\omega = \left[\frac{\pi}{2}, 1\right]$, las bandas de rechazo que incrementan su atenuación son las pares. En tanto que, cuando M es un número impar, únicamente las bandas de rechazo impares incrementan su atenuación entre 10 y 25 dB.

Ejemplo 3.5. Se pretende aumentar la atenuación entre las folding bands de un filtro comb definido por los siguientes parámetros: $M = 16$ y $K = 3$. Al emplear el método descrito con antelación, se obtiene: $N_1 = 9$, $N_2 = 7$. Se resalta el incremento mayor a 15 dB en la atenuación de la primera, tercera, sexta y octava bandas de rechazo, como se puede ver en la Figura 3.7, donde se presentan las respuestas de magnitud del filtro comb y del filtro $H_{CS}(z)$.

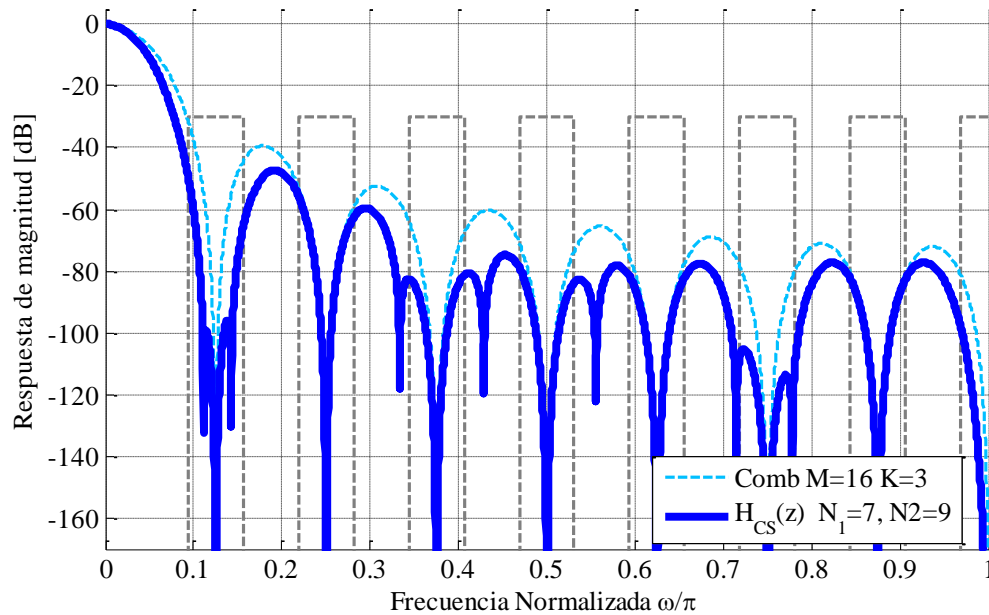


Figura 3.7 Respuestas de magnitud del filtro comb y del filtro $H_{CS}(z)$ del ejemplo 3.5.

3.2 Métodos para la disminución de la caída en banda de paso

Como ya se vio, la magnitud del filtro comb presenta una gran caída en su banda de paso que crece monótonicamente al incrementar el orden. Dicha caída distorsiona la señal por lo que debe ser reducida. Generalmente, esto se hace al conectar un filtro inmediatamente después del submuestreador, que recibe el nombre de filtro compensador. Las características más deseadas en el compensador se enuncian como sigue:

- No debe utilizar multiplicadores.
- El número de recursos utilizados tiene que ser reducido.
- Debe trabajar a la frecuencia más baja
- Su diseño no tiene que depender de M .

La idea fundamental en el desarrollo de un compensador consiste en aproximar la respuesta de magnitud inversa de un filtro comb dentro de la banda de paso, dada por:

$$|H_{inv}(e^{j\omega})| = \frac{1}{|H(e^{j\omega})|} = \left| M \frac{\sin\left(\frac{\omega}{2}\right)}{\sin\left(\frac{\omega M}{2}\right)} \right|^K, \quad 0 \leq \omega \leq \omega_p. \quad (3.15)$$

Se han sugerido diversos métodos para lograr tal aproximación usando filtros FIR de orden bajo que no requieren multiplicadores. A continuación se hace una revisión de algunos de ellos.

3.2.1 Filtro compensador de segundo orden basado en un enfoque trigonométrico [22]

Jovanovic y Fernandez [22] propusieron un filtro compensador muy simple que tiene respuesta de magnitud dada por:

$$G(e^{j\omega M}) = 1 + B \cdot \sin^2\left(\frac{\omega M}{2}\right) \approx \left| M \frac{\sin\left(\frac{\omega}{2}\right)}{\sin\left(\frac{\omega M}{2}\right)} \right|^K, \quad 0 \leq \omega \leq \omega_p, \quad (3.16)$$

donde B es una constante positiva y a su vez es el único parámetro de diseño. Más adelante se aborda la manera de elegirlo.

La anterior respuesta de magnitud corresponde a un filtro FIR, cuya función de transferencia es:

$$G(z) = 2^{-2}[B(-1 + 2z^{-M} - z^{-2M}) + 2^2z^{-M}]. \quad (3.17)$$

La conexión en cascada de este compensador con el filtro comb, produce que la caída en la banda de paso disminuya al seleccionar adecuadamente el valor de B . Éste depende del orden del filtro comb y de la máxima desviación en la banda de paso que puede tolerarse en determinada aplicación. Su elección se hace mediante un algoritmo de optimización como se definió en [23]. Tomando en cuenta un valor típico de valor absoluto de la desviación en banda de paso, $\delta \leq 0.4$ dB, en la Tabla 3.1 se muestran los valores de B que corresponden ante diferentes valores del orden K . También se incluyen los valores redondeados hacia números que son sumas de potencia de dos, con ello se pueden sustituir los multiplicadores por corrimientos y sumadores.

K	B	Redondeo
1	0.321	$0.25 = 2^{-2}$
2	0.5677	$0.5 = 2^{-1}$
3	0.8035	$0.75 = 2^{-1} + 2^{-2}$
4	1.0372	$1 = 2^0$
5	1.2693	$1.25 = 2^0 + 2^{-2}$

Tabla 3.1 Valores de B que satisfacen $\delta \leq 0.4$ dB.

Finalmente, haciendo uso de la tercera identidad multitasas, el compensador se implementa a la menor frecuencia después del submuestreador, con lo cual, este filtro satisface las características deseables que se describieron al principio de esta sección.

Ejemplo 3.5. Se reduce la caída en banda de paso de un filtro comb con $M = 20$ y $K = 5$. De acuerdo a la Tabla 3.1, se elige $B = 1.25$. Las gráficas de la respuesta de magnitud del filtro no compensado y compensado se muestran en la Figura 3.8. Nótese que la máxima desviación en la banda de paso es menor a 0.4 dB en el filtro compensado.

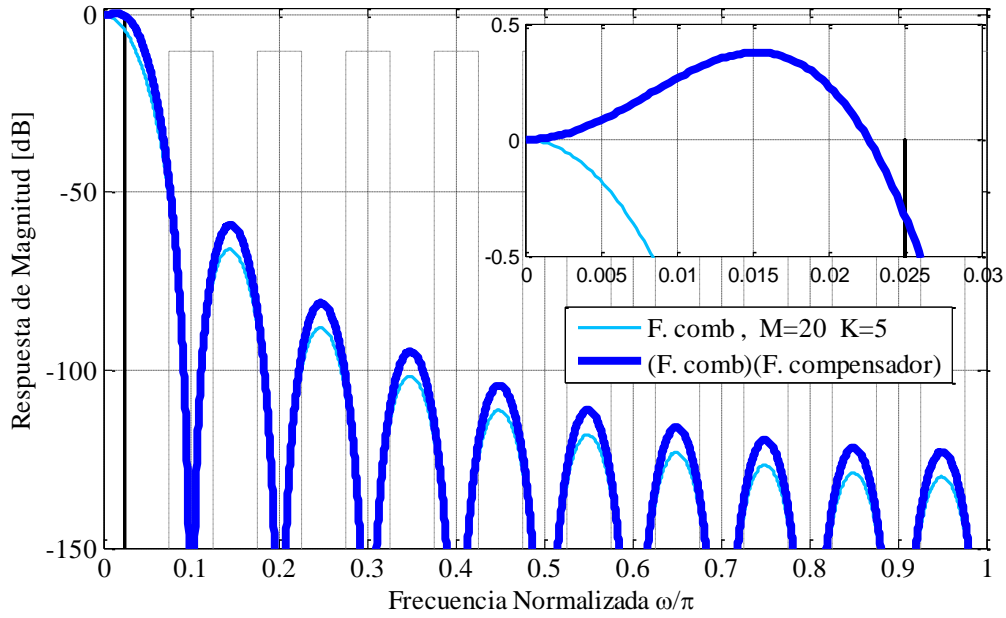


Figura 3.8 Compensación del filtro comb utilizando método [22].

3.2.2 Filtro compensador basado en el criterio de error minimax [23]

Glavinic, Molnar y Vucic [23] presentan un método para el diseño de compensadores comb. Dicho método está basado en el criterio de error minimax. La obtención de los valores óptimos de los coeficientes del compensador, los cuales son expresados como sumas de potencias de dos (SPT), se consigue mediante análisis de intervalos.

La propuesta consiste en aproximar la respuesta de magnitud de un compensador comb ideal mediante un filtro FIR tipo 1. En este caso para un filtro FIR tipo 1, cuyos coeficientes son SPT y están dados en un determinado número de términos P , la amplitud es:

$$H_{comp}(\omega, \mathbf{x}) = 1 + x(0) + 2 \sum_{l=1}^{(L-1)/2} x(l) \cos(\omega l). \quad (3.18)$$

El objetivo es encontrar el conjunto de coeficientes óptimos SPT del compensador, que satisfagan el criterio de error minimax sobre una banda deseada. De esta manera, se define la función objetivo como:

$$\epsilon(\mathbf{x}) = \max_{|\omega| \leq |\omega_p|} \left| 1 - H\left(\frac{\omega}{M}\right) H_{comp}(\omega, \mathbf{x}) \right|, \quad (3.19)$$

donde ω_p es el borde superior de la banda de paso. El vector \mathbf{x} es de longitud L y contendrá los coeficientes óptimos SPT del compensador. La solución se obtiene mediante un análisis de intervalos de la función objetivo y se resume en tablas en la referencia [23].

Ejemplo 3.6. Con el fin de ilustrar las características de los compensadores obtenidos con el método presentado, se describe la compensación de un filtro CIC con $M = 18$ y $K = 5$. La compensación se hace sobre la banda con borde superior $\omega_p = 0.5\pi$. Se diseñan tres compensadores con diferente complejidad. En los primeros dos casos se establece el número de coeficientes $L = 3$, y considerando número de términos $P = 2$ y $P = 3$. También se diseña un compensador con $L = 5$ y $P = 2$. De este modo, los coeficientes de los compensadores, obtenidos de la Tabla I y II de [23], son los siguientes:

$$h_{C1} = [-2^{-2}-2^{-4}, 1 + 2^{-1} + 2^{-3}, -2^{-2}-2^{-4}],$$

$$h_{C2} = [-2^{-2}-2^{-4} - 2^{-6}, 1 + 2^{-1} + 2^{-3} + 2^{-7}, -2^{-2}-2^{-4} - 2^{-6}] \text{ y}$$

$$h_{C3} = [2^{-4}-2^{-7}, -2^{-1}+2^{-4}, 1 + 2^{-1} + 2^{-2}, -2^{-1}+2^{-4}, 2^{-4}-2^{-7}].$$

El número de sumadores para cada compensador es 5, 7 y 8 respectivamente. La Figura 3.9 muestra las repuestas de magnitud en el intervalo de la banda de paso del filtro comb y los filtro comb compensados. Se observa que la caída se reduce considerablemente, sin embargo para los primeros dos compensadores la desviación en banda de paso es mayor a 0.4 dB, mientras que para el compensador con 5 coeficientes la desviación en banda de paso es menor a 0.3 dB. Se observa que el incremento de número de coeficientes produce mejor compensación que el incremento de número de términos, aunque ello implica un incremento en el número de sumadores.

3.2.2 Filtro compensador basado en la transformación de amplitud del filtro coseno cuadrado [24]

Troncoso y Jovanovic [24] proponen el diseño de filtros compensadores mediante la aplicación de una función de cambio de amplitud (ACF, por sus siglas en inglés) de primer grado a un filtro coseno cuadrado.

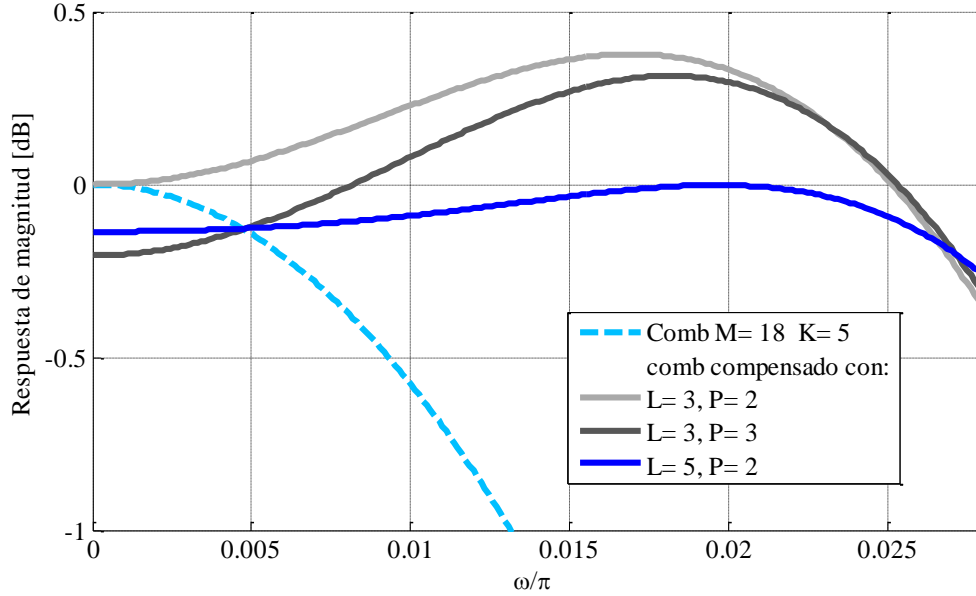


Figura 3.9 Respuestas de magnitud de filtro comb no compensado y compensado con diferentes longitudes del compensador (L) y números de términos (P), utilizando método [23].

La técnica de ACF está basada en la utilización repetida de un filtro, llamado filtro prototipo y denotado por $F(z)$, para modificar su respuesta de magnitud, $F(e^{j\omega})$. Su empleo está limitado a filtros no recursivos con respuesta al impulso simétrica. Al aplicar una ACF a un filtro se obtiene una transformación en su función de transferencia dada por:

$$F_s(z) = \sum_{i=0}^Q z^{-\frac{(Q-i)L}{2}} p_i F^i(z), \quad (3.20)$$

donde Q es el grado de la ACF y p_i son sus coeficientes. La correspondiente respuesta frecuencia es:

$$F_s(\omega, \mathbf{p}) = \sum_{i=0}^Q p_i F^i(\omega). \quad (3.21)$$

El filtro coseno cuadrado tiene función de transferencia dada por:

$$F_c(z) = \frac{1}{4} [1 + 2z^{-1} + z^{-2}]. \quad (3.22)$$

Puesto que el filtro coseno cuadrado es muy simple, y su respuesta de magnitud exhibe un comportamiento monotónico similar al que presenta el filtro comb en el intervalo $\left[0, \frac{\pi}{\rho}\right]$, es una buena elección para poder realizar un compensador. Esto se logra al utilizarlo como filtro prototipo en una ACF de primer grado. De manera que al sustituir (3.22) en (3.20) y eligiendo $Q=1$, se obtiene:

$$F_s(z) = \frac{1}{4} [p_1 + 2(2y_0 - p_1)z^{-1} + p_1z^{-2}], \quad (3.23)$$

donde se ha considerado $p_0 = y_0 - p_1$, siendo y_0 el valor que toma la magnitud del compensador cuando $\omega = 0$. Al elegir $y_0 = 1$, el compensador se realiza en su forma más simple. De este modo, el objetivo es encontrar el valor del coeficiente p_1 de manera que se minimice una función de error, esto es:

$$p_1 = \arg \min_{0 \leq \omega \leq \frac{\pi}{\rho}} \left\{ \left\| 1 - F_s(\omega, p_1) H^K \left(\frac{\omega}{M} \right) \right\|_{L_p} \right\}. \quad (3.24)$$

El resultado al que se llega es:

$$|p_1| = 2^{-r} \left| \text{round} \left(\frac{q_2 K^2 + q_1 K + q_0}{2^{-r}} \right) \right|, \quad (3.24)$$

donde se ha propuesto la representación de p_1 utilizando r bits de parte decimal, $2 \leq r \leq 6$, para sustituir el multiplicador por corrimientos y sumadores. Los coeficientes q_2, q_1 y q_0 dependen del factor de decimación residual, ρ , y se eligen como sigue:

$$q_2 = 0.0736\rho^{-2.578}; \quad (3.25)$$

$$q_1 = 0.1717; \quad (3.26)$$

$$q_0 = 0.5438\rho^{-3.3} - 0.001845. \quad (3.27)$$

Ejemplo 3.7. Mediante el filtro compensador revisado, se debe reducir la caída en la banda de paso de un filtro comb de orden $K = 5$ y factor de decimación $M = 20$. Se toma el mínimo factor de decimación residual, $\rho = 2$, y se considera la representación de p_1 con 3 bits de parte decimal. El valor del parámetro de diseño que se obtiene es

$|p_1| = 1.25$. En la Figura 3.10 se aprecia que con este método la máxima desviación en banda de paso es 0.376 dB.

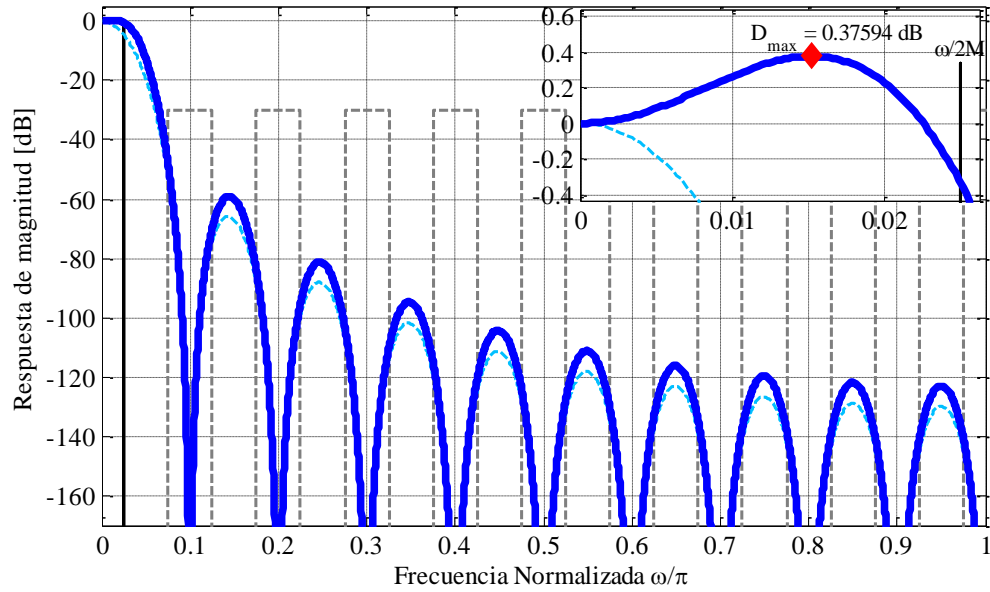


Figura 3.10 Filtro comb compensado utilizando el método [24].

Con los filtros compensadores descritos anteriormente, concluye la revisión de métodos enfocados a mejorar la respuesta de magnitud del filtro comb.

Capítulo 4

Propuesta para Mejorar el Decimador Comb Basada en Filtros Coseno y Compensadores

En este capítulo se propone un método para mejorar el decimador comb, siendo aplicable a decimadores cuyo factor es un número compuesto. La propuesta consiste de una estructura de dos etapas, donde se introducen dos filtros coseno en la segunda etapa para lograr un incremento en la atenuación de las bandas de rechazo. Asimismo se presentan dos arquitecturas eficientes tanto en consumo de potencia como en área de chip. Además, mediante un filtro compensador que trabaja a la más baja frecuencia, se consigue una desviación en la banda de paso entre 0.3 dB y 0.6 dB. Finalmente, a través de diversas comparaciones, se muestran las ventajas del decimador propuesto.

4.1 Incremento de la atenuación entre las bandas de rechazo

En primer lugar se considera el mejoramiento del rechazo antialiasing. El objetivo es incrementar la atenuación que exhibe el filtro comb entre las bandas de rechazo y a la vez, hacerlo sin incrementar considerablemente su complejidad. Se busca conseguir un filtro decimador más simple en relación con los filtros comb-coseno revisados en la sección 3.1. Para lograr tal fin, la propuesta de esta tesis se ha desarrollado en un decimador de dos etapas basado en filtros comb y coseno.

4.1.1 Estructura comb de dos etapas

Un primer paso para conseguir un filtro comb de menor complejidad se enfoca en la obtención de una estructura de dos etapas, siendo particularmente válido para su realización no recursiva. Se ha visto en la sección 2.4.1 que si el factor de decimación es número compuesto, entonces el filtro comb se puede dividir en múltiples etapas. La ventaja de la estructura multietapas radica en que requiere menores recursos de hardware y permite efectuar el filtrado a diferentes frecuencias, que se vuelven cada vez más bajas. De esta manera, al combinar el filtro comb multietapas con los sumbuestreadores,

la primera etapa tiene la mayor velocidad de procesamiento, pero su ancho de bus es el menor, y, por el contrario, en la última etapa se requiere un gran ancho de bus, no obstante opera a la frecuencia más baja.

Con base en lo anterior, se propone utilizar una estructura comb de dos etapas. Esto se logra al descomponer en dos números enteros el factor de decimación. Bajo la condición de que éste no es un número primo, como se considera en lo sucesivo, entonces se puede representar como $M = M_1 \cdot M_2$. Los sub índices indican la etapa correspondiente. A partir de la descomposición de M , la función de transferencia del filtro comb se puede reescribir de la siguiente forma:

$$\left[\frac{1}{M} \frac{1 - z^{-M}}{1 - z^{-1}} \right]^K = \left[\frac{1}{M_1} \frac{1 - z^{-M_1}}{1 - z^{-1}} \right]^K \cdot \left[\frac{1}{M_2} \frac{1 - z^{-M_1 M_2}}{1 - z^{-M_1}} \right]^K = H_1(z) \cdot H_2(z^{M_1}). \quad (4.1)$$

En esta tesis se tiene en cuenta el caso general, en el que los órdenes de los filtros de cada etapa son diferentes, por ello el filtro comb que se utiliza tiene su función de sistema dada por:

$$H(z) = \left[\frac{1}{M_1} \frac{1 - z^{-M_1}}{1 - z^{-1}} \right]^{K_1} \cdot \left[\frac{1}{M_2} \frac{1 - z^{-M_1 M_2}}{1 - z^{-M_1}} \right]^{K_2}, \quad (4.2)$$

donde K_1 y K_2 representan los órdenes de los filtros comb de la primera y segunda etapa, respectivamente. La virtud principal que surge al definir $K_1 \neq K_2$ es la posibilidad de incrementar la atenuación de un filtro comb de orden K , al hacer la asignación $K_1 = K$, y confiriéndole a K_2 un valor mayor. Con ello, la complejidad del filtro incrementa ligeramente, pero sucede en la segunda etapa cuya frecuencia de operación es M_1 veces menor. Esta idea se clarifica con el siguiente ejemplo.

Ejemplo 4.1. Se tiene un filtro comb cuyo factor de decimación es $M = 15$ y su orden es $K = 4$. Como ya se mencionó, una manera de aumentar el rechazo antialiasing sin incrementar significativamente la complejidad del filtro es mediante una estructura de dos etapas y estableciendo $K_1 = K$ y $K_2 > K_1$. Para ejemplificarlo, se descompone el factor de decimación de la siguiente manera $M_1 = 3$ y $M_2 = 5$, y se asignan los siguientes valores a los órdenes $K_1 = K = 4$ y $K_2 = 6$. En la Figura 4.1 se presentan las

respuestas de magnitud de los dos filtros anteriores. Nótese que la magnitud del filtro comb de dos etapas exhibe mayor atenuación en la mayoría de las bandas de rechazo, excepto en la banda M_2 -ésima, donde la atenuación es muy similar a la que presenta el filtro comb de una etapa.

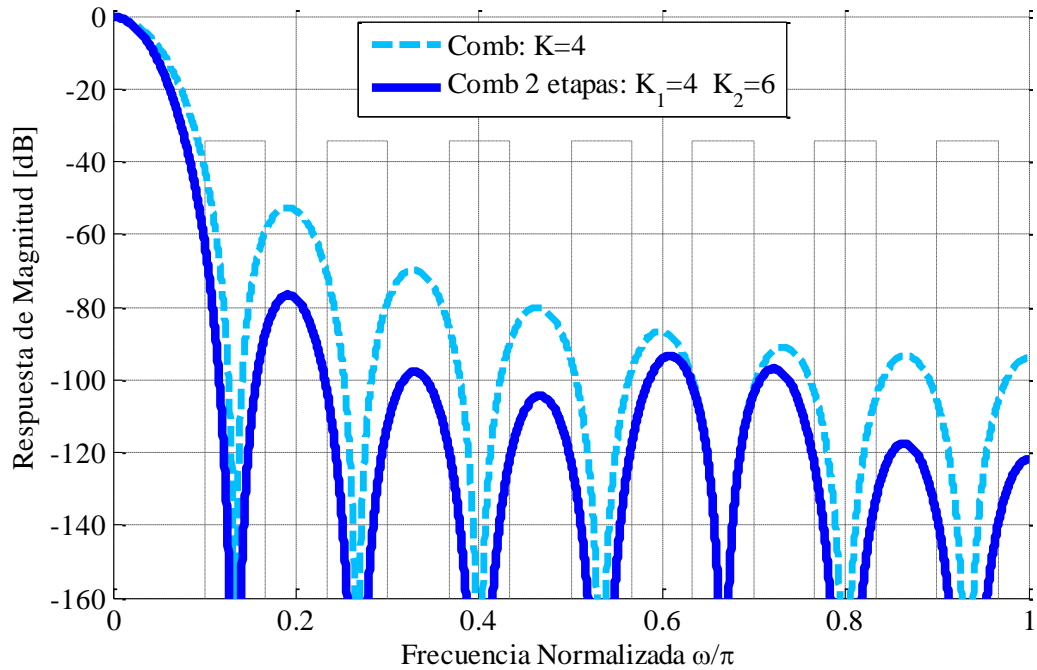


Figura 4.1 Respuestas de magnitud de los filtros del Ejemplo 4.1.

Es importante aclarar que la manera de descomponer el factor de decimación guarda una relación estrecha con el área y potencia utilizadas por el filtro. Más adelante, en la sección 4.1.4 se explica con mayor detalle la mejor manera de hacerlo, una vez que ya se haya definido por completo el filtro propuesto.

A parte de las características favorables ya mencionadas, la estructura decimadora de dos etapas resulta conveniente para mejorar la atenuación de las bandas de rechazo a un costo relativamente bajo, ya que permite insertar en la segunda etapa algunos filtros adicionales de baja complejidad que incrementen el rechazo antialiasing. En seguida se presenta un método basado en esta idea.

4.1.2 Inserción de dos filtros coseno en la segunda etapa de decimación

La conexión en cascada de filtros coseno es adecuada para lograr mayor atenuación entre las folding bands, pues son estructuras simples, sin multiplicadores y que requieren muy pocos recursos. Esta idea ya se revisó en la sección 3.1, donde se refieren diferentes métodos que la abordan. En este sentido, se propone insertar en la segunda etapa dos filtros coseno que incrementen el rechazo antialiasing. Su función de transferencia, en relación con la frecuencia más alta, se expresa como sigue:

$$H_S(z^{M_1}) = \left[\frac{1 + z^{-M_1 N_1}}{2} \right] \left[\frac{1 + z^{-M_1 N_2}}{2} \right]. \quad (4.3)$$

Al insertar los filtros coseno en la segunda etapa se logra una disminución en la complejidad del filtro decimador basado en filtros comb y coseno. Por un lado, tanto el número de sumas por muestra de salida (APOS) como el número de registros necesarios, son menores con respecto a los que se requerirían al tener un filtro comb de una sola etapa. Además, la frecuencia de operación de los filtros coseno insertados es M_1 veces menor, lo que implica un bajo consumo de potencia.

Primordialmente, se introducen los filtros coseno para incrementar la atenuación en la mayoría de las bandas de rechazo, sobretodo en la primera, ya que es bien sabido que en ésta ocurre el peor caso de atenuación (PCA). Se propone situar un cero a la izquierda y otro a la derecha del primer cero del filtro comb de dos etapas, a semejanza del método [21]. Para lograrlo se dispone de los parámetros N_1 y N_2 . De este modo, conociendo que el primer cero del filtro comb se ubica en la frecuencia:

$$\omega_{z1} = \frac{2\pi}{M_1 M_2}. \quad (4.4)$$

Primero se averigua el valor que debe tomar N_1 , que permita posicionar un cero a la derecha de la frecuencia (4.4). Por otro lado, se sabe que el primer cero del filtro coseno está en la frecuencia:

$$\omega_{z1,c} = \frac{\pi}{M_1 N_1}. \quad (4.5)$$

Al igualar las expresiones (4.4) y (4.5), y despejando N_1 se obtiene su valor mediante el cual se hacen coincidir los ceros del filtro comb y del filtro coseno. Dado que el objetivo es colocar el cero a la derecha, el valor de N_1 se precisa con la siguiente expresión:

$$N_1 = \left\lfloor \frac{M_2 - 1}{2} \right\rfloor. \quad (4.6)$$

Mediante un procedimiento similar se encuentra el valor de N_2 que posiciona un cero a la izquierda del primer cero del filtro comb, llegando a:

$$N_2 = \left\lceil \frac{M_2 + 1}{2} \right\rceil. \quad (4.7)$$

De las dos expresiones anteriores es fácil notar que los valores de N_1 y N_2 dependen únicamente del factor de decimación de la segunda etapa.

4.1.3 Filtro propuesto para el filtrado entre las bandas de rechazo

La función de sistema del filtro propuesto $H_p(z)$ que presenta mejor rechazo antialiasing que el filtro comb clásico y que se pretende sea de menor complejidad que los filtros expuestos en el Capítulo 3, se obtiene al combinar las expresiones (4.2) y (4.3), y está dada por:

$$\begin{aligned} H_p(z) &= H(z)H_S(z^{M_1}) \\ &= \left[\frac{1 - z^{-M_1}}{M_1(1 - z^{-1})} \right]^{K_1} \left[\frac{1 - z^{-M_1 M_2}}{M_2(1 - z^{-M_1})} \right]^{K_2} \left[\frac{1 + z^{-M_1 N_1}}{2} \right] \left[\frac{1 + z^{-M_1 N_2}}{2} \right]. \end{aligned} \quad (4.8)$$

Las variables M_1 , M_2 , K_1 y K_2 son los parámetros de diseño, cuya elección se discute en la sección 4.1.4. La respuesta de magnitud de este filtro es:

$$|H_p(e^{j\omega})| = \left| \frac{1}{M_1^{K_1} \cdot M_2^{K_2}} \cdot \left[\frac{\sin\left(\frac{\omega M_1}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right]^{K_1} \left[\frac{\sin\left(\frac{\omega M_2 \cdot M_1}{2}\right)}{\sin\left(\frac{\omega M_1}{2}\right)} \right]^{K_2} \cos\left(\frac{\omega N_1 M_1}{2}\right) \cos\left(\frac{\omega N_2 M_1}{2}\right) \right|. \quad (4.9)$$

En el siguiente ejemplo se muestran las características que sobresalen de la respuesta de magnitud del filtro propuesto en comparación con el filtro comb.

Ejemplo 4.2. Se diseña un filtro antialiasing mediante la función de transferencia propuesta, dada por la expresión (4.8). El factor de decimación es $M = 18$. De este modo, con fines ilustrativos, se asignan los siguientes valores $M_1 = 3$, $M_2 = 6$, $K_1 = 4 = K_2$. De acuerdo con (4.6) y (4.7) se tiene $N_1 = 2$ y $N_2 = 4$. La respuesta de magnitud del filtro obtenido se muestra en la Figura 4.2, donde también se muestra la magnitud de un filtro comb con $M = 18$ y $K = 4$. Adviértase que el rechazo antialiasing entre la mayoría de las bandas de rechazo es mayor para el filtro propuesto. Simultáneamente, el filtro propuesto presenta mayor atenuación en la primera banda de rechazo, teniendo un PCA de aproximadamente 25 dB mayor que el del filtro comb.

Por su parte, la fase del filtro propuesto es:

$$\arg\{H_{BR}(e^{j\omega})\} = \frac{\omega}{2} [K_1(M_1 - 1) + M_1(K_2(M_2 - 1) + N_1 + N_2)], \quad (4.10)$$

continúa siendo lineal, aunque el retraso de grupo es ligeramente mayor con respecto al del filtro comb clásico.

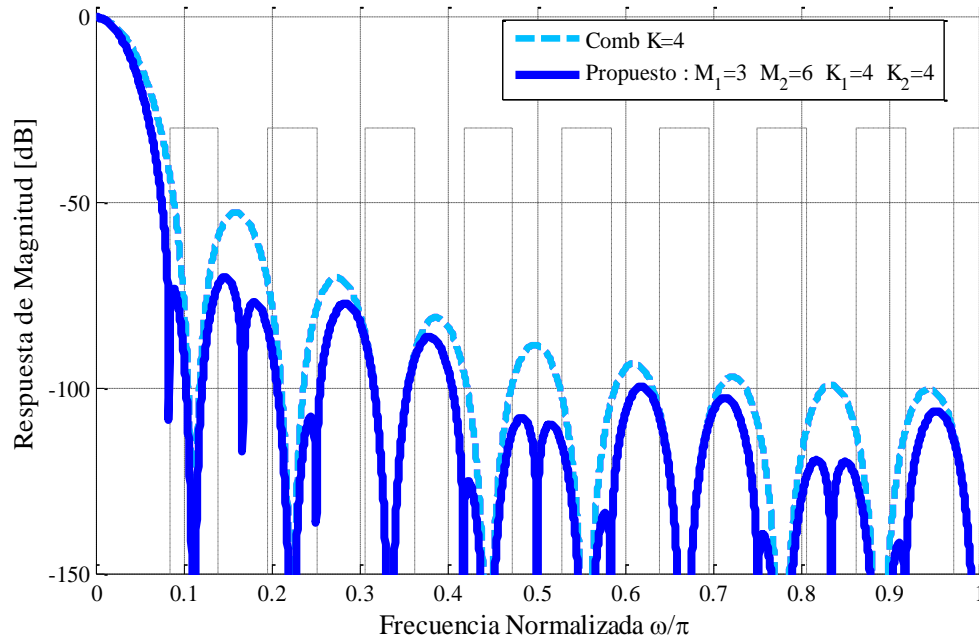


Figura 4.2 Comparación entre las respuestas magnitud del filtro propuesto y el filtro comb.

La estructura del filtro propuesto, en combinación con los respectivos submuestreadores, se puede ver en la Figura 4.3. Al principio, se encuentra la primera etapa cuya frecuencia

de operación es la mayor. El filtro comb, $H_1(z)$, y un submuestreador que reduce M_1 veces la frecuencia son los bloques que componen a la primera etapa. Posteriormente, se encuentra la segunda etapa. Nótese que su frecuencia de operación es M_1 veces menor. A consecuencia de ello, y utilizando la tercera identidad multitasa, los filtros $H_2(z^{M_1})$ y $H_S(z^{M_1})$ se pasan fácilmente a la segunda etapa, en donde su función de transferencia es $H_2(z)$ y $H_S(z)$. Finalmente, en el extremo de esta etapa se localiza su correspondiente submuestreador.

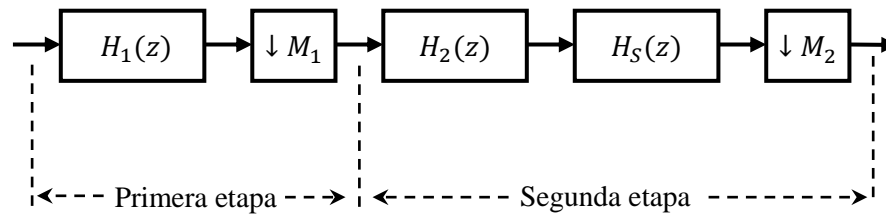


Figura 4.3 Estructura del filtro propuesto.

4.1.4 Arquitecturas eficientes

Actualmente, un gran número de aplicaciones demandan decimadores de área reducida y de bajo consumo de potencia. Por un lado, el costo de la implementación tiene dependencia directa con el área utilizada. Además, cada vez surgen más dispositivos móviles que deben reducir al mínimo su gasto de energía. Por tales razones, en esta tesis se propone dos arquitecturas eficientes.

4.1.4.1 Arquitectura eficiente cuando M es el producto de dos números enteros

En diversos trabajos se han desarrollado arquitecturas que son eficientes tanto en área utilizada como en consumo de potencia [25]. Esto se puede llevar a cabo cuando se tiene un filtro comb de dos etapas, por ello es viable realizarlo en el filtro propuesto (4.8). Como ya se ha visto en las Secciones 2.4 y 2.5, el filtro comb se puede implementar en diferentes maneras. La estructura CIC es conveniente para lograr que el área utilizada sea mínima, pero tiene como principal inconveniente el elevado consumo de potencia. Contrariamente, la arquitectura no recursiva polifásica presenta el menor consumo de potencia, empero, requiere de un área excesiva. Por lo tanto, se infiere que al utilizar una

arquitectura mixta, en la cual se combinen las dos realizaciones mencionadas, se puede conseguir un decimador de reducida área y bajo consumo de potencia.

Cuando M se expresa como el producto de dos enteros cualesquiera, se puede lograr el equilibrio en el decimador propuesto en términos de área y potencia. Primero se analiza su estructura, ya mostrada en la Figura 4.3. Al fijar la atención en la primera etapa, se puede estimar que su filtro comb trabaja a la mayor frecuencia, y que su ancho de bus es el menor. Con estos indicios, se deduce que resulta ventajoso implementar la primera etapa en la forma no recursiva polifásica, ya que a través de ésta se disminuye la frecuencia de operación del filtro, y consecuentemente también se reduce la potencia que consume. Además, el área extensa utilizada por esta realización se ve ligeramente compensada por el pequeño ancho de bus requerido. Debido a ello, se deriva que la implementación de la primera etapa debe llevarse a cabo en la forma no recursiva polifásica. Por otro lado, la segunda etapa presenta propiedades opuestas a la primera, es decir, su ancho de bus es grande, pero su frecuencia de operación es menor. Anteriormente, en la primera etapa se ha sacrificado área por potencia, en esta etapa se pretende lograr un equilibrio, y se opta por utilizar menor área a costa del consumo de potencia. La mejor manera de conseguir esto es implementado el filtro comb de esta etapa en su realización CIC. Sin embargo, en esta etapa también se encuentran dos filtros coseno, que indudablemente impactan en el área y potencia totales. Bajo tales circunstancias, se propone combinar los dos filtros coseno, de modo que el filtro resultante se pueda descomponer en sus componentes polifásicas, permitiendo su traslado a una frecuencia menor, después del segundo sumbuestrador. Con esto se logra que los filtros coseno consuman menos potencia. Para este caso, donde el factor de decimación es el producto de dos números enteros, la arquitectura propuesta para lograr el equilibrio entre área y potencia se puede ver en la Figura 4.4.

En general, el desempeño de los filtros decimadores se evalúa en términos de área y potencia. Debido a ello se han desarrollado diversas metodologías para estimar estos valores. Aquí se adopta la metodología desarrollada por Abbas y Gustafson [26], donde el área se estima de acuerdo al número de sumadores y registros utilizados por el filtro, ponderados por la longitud de palabra necesaria. Similarmente, la potencia se determina ponderando el área utilizada por la frecuencia de operación.

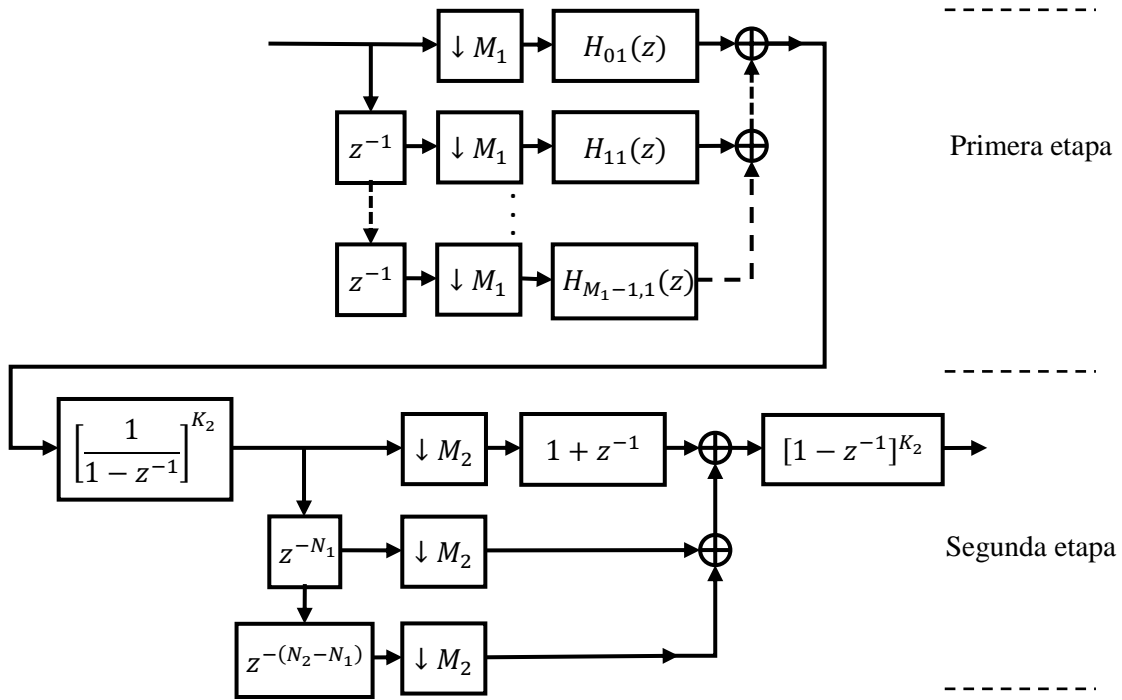


Figura 4.4 Arquitectura propuesta.

Para estimar el área y potencia del filtro propuesto, se divide el análisis en dos partes, examinando cada etapa de la arquitectura propuesta. En cuanto a la primera etapa, el número de sumadores está dado por:

$$S_{e1} = (M_1 - 1)K_1 + S_{coef} , \quad (4.11)$$

donde S_{coef} es el número de sumadores necesarios para implementar los coeficientes del filtro comb en su forma no recursiva polifásica. Debido a que estos son números enteros, fácilmente pueden realizarse mediante sumas y corrimientos. En la Tabla 4.1 se presenta los valores de S_{coef} contra M_1 y K_1 . Los valores mostrados en la Tabla 4.1 fueron obtenidos al utilizar el algoritmo de Voronenko y Püshel, el cual permite obtener el número óptimo de sumadores para la generación de los coeficientes [27]. De manera general, se advierte que S_{coef} incrementa ante cualquier aumento, ya sea de K_1 o de M_1 , sobre todo para valores grandes.

El número de registros empleados en la primera etapa, se calcula mediante:

$$R_{e1} = (M_1 - 1)K_1 + 1 . \quad (4.12)$$

Dependencia de S_{coef} contra M_1 y K_1					
$M_1 \backslash K_1$	2	3	4	5	6
2	0	2	1	2	3
3	1	3	4	9	12
4	2	6	5	18	14
5	3	10	10	22	30
6	4	14	17	32	38
7	7	17	24	33	54
8	8	18	28	50	58

Tabla 4.1 Número mínimo de sumadores necesarios para implementar los coeficientes del filtro comb con parámetros M_1 y K_1

La longitud de palabra de la primera etapa es:

$$B_{e1} = B_{in} + [K_1 \log_2(M_1)] . \quad (4.13)$$

En lo referente a la segunda etapa, el número de sumadores y registros, y la longitud de palabra se calculan a través de las siguientes expresiones:

$$S_{e2} = 2K_2 + 3 , \quad (4.14)$$

$$R_{e2} = 2K_2 + 4 , \quad (4.15)$$

$$B_{e2} = B_{e1} + [K_2 \log_2(M_2)] + 2 . \quad (4.16)$$

Con las expresiones anteriores, (4.11)-(4.16) ya se puede estimar el área de la arquitectura propuesta:

$$A_p = (2(M_1 - 1)K_1 + S_{coef} + 1)B_{e1} + (4K_2 + 7)B_{e2} . \quad (4.17)$$

Y también se puede estimar su potencia consumida mediante la siguiente expresión:

$$P_p = \frac{(2(M_1 - 1)K_1 + S_{coef} + 1)B_{e1} + 2K_2B_{e2}}{M_1} + \frac{(2K_2 + 7)B_{e2}}{M_1M_2} . \quad (4.18)$$

Para el diseño del filtro propuesto se cuenta con cuatro parámetros, los dos factores de decimación, M_1 y M_2 , y el número de filtros comb conectados en cascada en cada etapa, K_1 y K_2 . De su elección se deriva el desempeño del filtro tanto en su respuesta de magnitud como en la potencia consumida y área utilizada. Por lo tanto, hacerlo apropiadamente es muy importante. Enseguida se discute este asunto.

4.1.4.2.1 Elección de M_1 y M_2

Anteriormente se ha estipulado que M_1 y M_2 deben ser dos números enteros que multiplicados entre si den el factor de decimación deseado. Sin embargo, para un M dado, podrían existir deferentes maneras de llevar a cabo tal asignación. De modo que, en primer lugar se intenta encontrar la mejor forma de elegir M_1 y M_2 .

Al haber hecho un análisis a la magnitud del filtro propuesto, se observó que la atenuación exhibida está ligeramente relacionada con M_2 , y que el impacto que ejerce M_1 sobre ella es insignificante. Por ello, en la elección de M_1 y M_2 se considera que el efecto que tienen sobre la atenuación se puede despreciar. En cambio, el área y la potencia del filtro guardan una relación más estrecha contra los valores de M_1 y M_2 . De este modo, para efectuar la elección de estos parámetros, se pretende mantener el área y la potencia tan bajos como sea posible. Así, conviene mostrar la dependencia existente entre ellos. En las figuras 4.5 y 4.6 se han graficado las expresiones 4.17 y 4.18 para diversos valores de M , y considerando diferentes formas de factorizarlos. Nótese que el eje de las abscisas corresponde al valor de M_1 . El valor de M_2 está implícito en cada gráfica, calculándose como $M_2 = M/M_1$. Para simplificar el análisis, pero sin pérdida de generalidad, se hicieron las asignaciones $K_1 = 4$ y $K_2 = 5$. Al observar la Figura 4.5, se infiere que el menor valor de área estimada siempre se obtiene cuando M_1 es el factor más pequeño de aquellos que componen a M , y que aumenta gradualmente conforme M_1 se va haciendo más grande. En tanto que en la gráfica de la potencia, mostrada en la Figura 4.6, se puede ver que el valor mínimo se presenta cuando M_1 y M_2 son lo más cercanos entre sí, pero manteniendo la relación $M_1 < M_2$.

Teniendo en cuenta los comportamientos descritos, se propone que escoger los factores de decimación de la siguiente forma:

$$M_1 \approx M_2 \quad / \quad M_1 \leq M_2 . \quad (4.19)$$

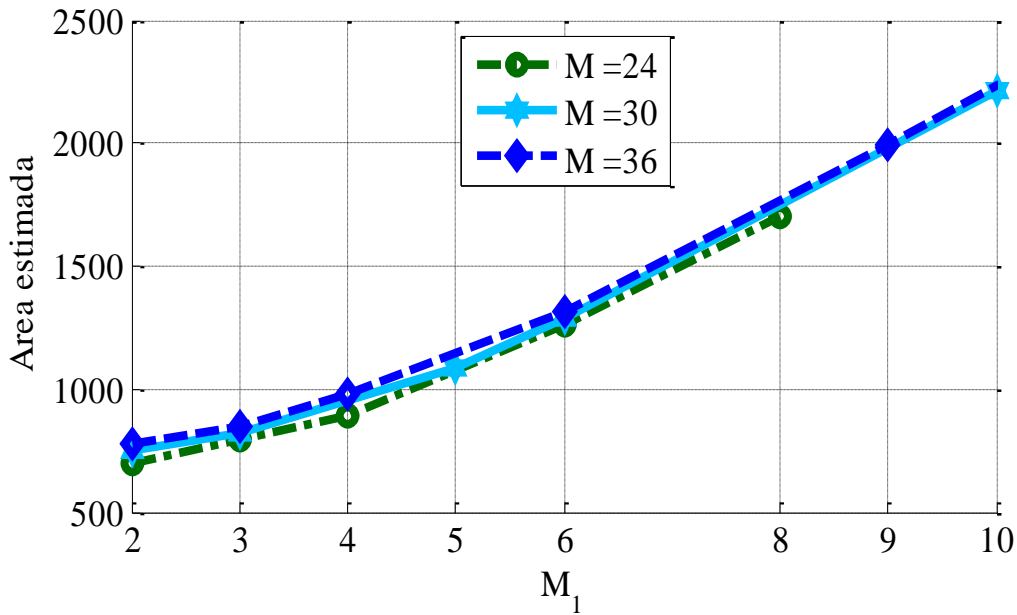


Figura 4.5 Área estimada del filtro propuesto ante diferentes valores de M y modos de factorizarlo.

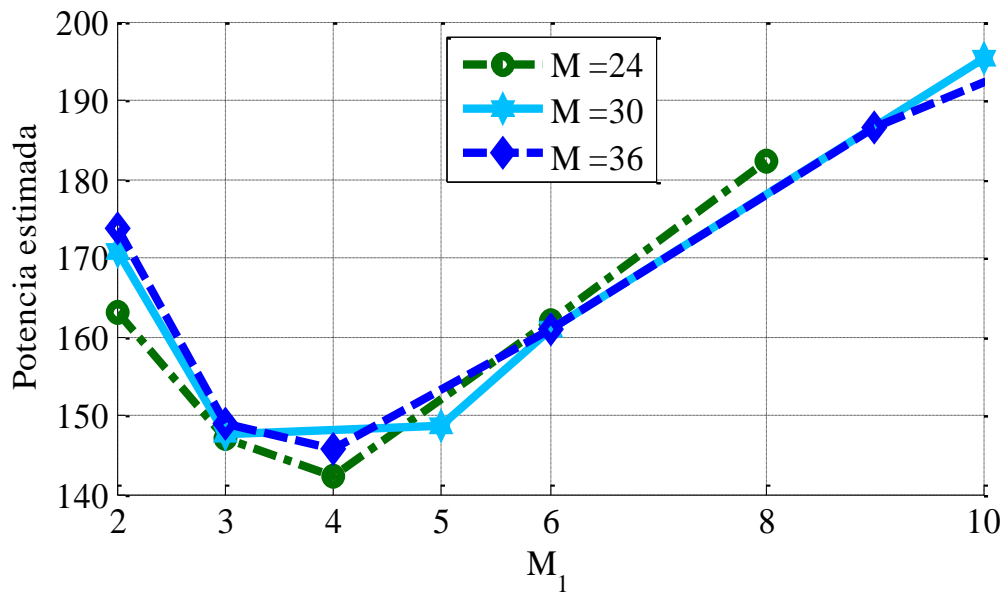


Figura 4.6 Potencia estimada del filtro propuesto ante diferentes valores de M y modos de factorizarlo.

De esta manera, la potencia consumida es la mínima y el área se mantiene en un bajo nivel. Por ejemplo, si $M = 24$, las diversas formas de elegir M_1 y M_2 se muestran en la Tabla 4.2, y con el criterio descrito anteriormente, los valores que se deben asignar son: $M_1 = 4$ y $M_2 = 6$.

M_1	M_2
2	12
3	8
4	6
6	4
8	3
12	2

Tabla 4.2 Diferentes formas de elegir M_1 y M_2 , para $M = 24$.

4.1.4.2 Arquitectura cuando M es un número potencia de dos

Cuando el factor de decimación es un número potencia de dos 2^P , la función de transferencia de un filtro comb se puede expresar como la conexión en cascada de P filtros comb más simples:

$$H(z) = \left[\prod_{l=1}^P \frac{1}{2} (1 + z^{-2^{l-1}}) \right]^K. \quad (4.20)$$

Al combinar (4.20) con P submuestreadores de factor $M = 2$, la estructura del decimador resultante se compone de P etapas de decimación por dos. Su diagrama a bloques se presenta en la Figura 4.7. Tal estructura presenta un menor consumo de potencia.

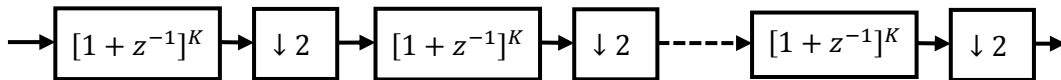


Figura 4.7 Estructura multietapa de un filtro comb con $M = 2^P$.

Para aprovechar la estructura de la Figura 4.7 se hacen las siguientes consideraciones. El filtro que se ha propuesto (4.8) se obtiene al factorizar a M en dos números enteros $M = M_1 M_2$. De este modo, al ser M un número potencia de dos, los valores que se pueden asignar a M_1 y M_2 son:

$$M_1 = 2^{p_1}; M_2 = 2^{P-p_1}. \quad p_1 \geq 1, \quad P > 2 \quad (4.21)$$

Si se considera que los filtros comb $H_1(z)$ y $H_2(z)$ se realizan en su estructura no recursiva multietapas, el filtro comb de la primera etapa se compone de p_1 sub-etapas de decimación por dos. De igual manera, el filtro comb de la segunda etapa tiene $P - p_1$ sub-etapas de decimación por dos. De este modo, en total existen P sub-etapas de decimación por dos y, de acuerdo al método propuesto, los filtros coseno (4.3) se insertan en la sub-etapa $p_1 + 1$, como se observa en la Figura 4.8.

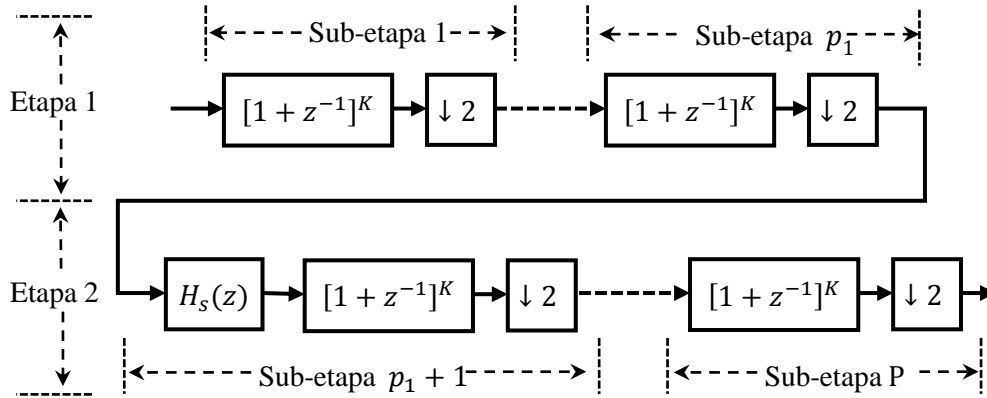


Figura 4.8 Arquitectura del filtro propuesto para $M = 2^P$.

Una reducción adicional al consumo de potencia se puede lograr al utilizar la técnica de descomposición polifásica. En la Figura 4.9 se observa la arquitectura propuesta para el filtro (4.8), cuando M es una potencia de dos. Esta arquitectura utiliza la estructura multietapa y descomposición polifásica. En los subfiltros polifásicos, el primer subíndice indica la componente polifásica y el segundo subíndice representa la sub-etapa. Como ya se ha hecho mención, los filtros coseno se encuentran en la sub-etapa $p + 1$, en la arquitectura de la Figura 4.9 se han combinado con el filtro comb de dicha sub-etapa para poder pasarlos a más baja frecuencia.

Teniendo en cuenta que $M_1 \geq 2$ y $M_2 > 2$, entonces se puede deducir que existen $P - 2$ estructuras posibles. Se llama arquitectura p_1 , si el filtro $H_S(z)$ es colocado en la sub-etapa $p_1 + 1$. A manera de ejemplo, en la Tabla 4.3 se muestran los parámetros de las arquitecturas para $P = 3, 4, 5$ y 6 .

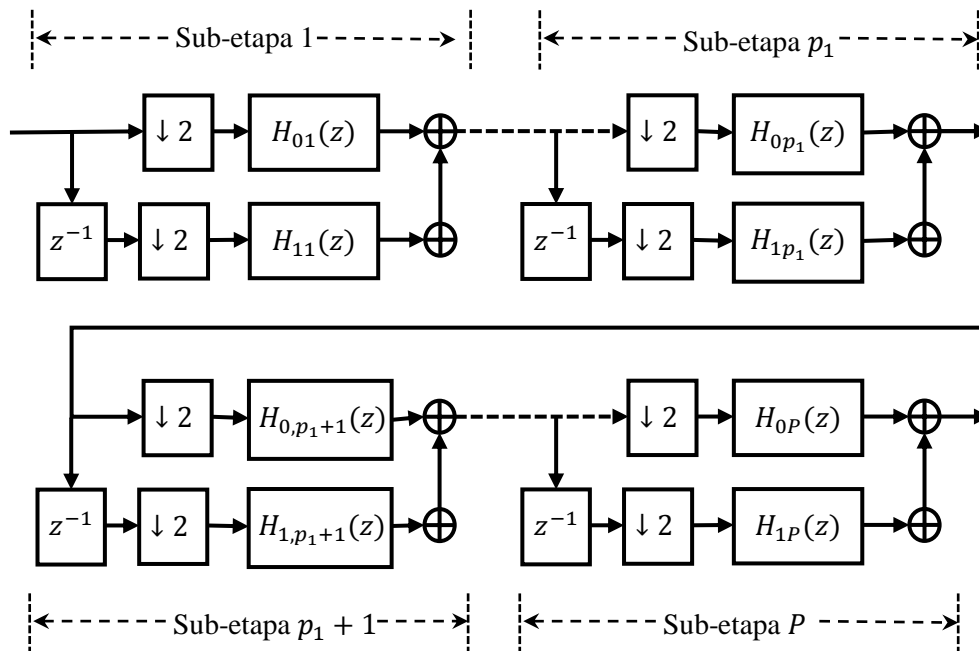


Figura 4.9 Arquitectura eficiente del filtro propuesto para $M = 2^P$.

P	M_1	M_2	p_1	$H_S(z)$ a alta frecuencia	N_1, N_2
3	2	4	1	$H_S(z^2)$	1, 3
4	2	8	1	$H_S(z^2)$	3, 5
4	4	4	2	$H_S(z^4)$	1, 3
5	2	16	1	$H_S(z^2)$	7, 9
5	4	8	2	$H_S(z^4)$	3, 5
5	8	4	3	$H_S(z^8)$	1, 3
6	2	32	1	$H_S(z^2)$	15, 17
6	4	16	2	$H_S(z^4)$	7, 9
6	8	8	3	$H_S(z^8)$	3, 5
6	16	4	4	$H_S(z^{16})$	1, 3

Tabla 4.3 Parámetros correspondientes a las diferentes estructuras para $P=3, 4, 5$ y 6 .

Ejemplo 4.3. Se considera $M = 16 = 2^4$ y $K = 4$. Utilizando la Tabla 4.3, se pueden obtener las dos arquitecturas siguientes:

- *Arquitectura 1* ($p_1 = 1$), los valores que se obtienen son: $M_1 = 2$, $M_2 = 8$, y los filtros coseno se insertan en la segunda sub-etapa de decimación por dos, siendo sus factores de expansión $N_1 = 3$ y $N_2 = 5$.
- *Arquitectura 2* ($p_1 = 2$), ambas etapas tienen $M_1 = 4 = M_2$. Esta vez los filtros coseno se insertan en la tercera sub-etapa de decimación por dos, y sus factores de expansión son $N_1 = 1$ y $N_2 = 3$.

Cada arquitectura tiene 4 sub-etapas de decimación por dos, como se ve en la Figura 4.10, donde se representa de forma general las dos arquitecturas.

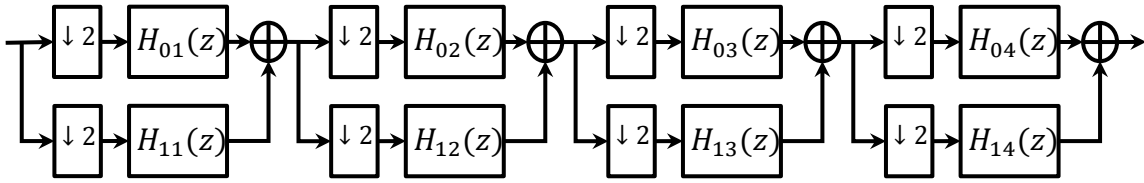


Figura 4.10 Arquitectura general para los filtros del ejemplo 4.3.

La diferencia entre la *Arquitectura 1* y la *Arquitectura 2* radica en las funciones de transferencia de los subfiltros polifásicos. Para la *Arquitectura 1*, los subfiltros polifásicos son:

$$H_{01}(z) = 1 + 6z^{-1} + z^{-2} = H_{03}(z) = H_{04}(z), \quad (4.22)$$

$$H_{11}(z) = 4(1 + z^{-1}) = H_{13}(z) = H_{14}(z), \quad (4.23)$$

$$H_{02}(z) = [1 + 6z^{-1} + z^{-2}][1 + z^{-4}] + z^{-1}[4(1 + z^{-1})][z^{-1} + z^{-2}], \quad (4.24)$$

$$H_{12}(z) = [4(1 + z^{-1})][1 + z^{-4}] + [1 + 6z^{-1} + z^{-2}][z^{-1} + z^{-2}]. \quad (4.25)$$

Similarmente, para la *Arquitectura 2* se tiene:

$$H_{01}(z) = 1 + 6z^{-1} + z^{-2} = H_{02}(z) = H_{04}(z), \quad (4.26)$$

$$H_{11}(z) = 4(1 + z^{-1}) = H_{12}(z) = H_{14}(z), \quad (4.27)$$

$$H_{03}(z) = [1 + 6z^{-1} + z^{-2}][1 + z^{-2}] + z^{-1}[4(1 + z^{-1})][1 + z^{-1}] \quad (4.28)$$

$$H_{13}(z) = [4(1 + z^{-1})][1 + z^{-2}] + [1 + 6z^{-1} + z^{-2}][1 + z^{-1}] \quad (4.29)$$

Las respuestas de magnitud de los dos filtros resultantes se pueden ver en la Figura 4.11, donde también se muestra la magnitud del filtro comb con $K = 4$. Nótese que la *Arquitectura 2* presenta mayor atenuación en las bandas de rechazo 2, 3, 5 y 6. No obstante, la *Arquitectura 1* tiene mayor rechazo antialiasng en las bandas de rechazo número 1, 4 y 7, y su PCA es 8 dB mayor.

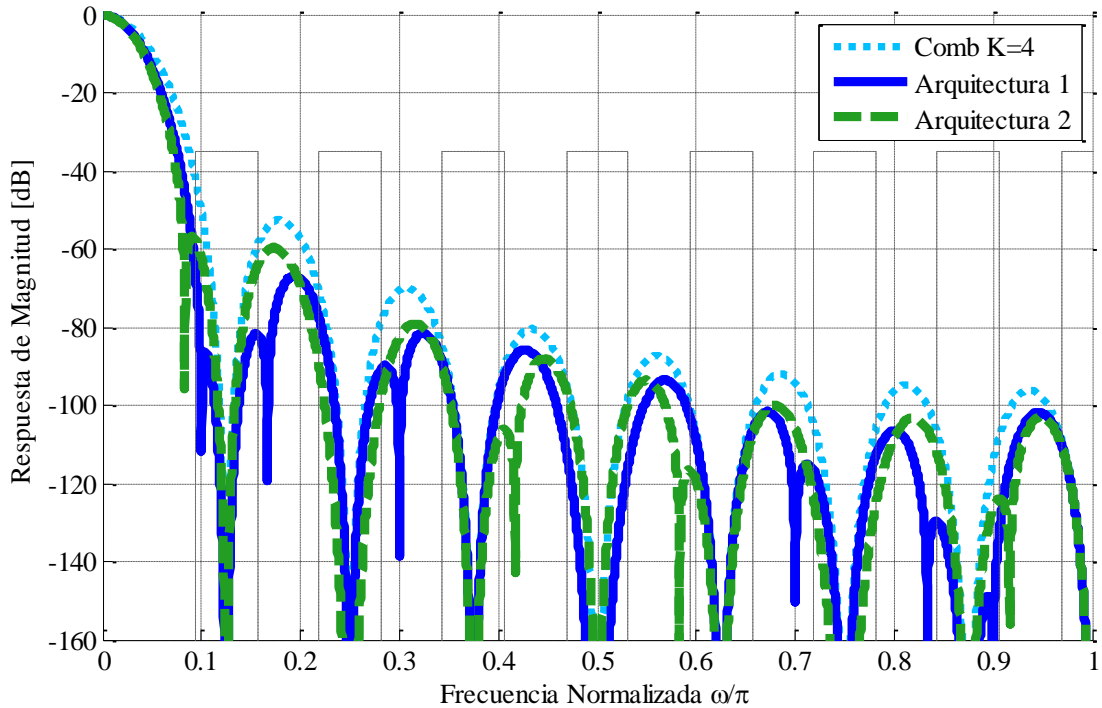


Figura 4.11 Respuestas de magnitud de los filtros del ejemplo 4.3.

Ejemplo 4.4. Se considera $M = 32 = 2^5$ y $K = 3$. De acuerdo con la Tabla 4.3, las arquitecturas posibles son tres:

- Arquitectura 1 ($p_1 = 1$), esta arquitectura se deriva al escoger $M_1 = 2$, $M_2 = 16$. Los filtros coseno se insertan en la segunda sub-etapa de decimación por dos, y tienen $N_1 = 7$ y $N_2 = 9$.
- Arquitectura 2 ($p_1 = 2$), para este caso, se elige $M_1 = 4$, $M_2 = 8$, y es en la tercera sub-etapa de decimación por dos donde se insertan los filtros coseno cuyos parámetros son $N_1 = 3$ y $N_2 = 5$.

- Arquitectura 3 ($p_1 = 3$), en esta arquitectura se tiene $M_1 = 2$, $M_2 = 16$ y los filtros coseno se insertan en la cuarta sub-etapa de decimación por dos, teniendo $N_1 = 1$ y $N_2 = 3$.

En la Figura 4.12 se muestra una comparación entre las respuestas de magnitud de los filtros obtenidos con las tres arquitecturas anteriores. Se advierte que la atenuación exhibida en cada folding band depende de la arquitectura, aunque, en general, la arquitectura 2 proporciona mayor atenuación en la mayoría de las bandas de rechazo, además, el PCA de la arquitectura 2 es hasta 8 dB más grande en relación con las otras arquitecturas.

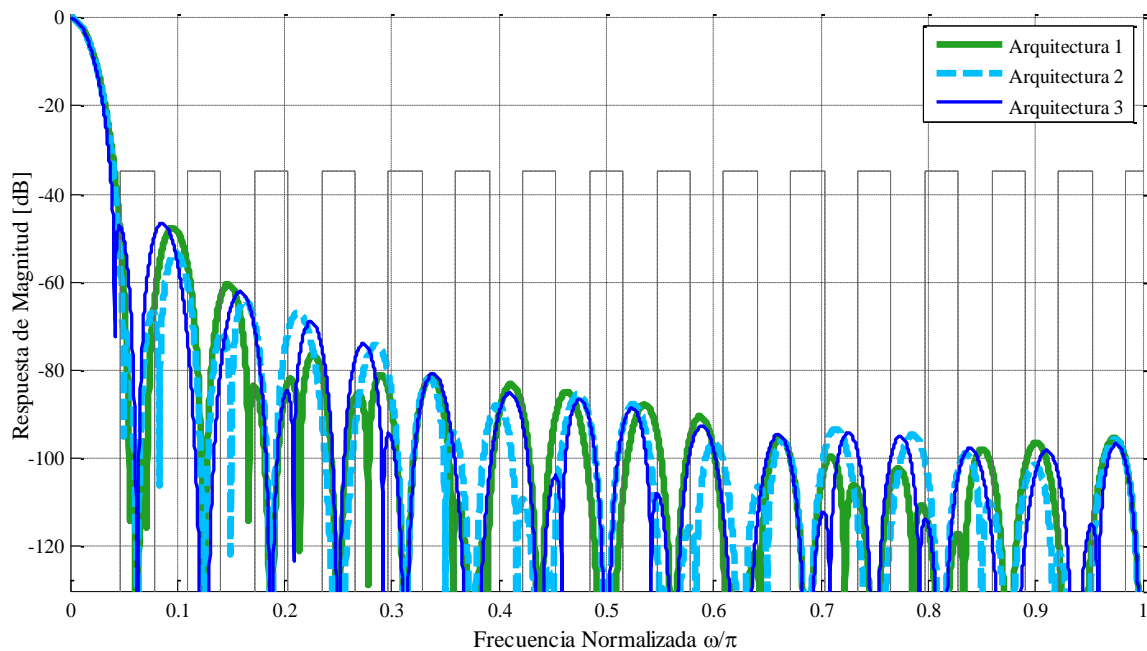


Figura 4.12 Respuestas de magnitud de los filtros del ejemplo 4.4.

4.2 Disminución de la caída en la banda de paso

El filtro (4.8) que se ha propuesto exhibe mejores características en las bandas de rechazo, pero presenta una gran caída en la banda de paso que indudablemente distorsiona la señal que se decima. De esta manera, el objetivo es disminuir la caída utilizando un filtro sin multiplicadores que trabaje a la frecuencia más baja. Con este propósito, se propone utilizar el filtro compensador ya explicado en el capítulo 3 [26] y adaptarlo a los requerimientos del filtro propuesto en la sección 4.1. Se opta por este

filtro ya que su magnitud es muy próxima a la respuesta de magnitud inversa del filtro propuesto. Además su simplicidad es mayor en relación a otros compensadores.

Nuevamente se escribe la función de transferencia y la respuesta de magnitud del filtro compensador:

$$G(z^M) = 2^{-2}[B(-1 + 2z^{-M} - z^{-2M}) + 2^2z^{-M}]. \quad (4.30)$$

$$G(e^{j\omega M}) = 1 + B \cdot \sin^2\left(\frac{\omega M}{2}\right) \quad (4.31)$$

Cabe recordar que este compensador solo cuenta con un parámetro de diseño, B . En el apartado 3.3 se presentó cómo elegirlo de acuerdo al orden del filtro comb, a través de una tabla. No obstante, en nuestro caso la inserción de los filtros coseno en la segunda etapa, que fundamentalmente se hizo con la finalidad de incrementar el rechazo antialiasing, incrementa aún más la caída en la banda de paso. Por lo que se deben obtener nuevos valores del parámetro B , que se ajusten al filtro propuesto. Así, mediante el software Matlab se desarrolló un algoritmo, basado en análisis intervalar, capaz de hallar los valores óptimos de B que minimizan la desviación en la banda de paso. Estos valores dependen fuertemente del parámetro K_2 y en menor medida de M_2 , sobre todo cuando este es 4. Las variaciones de M_1 y K_1 afectan muy poco en su elección. En la Tabla 4.4 se presentan de manera concisa los resultados obtenidos. Adviértase que los valores de B están expresados como sumas de potencias de dos para evitar multiplicadores. Asimismo, se puede ver la máxima desviación que se consigue.

K_2	B	<i>desv</i>
2	$0.875 = 1 - 2^{-3}$	< 0.3 dB
3	$1.125 = 1 + 2^{-3}$	≈ 0.36 dB
4	$1.4375 = 1 + 2^{-1} - 2^{-4}$	≈ 0.5 dB
5	$1.75 = 2 - 2^{-2}$	≈ 0.6 dB

Tabla 4.4 Valores de B .

4.3 Decimador propuesto

El decimador que se propone en esta tesis, se obtiene al conectar en cascada los filtros $H_p(z)$ y $G(z^M)$, obteniéndose:

$$\begin{aligned}
 H_{PC}(z) &= H_p(z)G(z^M) \\
 &= \left[\frac{1}{M_2} \frac{1 - z^{-M_1}}{1 - z^{-1}} \right]^{K_2} \left[\frac{1}{M_2} \frac{1 - z^{-M_1 M_2}}{1 - z^{-M_1}} \right]^{K_2} \left[\frac{1 + z^{-M_1 N_1}}{2} \right] \left[\frac{1 + z^{-M_1 N_2}}{2} \right] \\
 &\quad \left[2^{-2} \left[B(-1 + 2z^{-M} - z^{-2M}) + 2^2 z^{-M} \right] \right] \quad (4.32)
 \end{aligned}$$

Su respectiva respuesta de magnitud está dada por:

$$\begin{aligned}
 |H_{PC}(e^{j\omega})| &= \\
 &= \left| \frac{1}{M_1^{K_1} \cdot M_2^{K_2}} \cdot \left[\frac{\sin\left(\frac{\omega M_1}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right]^{K_1} \left[\frac{\sin\left(\frac{\omega M_2 \cdot M_1}{2}\right)}{\sin\left(\frac{\omega M_1}{2}\right)} \right]^{K_2} \cos\left(\frac{\omega N_1 M_1}{2}\right) \cos\left(\frac{\omega N_2 M_1}{2}\right) \left[1 + B \cdot \sin^2\left(\frac{\omega M}{2}\right) \right] \right| \quad (4.33)
 \end{aligned}$$

La estructura del decimador propuesto se observa en la Figura 4.13, donde únicamente se ha agregado el compensador al final.

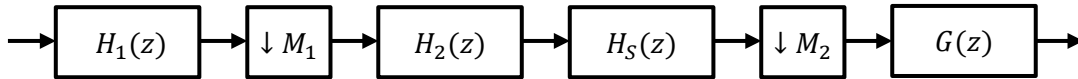


Figura 4.13 Estructura del decimador propuesto

Mediante el siguiente ejemplo se muestran las ventajas del decimador propuesto.

Ejemplo 4.5. Considérese un decimador comb con $M = 20$ y $K = 3$. Al aplicar el método propuesto se obtiene: $M_1 = 4$, $M_2 = 5$, $N_1 = 2$, $N_2 = 3$, $K_1 = 3$, $K_2 = 4$ y $B = 1.75$. La arquitectura eficiente se muestra en la Figura 4.14. En ella las componentes polifásicas se definen de la siguiente manera:

$$H_{01}(z) = 1 + 2z^{-1} + 3z^{-2}, \quad (4.34)$$

$$H_{11}(z) = 3 + 12z^{-1} + z^{-2}, \quad (4.35)$$

$$H_{21}(z) = 6 + 10z^{-1}, \quad (4.36)$$

$$H_{31}(z) = 10 + 6z^{-1}. \quad (4.37)$$

En la Figura 4.15 se presentan las respuestas de magnitud del filtro comb convencional y del filtro propuesto. Se observa que el filtro propuesto presenta mayor rechazo antialiasing en la mayoría de las bandas de rechazo, excepto en las bandas quinta y decima. Además, la desviación en banda de paso del filtro propuesto es menor a 0.47 dB. En cambio, la DBP del filtro comb supera 2.7 dB.

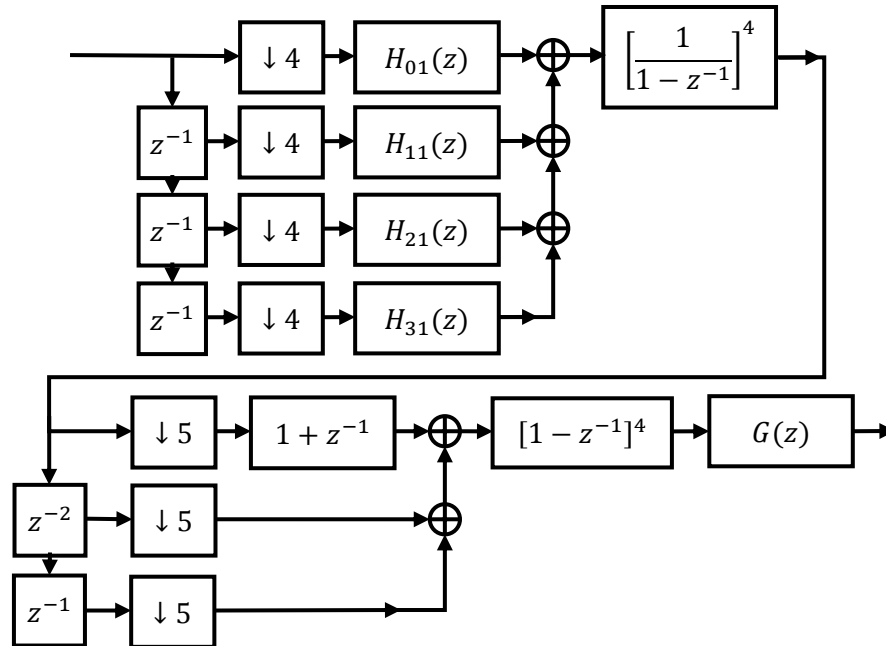


Figura 4.14 Arquitectura del filtro del Ejemplo 4.4.

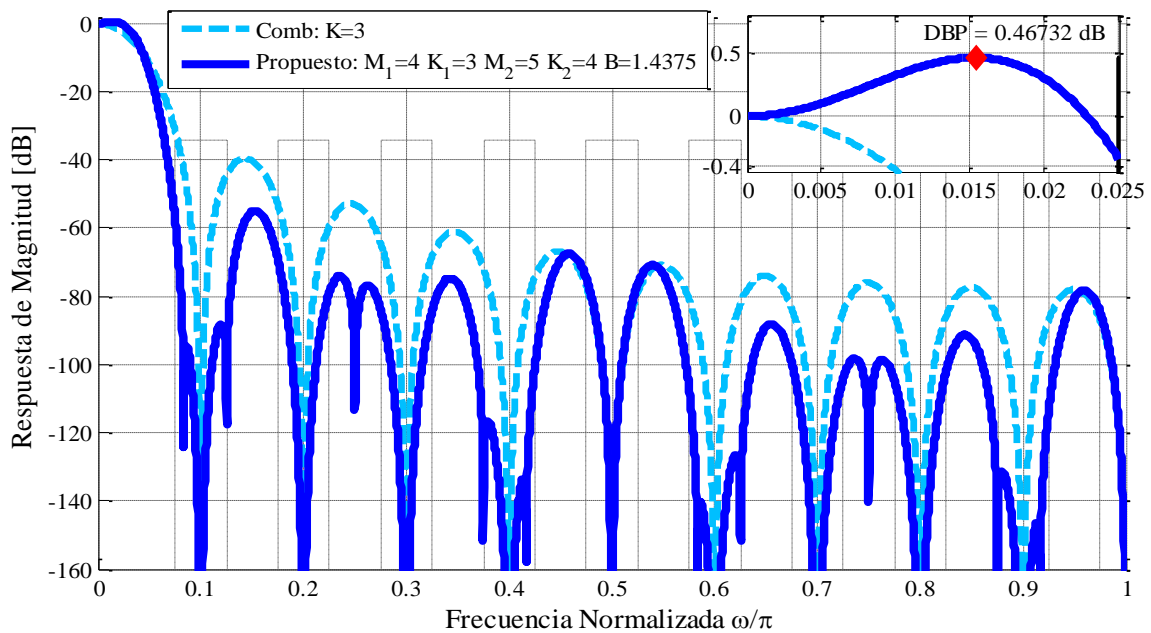


Figura 4.15 Magnitud del filtro del Ejemplo 4.4.

4.3.1 Comparación con otros métodos

En este apartado se evalúa el desempeño del filtro propuesto. Por esta razón, se compara con algunas metodologías que ya se han presentado en el capítulo 3.

4.3.1.1 Comparación con el método [18]

El filtro propuesto se compra con el filtro del método [18], para ello se considera $M = 18$. Se utilizan los siguientes parámetros para el filtro del método [18]: $K = 4$, $L = 1$, $N = 9$, $b = -1$. Por su parte, para el filtro propuesto en esta tesis se elige: $M_1 = 3$, $M_2 = 6$, $N_1 = 2$, $N_2 = 4$, $K_1 = 4$, $K_2 = 5$, $B = 1.75$. En la Figura 4.16 se han graficado las respuestas de magnitud de ambos filtros, se puede ver que la atenuación exhibida en la mayoría de las bandas de rechazo es mayor en el filtro propuesto. Únicamente, las bandas número tres, seis y nueve tienen atenuaciones similares. El PCA es mayor a 80 dB en el filtro propuesto, mientras que en el filtro del método [18] es menor a 60 dB. En lo que respecta a la banda de paso, el filtro propuesto tiene una desviación máxima de 0.62 dB, pero el filtro del método [18] tiene una enorme caída cercana a 2 dB. Por otra parte, la utilización de recursos es mayor en el filtro del método [18], pues utiliza al menos 76 sumadores, y el método propuesto requiere únicamente 28 sumadores.

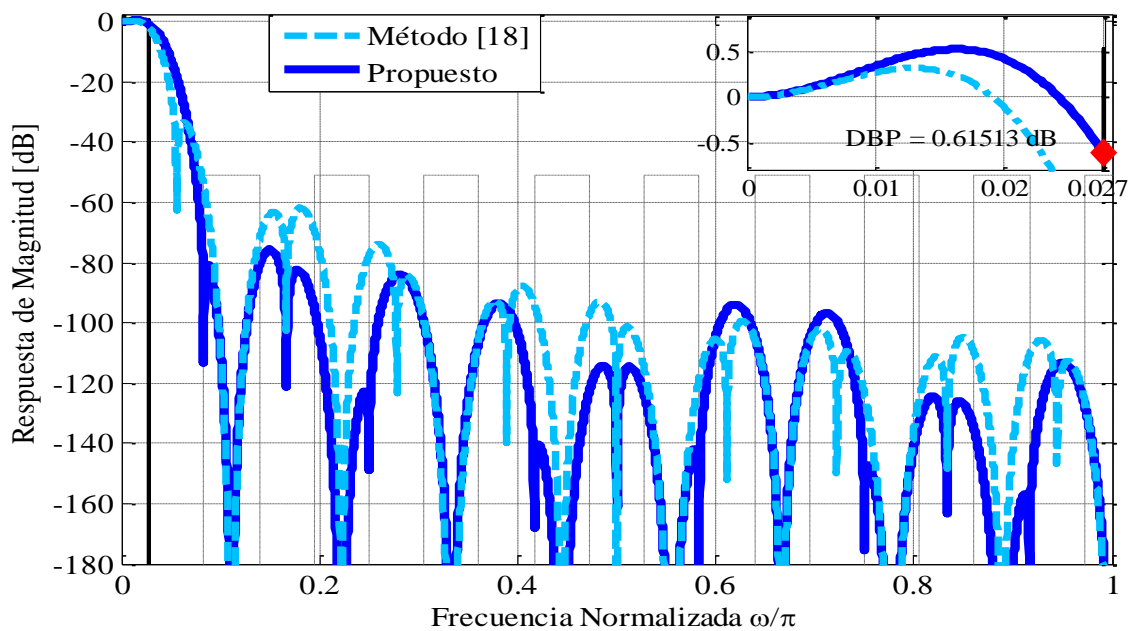


Figura 4.16 Comparación entre el filtro propuesto con método [18].

4.3.1.2 Comparación con el filtro del método [20]

Esta vez se compara el filtro propuesto con el filtro del método [20]. Se utiliza un factor de decimación $M = 32$. Los parámetros del filtro del método [20] son: $K = 4$, $R = 2$, $\alpha = 0.0491$ y $B = 1.75$. Para el filtro propuesto se asignan los siguientes valores: $M_1 = 4$, $M_2 = 8$, $N_1 = 3$, $N_2 = 5$, $K_1 = 4$, $K_2 = 5$, $B = 1.75$. Las respuestas de magnitud de estos filtros se pueden ver en la Figura 4.17. Nótese que la atenuación entre la mayoría de las bandas de rechazo es muy similar, pero el filtro del método [20] presenta menor atenuación en las bandas cuarta y decimosegunda. En cuanto a la desviación de banda de paso se aprecia que en el filtro propuesto es menor a 0.56 dB, y en el filtro del método [20] es de aproximadamente 0.7 dB.

En la utilización de recursos, ambos filtros requieren 34 sumadores. No obstante, en el filtro propuesto 12 de sus sumadores trabajan a la frecuencia más baja y los 22 sumadores restantes operan a una frecuencia 4 veces menor que la frecuencia más alta de entrada. Entre tanto, en el filtro del método [20] únicamente tres sumadores operan a la frecuencia más baja. Los otros sumadores se distribuyen en cinco etapas con diferentes frecuencias de trabajo. En la primera etapa la frecuencia de operación es la mayor, posteriormente, se va reduciendo dos veces de etapa en etapa, hasta que en la quinta etapa es 16 veces menor.

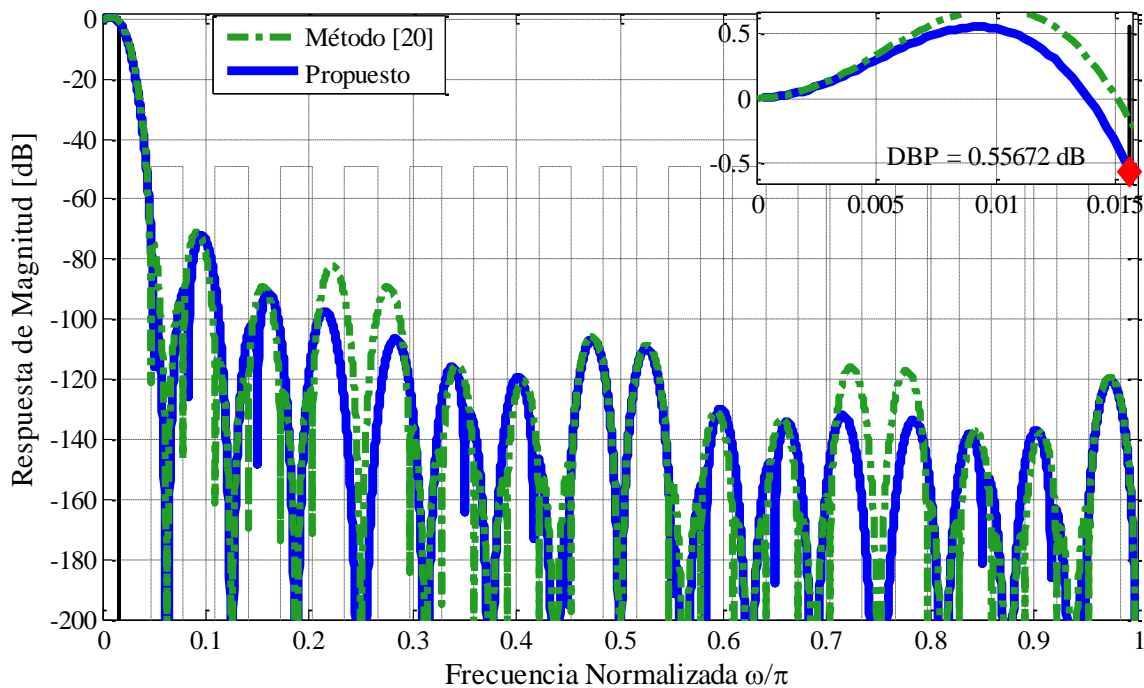


Figura 4.17 Comparación entre el filtro propuesto con método [20].

4.3.1.3 Comparación con el método [21]

Finalmente, se hace una comparación con el método [21]. Para ello, se considera $M = 27$. En el filtro del método [21] se tiene: $K = 3$, $N_1 = 13$, $N_2 = 14$ y $b = 0$. Para el método propuesto se elige: $M_1 = 3$, $M_2 = 9$, $N_1 = 4$, $N_2 = 5$, $K_1 = 3$, $K_2 = 4$ y $B = 1.4375$. Las respuestas de magnitud se presentan en la Figura 4.18. Se observa que el filtro propuesto presenta mayor rechazo antialiasing en todas las bandas de rechazo, menos en la novena. En la banda de paso, el filtro propuesto tiene una desviación menor a 0.47 dB, pero el filtro del método [21] tiene una desviación cercana a 0.6 dB. En ambos filtros se insertan dos filtros coseno, éstos trabajan a la más baja frecuencia en el método propuesto, en cambio en el filtro del método [21] operan a la frecuencia más alta. El número total de sumadores requeridos por el filtro propuesto son 25. En el filtro del método [21] se requieren al menos 50 sumadores, es decir, el doble de los requeridos por el filtro propuesto.

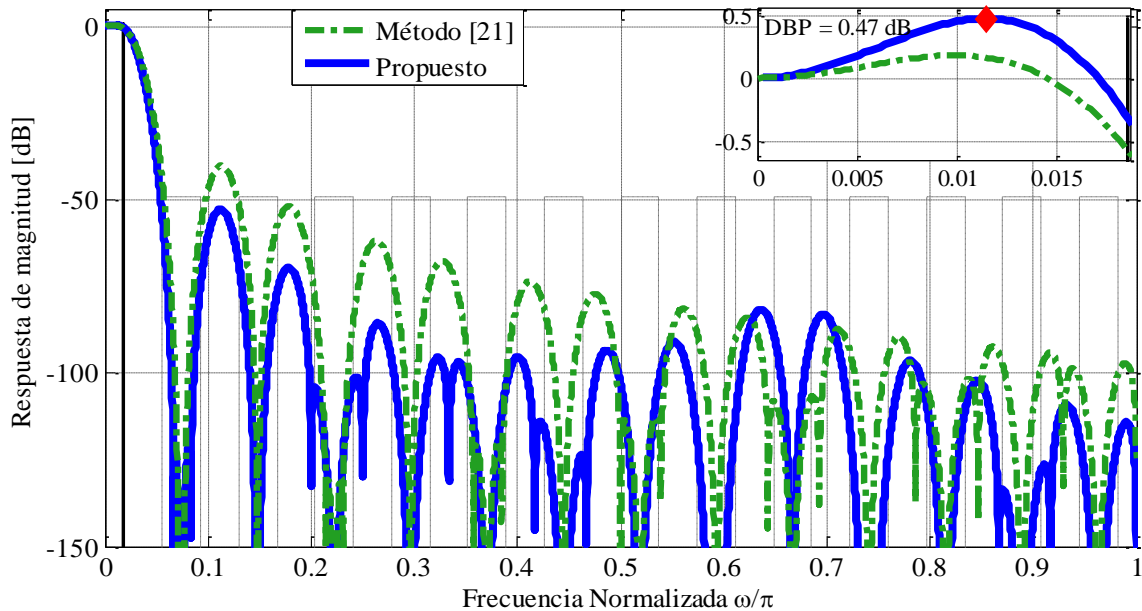


Figura 4.18 Comparación entre el filtro propuesto con método [21].

A través de las comparaciones realizadas, se advierte el filtro propuesto utiliza una cantidad menor de recursos para obtener similares o mejores respuestas de magnitud que los filtros reportados. Además, exhibe mejores características en la banda de paso.

Capítulo 5

Implementación y resultados

En este capítulo se presenta la descripción en VHDL de los decimadores propuestos. La correcta operación de los filtros se verifica a través de simulaciones. Finalmente, el código VHDL se sintetiza en ISE Design Suite para obtener dos resultados de interés, los recursos utilizados y una estimación de la potencia consumida.

5.1 Descripción en VHDL del decimador propuesto

Actualmente, los dispositivos conocidos como FPGA (Filed Programmable Gate Array) son una opción muy atractiva para implementar muchos algoritmos de frontera de procesamiento digital de señales. Los FPGA exhiben muchas características en común con los ASIC: tienen área reducida, baja disipación de potencia, su rendimiento y velocidad de operación es elevado y el costo del dispositivo es bajo. En cambio, los FPGA tienen ventajas sobre los ASIC, tales como la programación en sistema, y la reducción en los costos y tiempo de desarrollo. Por otra parte, en comparación con los procesadores de señales programables (PDSP), los FPGA ofrecen mayor rendimiento, debido a su paralelismo. Por estas razones, en esta tesis se prefiere utilizar a los FPGA para evaluar el desempeño de los decimadores propuestos.

Puesto que se decidió realizar la implementación de los decimadores propuestos en un FPGA, es conveniente emplear el lenguaje VHDL para modelarlos, ya que es uno de los lenguajes más utilizados para describir y simular circuitos digitales. Además, VHDL es independiente de la tecnología y del fabricante de los dispositivos.

En la sección 4.1.4 se presentaron las dos arquitecturas del decimador propuesto. La primera arquitectura logra un equilibrio en el área utilizada y la potencia consumida. Por otro lado, la segunda arquitectura es apropiada en aplicaciones de bajo consumo de potencia y es válida para factores de decimación que son potencia de dos. En la parte final de ambas arquitecturas se encuentra el filtro compensador. Enseguida se describe con mayor detalle cada arquitectura, y se realiza su descripción en VHDL.

5.1.1 Arquitectura para M es el producto de dos enteros

Cuando el factor de decimación se representa como el producto de dos enteros, la arquitectura propuesta consiste de dos etapas de decimación, en ella se utilizan las realizaciones CIC y no recursiva polifásica del filtro comb.

5.1.2.1 Descripción de la primera etapa de decimación

En la primera arquitectura propuesta, se ha determinado implementar la primera etapa de decimación en forma no recursiva y aprovechar la técnica de descomposición polifásica para disminuir la potencia consumida. En la Figura 5.1(a) se muestra nuevamente la arquitectura de la primera etapa. Se puede ver que se compone de un bloque de retrasos conectados a un bloque de submuestreadores y al final se encuentran los filtros polifásicos. En sistemas prácticos, la combinación de retrasos con submuestreadores se implementa a través de un demultiplexor registrado en sus salidas. Esto se ilustra en la Figura 5.1(b). Se advierte que el selector del demultiplexor, el cual indica la salida a la cual se debe asignar la señal de entrada, es controlado por un contador ascendente de 0 a $M_1 - 1$. En el Anexo 1 a) se muestra el código de VHDL que describe la arquitectura del demultiplexor registrado.

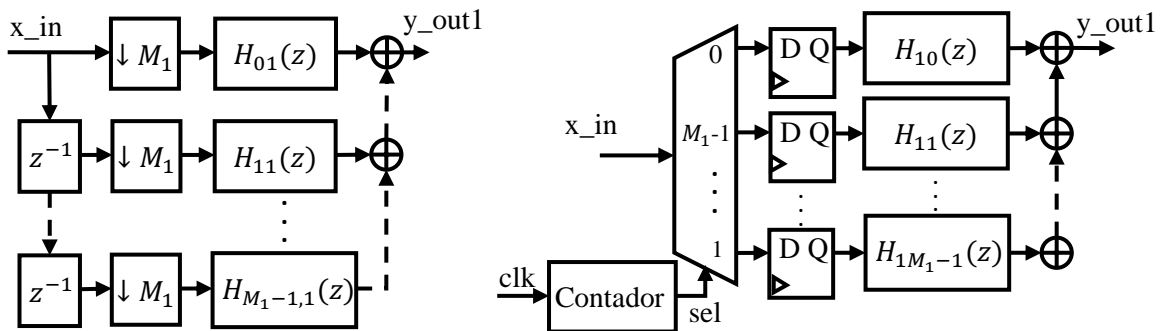


Figura 5.1 (a)Arquitectura de la primera etapa del decimador propuesto. (b)Modelo equivalente RTL usando un demultiplexor en la entrada.

En la Figura 5.1(b) se observa que después del demultiplexor registrado se encuentran los filtros polifásicos. Existen diversas realizaciones para implementar un filtro no recursivo. Las dos principales realizaciones son la forma directa y la forma directa transpuesta. En esta tesis se opta por utilizar la estructura directa transpuesta debido a los beneficios que proporciona. Por un lado, la realización directa transpuesta produce un pipeline automático al final de cada sumador, con lo cual disminuye la trayectoria crítica, y consecuentemente incrementa la máxima frecuencia de operación del filtro.

Adicionalmente, en dicha estructura la señal de entrada se multiplica simultáneamente por todos los coeficientes del filtro. De manera que, es posible utilizar alguna técnica para la generación de los coeficientes que involucren únicamente sumadores y corrimientos, y a su vez, que requieran la mínima cantidad de recursos para su implementación. Esta última observación resulta importante para lograr una reducción en el área y potencia de un filtro, pues, como es bien sabido, los multiplicadores son los elementos que gastan mayor potencia y ocupan una gran área. En la figura 5.3 se muestra la estructura directa transpuesta.

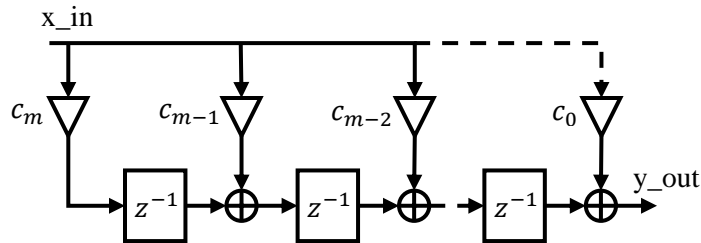


Figura 5.2 Realización de un filtro no recursivo en su forma directa transpuesta.

Como se puede ver en la Figura 5.2, resulta conveniente realizar el filtro en dos procesos en VHDL. El primer proceso tiene por finalidad describir el bloque de multiplicadores. Se ha mencionado que los coeficientes del filtro son números enteros, y que existen diferentes algoritmos para la generación de tales coeficientes que utilizan solo corrimientos y sumadores. Con base en esto, las multiplicaciones de la señal por los coeficientes del filtro se implementan siguiendo el algoritmo de Voronenko y Püshel, el cual permite efectuar tales operaciones, pero utilizando la cantidad más pequeña de sumadores y corrimientos.

El siguiente ejemplo ayuda a entender cómo generar los bloques generadores de multiplicaciones.

Ejemplo 5.1. Se desea implementar el decimador propuesto con los siguientes parámetros en la primera etapa: $M_1 = 4$, $K_1 = 3$. De modo que se tiene cuatro filtros polifásicos:

$$H_{01}(z) = 1 + 2z^{-1} + 3z^{-2}, \quad (5.1)$$

$$H_{11}(z) = 3 + 12z^{-1} + z^{-2}, \quad (5.2)$$

$$H_{21}(z) = 6 + 10z^{-1}, \quad (5.3)$$

$$H_{31}(z) = 10 + 6z^{-1}. \quad (5.4)$$

En la figura 5.3 se puede ver el diagrama RTL del decimador realizando los filtros polifásicos en la estructura directa transpuesta. Para alcanzar una mayor frecuencia de operación, se han registrado los sumadores que no contaban con el registro automático de la estructura. Esto se indica en la figura utilizando líneas puenteadas en tales registros.

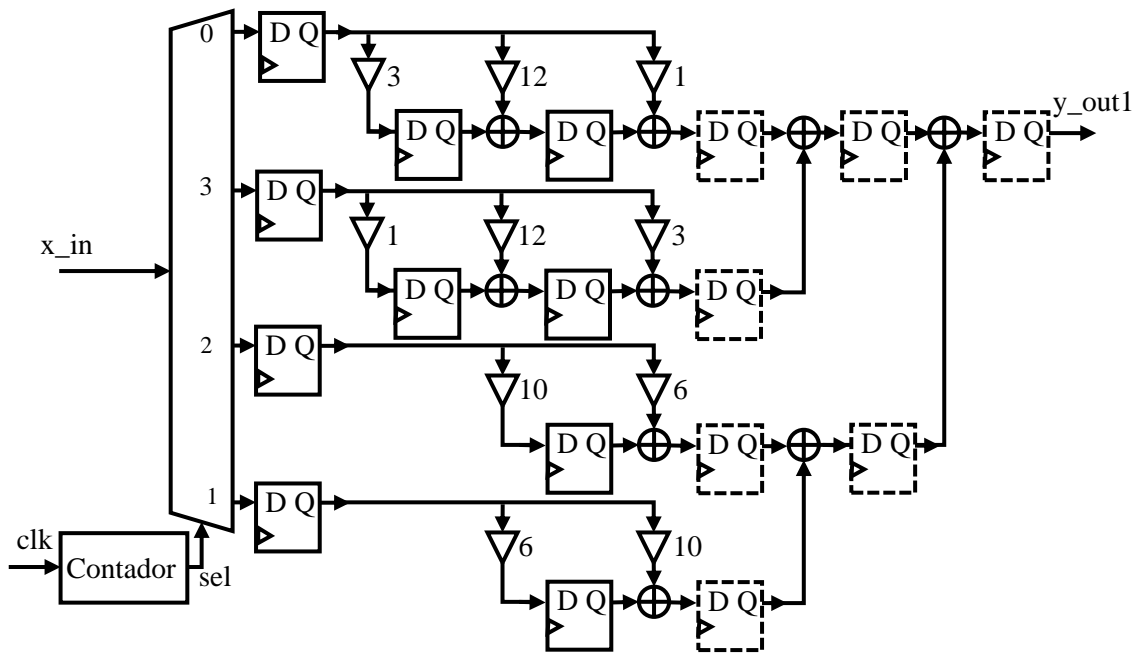


Figura 5.3 Diagrama RTL de la primera etapa del decimador propuesto considerando $M_1 = 4$ y $K_1 = 3$.

Se advierte que al principio de cada filtro polifásico la señal de entrada se debe multiplicar por los respectivos coeficientes. Al emplear la técnica de Voronenko y Püshel, en lugar de multiplicadores se utilizan bloques más simples para efectuar las multiplicaciones. Por ejemplo, el subfiltro $H_{01}(z)$ requiere multiplicar la señal por las constantes 3, 12. De tal forma que en lugar de utilizar dos multiplicadores, únicamente se necesitan dos corrimientos y un restador, como se ve en la figura 5.4(a), donde se presenta el bloque equivalente para efectuar las multiplicaciones. De manera similar, el subfiltro $H_{21}(z)$ ocuparía dos multiplicadores para ponderar la señal por las constantes 6 y 10. No obstante, con el uso de un sumador, un restador y tres corrimientos es posible llevar a cabo las mismas operaciones. Esto último se puede ver en la figura 5.4(b).

El código de VHDL que describe las multiplicaciones de los filtros polifásicos del ejemplo 5.1 se muestra en el Anexo 1 b), donde se puede notar que se ha evitado el uso

de multiplicadores y únicamente se han utilizado corrimientos y sumadores. Se ha utilizado un formato numérico de complemento a dos y punto fijo. Dicho formato se utiliza en todas las descripciones que se hacen de las arquitecturas propuestas.

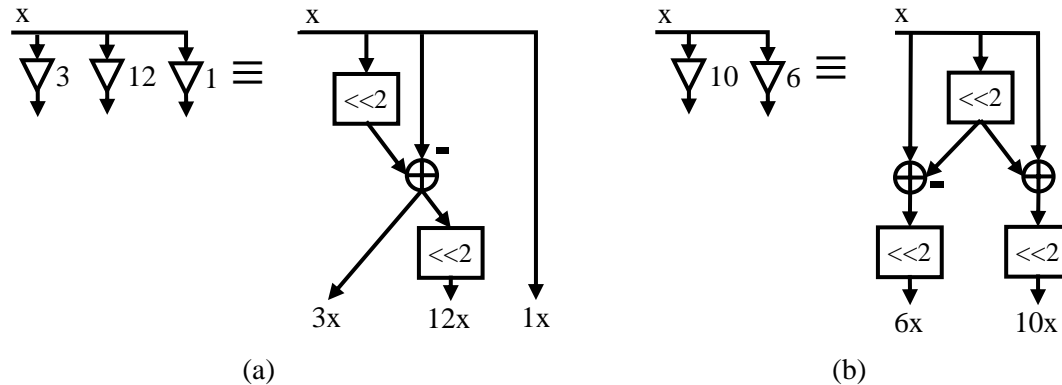


Figura 5.4 (a) Multiplicadores necesarios para el subfiltro polifásico $H_{01}(z)$ y su modelo equivalente utilizando sumas y corrimientos. (b) Multiplicadores necesarios para el subfiltro polifásico $H_{21}(z)$ y su modelo equivalente utilizando sumas y corrimientos.

Una vez que la señal ya se ha ponderado por los coeficientes, el segundo proceso que se debe realizar en VHDL es la descripción de los retrasos y las sumas correspondientes a la estructura del filtro. Dado que todas las sumas se han registrado, se debe tener especial atención en el momento de llevarlas a cabo. Para tener una latencia mínima, la clave es sincronizar adecuadamente cada operación. En el listado del Anexo 1 c) se presenta la descripción de la arquitectura en VHDL, correspondiente a las sumas y los retrasos de la estructura del filtro del ejemplo 5.1.

Con este proceso ha terminado la descripción en VHDL de la primera etapa del decimador propuesto. La señal se ha procesado por un filtro comb y submuestreado por el factor M_1 . A continuación se hace la descripción de la segunda etapa.

5.1.1.2 Descripción de la segunda etapa de decimación

En la segunda etapa de la primera arquitectura propuesta, se busca compensar el área utilizada por la primera etapa y debido a ello se implementa su filtro comb en la forma CIC. Con respecto a los filtros coseno, se optó por disminuir su consumo de potencia y se han pasado después del submuestreador de la segunda etapa al obtener sus componentes polifásicas. En la Figura 5.5 se muestra el decimador de la segunda etapa.

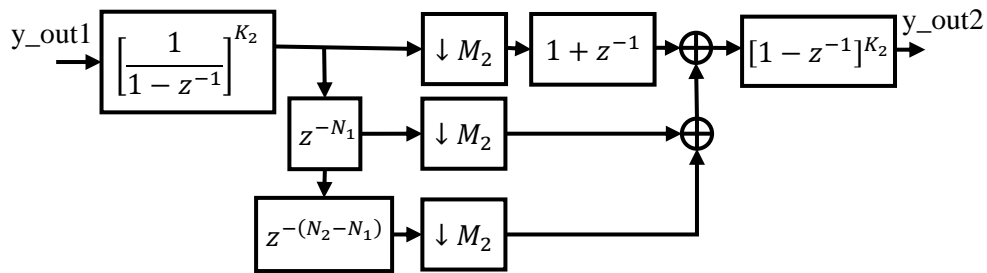


Figura 5.5 Estructura de la segunda etapa del decimador propuesto.

En la figura 5.8 se aprecia que solo los integradores trabajan a mayor frecuencia, la cual es M_1 veces menor que la máxima frecuencia de la entrada del decimador propuesto. La descripción en VHDL de la sección de integradores se muestra en el Anexo 1 d). En este caso se ha asignado $K_2 = 5$. Para evitar un incremento significativo en la latencia, cada operación se ha sincronizado con el contador de la primera etapa. Además, es importante resaltar que cada elemento debe tener la longitud de palabra apropiada para evitar cálculos erróneos.

Posterior a los integradores se encuentra la sección que combina los filtros coseno con el submuestreador de la segunda etapa. En la Figura 5.6 se representa esta sección. Como ya se vio en la etapa 1, en la práctica la combinación de retrasos con el submuestreador se implementa mediante un demultiplexor registrado. En el Anexo 1 e) se muestra el listado en VHDL que describe al demultiplexor, se ha tomado como ejemplo $M_2 = 8$.

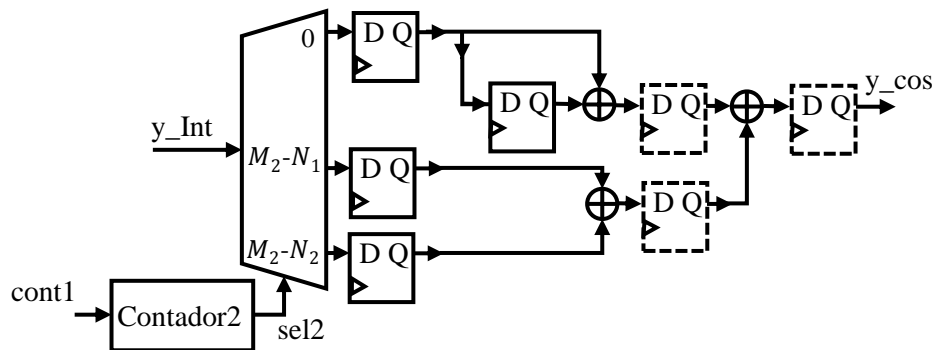


Figura 5.6 Arquitectura de la descomposición polifásica de los filtros coseno.

El código que describe el proceso de sumadores y retrasos de las componentes polifásicas de los filtros coseno se puede ver en el Anexo 1 f).

En la parte final de la segunda etapa del decimador propuesto se halla la sección de diferenciadores. Puesto que su descripción en VHDL es muy similar a la sección de integradores, se ha omitido el código respectivo.

5.1.2 Arquitectura para M es un número potencia de dos

La segunda arquitectura que se ha propuesto en esta tesis se puede utilizar únicamente cuando el factor de decimación es un número potencia de dos. Esta arquitectura se compone de $\log_2(M)$ etapas de decimación por dos. Se ha determinado utilizar la realización no recursiva y descomposición polifásica en cada etapa para disminuir la potencia consumida. La arquitectura generalizada de cada etapa se puede ver en la Figura 5.7 (a), donde p representa la p -ésima etapa de decimación por dos. Anteriormente se ha visto que la combinación de retrasos con submuestreadores se implementa mediante un demultiplexor, como se ve en la Figura 5.7 (b).

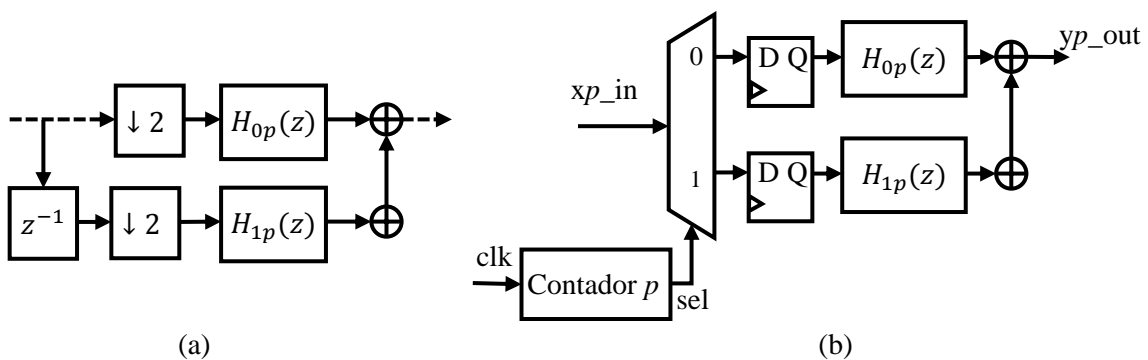


Figura 5.7 Arquitectura de la descomposición polifásica de los filtros coseno.

En cuanto a los filtros polifásicos, $H_{0p}(z)$ y $H_{1p}(z)$, es adecuado realizarlos en su estructura directa transpuesta para aprovechar las virtudes ya mencionadas en la Sección 5.1.2.1. La implementación de cada etapa es muy similar a la primera etapa de la primera arquitectura propuesta. Se tiene que considerar el incremento de longitud de palabra y la correcta sincronización de las operaciones.

5.1.3 Descripción del filtro compensador

El filtro compensador se encuentra en la parte final del decimador propuesto. Su arquitectura es muy simple y se presenta en la figura 5.13. En este filtro también se ha preferido la estructura directa transpuesta por las características que ofrece. Finalmente, en Anexo 1 g) se muestra el listado en VHDL que describe al compensador. En este caso el valor del parámetro de diseño es $B = 1.75$. Solo resta hacer una aclaración. Ya que se ha aplicado la técnica de pipeline a cada sumador, se ha condicionado el momento de ejecución de cada operación para evitar que la latencia aumente considerablemente.

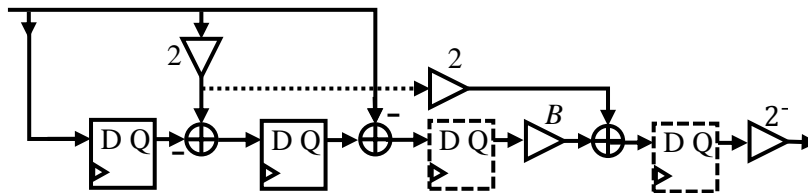


Figura 5.8 Arquitectura del filtro compensador.

5.2 Simulación y resultados de implementación

En este apartado se verifica la correcta operación de los decimadores propuestos y descritos en VHDL. Para tal fin, se efectúan simulaciones estimulando con diferentes señales a la entrada. Asimismo se determinan los recursos utilizados y se estima la potencia consumida, tomando como referencia el FPGA Artix 7 XC7A100T de Xilinx.

Ejemplo 5.2. El primer decimador que se implementa y que se pone bajo prueba es el que se comparó en la sección 4.3.1.2, cuyos parámetros son: $M_1 = 4$, $M_2 = 8$, $N_1 = 3$, $N_2 = 5$, $K_1 = 4$, $K_2 = 5$, $B = 1.75$. Cabe recordar que su PCA es de 70.66 dB y que tiene una DBP menor a 0.55 dB. Su arquitectura se muestra en la Figura 5.8, donde los subfiltros polifásicos tienen las siguientes funciones de transferencia:

$$H_{01}(z) = 1 + 31z^{-1} + 31z^{-2} + z^{-3}, \quad (5.5)$$

$$H_{11}(z) = 4 + 40z^{-1} + 20z^{-2}, \quad (5.6)$$

$$H_{21}(z) = 10 + 44z^{-1} + 10z^{-2}, \quad (5.7)$$

$$H_{31}(z) = 20 + 40z^{-1} + 4z^{-2}. \quad (5.8)$$

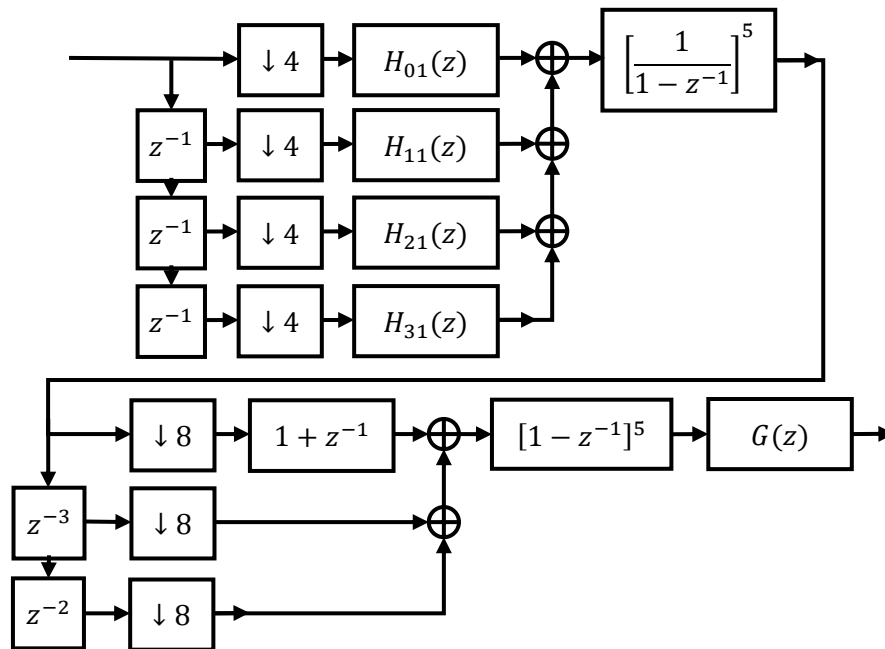


Figura 5.9 Arquitectura del decimador implementado en el Ejemplo 5.2.

Se llevó a cabo la descripción en VHDL de este decimador, y se utilizó el simulador que incluye Active para observar su comportamiento ante diferentes estímulos. La señal de reloj empleada es de 100 MHz. En la figura 5.16 se muestran las respuestas al impulso obtenidas con MATLAB y con el simulador de Active. Debido a la utilización de longitud de palabra de precisión completa en la descripción en VHDL, las respuestas son idénticas. Sin embargo, en los resultados de Active se puede ver que existe una pequeña latencia de 19 ciclos de reloj. Esto se esperaba, ya que todos los sumadores se registraron para lograr mayor frecuencia de operación. Similarmente, en la figura 5.17 se aprecia una comparación entre las respuestas al escalón empleando MATLAB y el simulador de Active. De igual manera, se puede apreciar que las respuestas son idénticas.

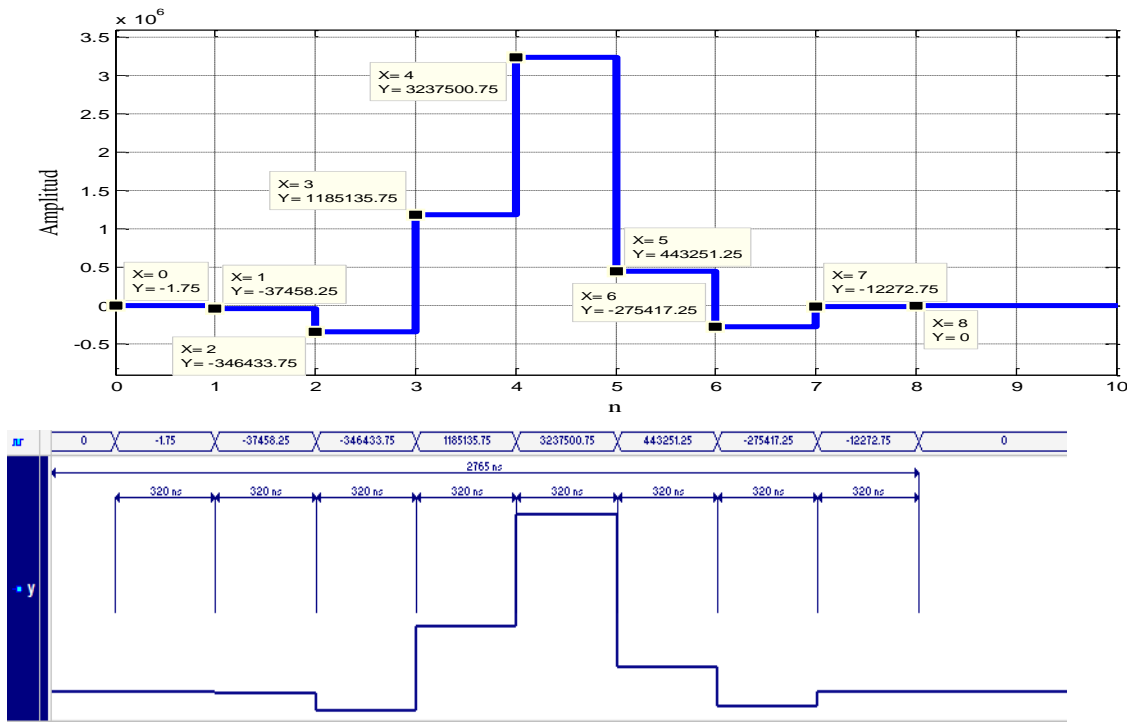


Figura 5.10 Comparación de respuestas al impulso obtenidas con Matlab y con Active.

La utilización de recursos del FPGA se resume en la Tabla 5.1. Se observa que el decimador ocupa pocos recursos. Al comparar los recursos utilizados con otras implementaciones previamente reportadas, resulta que el decimador propuesto en esta tesis requiere aproximadamente 30% menos recursos del FPGA con respecto al decimador reportado en [20], no obstante, las respuestas de magnitud de los filtros son muy similares. Además, gracias al pipeline realizado a cada sumador, se alcanza una muy buena frecuencia de operación de hasta 460 MHz. Por otra parte, la potencia que se reporta es la dinámica. Ésta es de especial interés ya que se encuentra asociada con la actividad del diseño y los eventos de conmutación en el núcleo del dispositivo. Para esta implementación la potencia dinámica es de 4.95 mW.

Ejemplo 5.3. De igual manera, se ha implementado el decimador del Ejemplo 5.2, pero empleando la segunda arquitectura propuesta. Para esta implementación, los parámetros son: $p_1 = 2$, $M_1 = 4$, $M_2 = 8$, $N_1 = 3$, $N_2 = 5$, $K_1 = 4$, $K_2 = 5$, $B = 1.75$. El diagrama a bloques consta de cinco etapas de decimación por dos y se muestra en la Figura 5.12.

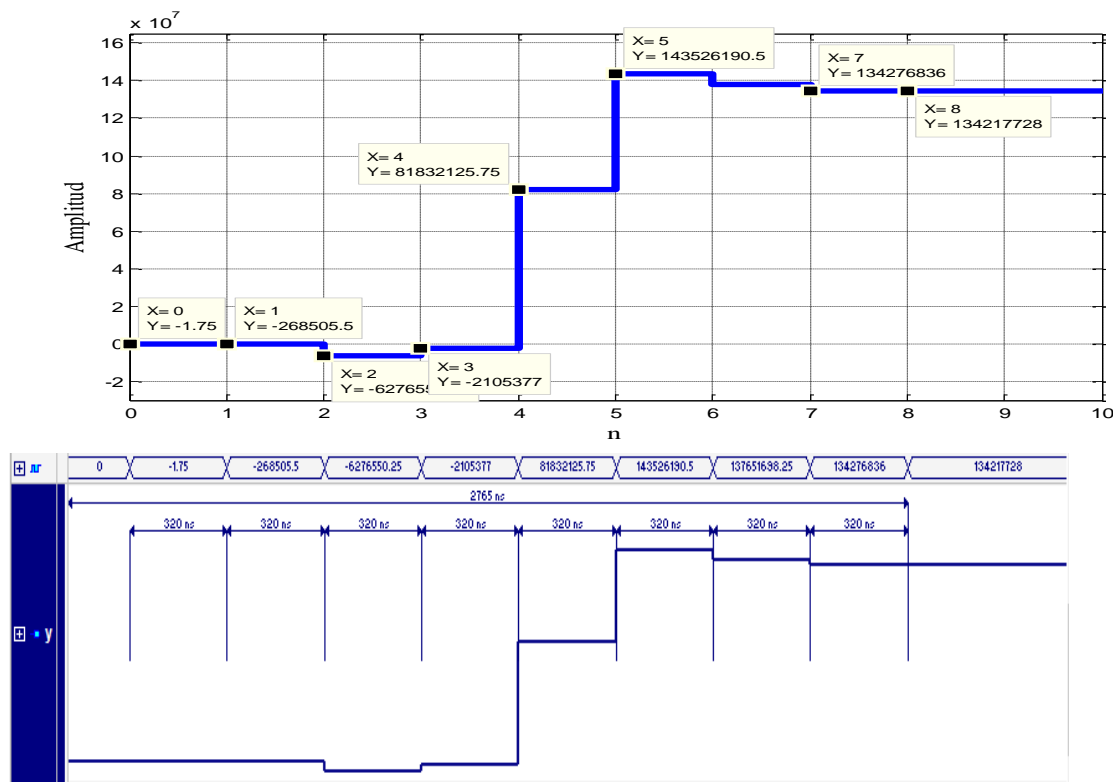


Figura 5.11 Comparación de respuestas al escalón obtenidas con Matlab y con Active.

Utilización lógica	Usado	Disponible	Utilización
Número de registros	835	126800	0%
Número de LUTs	569	63400	0%
Número de LUT-FF completos	543	861	63%
Número de IOBs	35	210	16%
Frecuencia máxima	460.7 MHz		
Potencia dinámica estimada	4.95 mW		

Tabla 5.1 Resumen de utilización de recursos del decimador implementado en el Ejemplo 5.2. El dispositivo destino es el FPGA Artix 7 XC7A100T de Xilinx.

Las componentes polifásicas de cada etapa son:

$$H_{01}(z) = 1 + 6z^{-1} + z^{-2} = H_{02}(z), \quad (5.9)$$

$$H_{11}(z) = 4(1 + z^{-1}) = H_{12}(z), \quad (5.10)$$

$$H_{03}(z) = 1 + 10z^{-1} + 10z^{-2} + 15z^{-3} + 12z^{-4} + 11z^{-5} + 5z^{-6}, \quad (5.11)$$

$$H_{13}(z) = 5 + 11z^{-1} + 12z^{-2} + 15z^{-3} + 10z^{-4} + 10z^{-5} + z^{-6}, \quad (5.12)$$

$$H_{04}(z) = 1 + 10z^{-1} + 5z^{-2} = H_{05}(z), \quad (5.13)$$

$$H_{14}(z) = 5 + 10z^{-1} + z^{-2} = H_{15}(z). \quad (5.14)$$

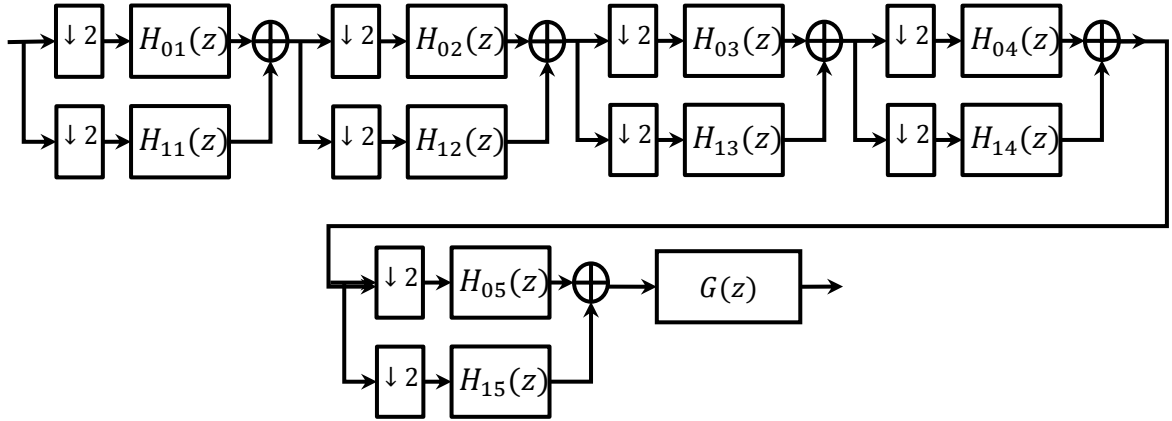


Figura 5.12 Arquitectura del decimador implementado en el Ejemplo 5.3.

Las respuestas al impulso y al escalón obtenidas son idénticas a aquellas del decimador del Ejemplo 5.2. Finalmente en la Tabla 5. 2 se presentan los recursos utilizados por el FPGA y los valores de potencia dinámica consumida y máxima frecuencia de operación. Se advierte que, a pesar de que el decimador del Ejemplo 5.3 utiliza mayor cantidad de recursos con respecto a aquel del Ejemplo 5.2, su potencia estimada es aproximadamente 12% menor. La máxima frecuencia de operación es la misma en ambos decimadores.

Utilización lógica	Usado	Disponible	Utilización
Número de registros	993	126800	0%
Número de LUTs	778	63400	1%
Número de LUT-FF completos	757	1016	74%
Número de IOBs	35	210	16%
Frecuencia máxima	460.7 MHz		
Potencia dinámica estimada	4.43 mW		

Tabla 5.2 Resumen de utilización de recursos del decimador implementado en el Ejemplo 5.3. El dispositivo destino es el FPGA Artix 7 XC7A100T de Xilinx.

Conclusiones

En esta tesis se logró mejorar los decimadores basados en el filtro comb. Mediante una estructura de dos etapas y la conexión en cascada de filtros simples que trabajan a bajas frecuencias se consiguió una mejoría a la respuesta de magnitud. Adicionalmente, se presentaron dos arquitecturas eficientes.

Por un lado se incrementó el rechazo antialiasing del filtro comb tradicional. Esto se hizo utilizando una estructura de dos etapas e insertando dos filtros coseno en la segunda etapa. Con ello, la atenuación en la mayoría de las bandas de rechazo se ha incrementado asimismo el PCA fue aumentado en un rango de 15 a 35 dB.

Además, con un filtro compensador, que opera a la más baja frecuencia, se disminuyó la caída en la banda de paso. La DBP se redujo considerablemente, pues los valores típicos que se alcanzan son menores a 0.6 dB y hasta 0.25 dB en el mejor de los casos.

Por otra parte, gracias a la estructura decimadora de dos etapas, se propuso una arquitectura que equilibra el área requerida y el consumo de potencia. Para ello, se emplea la técnica de descomposición polifásica en la primera etapa. De esta manera se evita tener filtros trabajando a la máxima frecuencia. Por su parte, en la segunda etapa se hizo una combinación de la estructura CIC con una estructura polifásica, logrando que únicamente los filtros integradores trabajen a una frecuencia M_1 veces menor, y que los filtros coseno trabajen a la frecuencia más baja. Además se presentó una arquitectura decimadora de bajo consumo de potencia, la cual es conveniente cuando el factor de decimación es una potencia de dos.

Con la metodología de diseño presentada, se pueden cumplir las especificaciones típicas de un decimador. Resulta sencillo satisfacer atenuaciones mínimas entre 40 y 90 dB. Simultáneamente, es fácil tener desviaciones en banda de paso no mayores a 0.6 dB. Todo ello con poca área utilizada y mínima potencia consumida. De este modo, el decimador propuesto es adecuado para utilizarse en aplicaciones de SD-ADC.

Finalmente, los decimadores propuestos se describieron en VHDL y se verificó su correcto funcionamiento a través de simulaciones. Posteriormente se implementaron en

un FPGA para evaluar su desempeño en términos de área utilizada, potencia estimada y máxima frecuencia de operación. Los resultados obtenidos demuestran que la adecuada operación del decimador y los beneficios que ofrece en términos de área, potencia y máxima frecuencia de operación.

Trabajo a futuro

Los siguientes puntos se pueden considerar como trabajo a futuro:

- Realizar un análisis de los efectos perjudiciales introducidos por el truncamiento de la longitud de palabra. Como se ha visto, la longitud de palabra se relaciona directamente con el área utilizada. En muchas aplicaciones, después de efectuada la decimación solo se conservan algunos bits más significativos y los restantes se desechan. De manera que se podría lograr un ahorro de área y potencia consumida al truncar progresivamente el ancho de bus de cada etapa. No obstante, ello podría ocasionar severos errores.

Anexos

1. Códigos en VHDL

a) Descripción en VHDL de la arquitectura del demultiplexor registrado

```
DEMUX:process(rst, clk, contM1)
begin
  if rst = '0' then
    RE1<=(others =>(others =>'0'));
    contM1<=(others =>'0');
  elsif rising_edge(clk) then
    if contM1=to_unsigned(M1-1, contM1'length) then
      contM1<=(others =>'0');
    else
      contM1<=contM1 + 1;
    end if;
    RE1(to_integer(contM1))<=sx_in;
  end if;
end process DEMUX;
```

b) Descripción en VHDL de la arquitectura de los bloques generadores de multiplicaciones para los filtros polifásicos del ejemplo 5.1

```
--Bloque Generador de Multiplicadores Polifásica 0:( 1, 12, 3 )
RE1_0_4<=RE1(0)&"00";
RE1_0_3<=RE1_0_4 - RE1(0);
RE1_0_12<=RE1_0_3&"00";

--Bloque Generador de Multiplicadores Polifásica 1:( 10, 6 )
RE1_1_4<=RE1(1)&"00";
RE1_1_3<=RE1_1_4 - RE1(1);
RE1_1_5<=resize(RE1_1_4, RE1_1_5'length) + RE1(1);
RE1_1_6<=RE1_1_3&'0';
RE1_1_10<=RE1_1_5&'0';

--Bloque Generador de Multiplicadores Polifásica 2:( 6, 10 )
RE1_2_4<=RE1(2)&"00";
RE1_2_3<=RE1_2_4 - RE1(2);
RE1_2_5<=resize(RE1_2_4, RE1_2_5'length) + RE1(2);
RE1_2_6<=RE1_2_3&'0';
RE1_2_10<=RE1_2_5&'0';

--Bloque Generador de Multiplicadores Polifásica 3:( 3, 12, 1 )
RE1_3_4<=RE1(3)&"00";
RE1_3_3<=RE1_3_4 - RE1(3);
RE1_3_12<=RE1_3_3&"00";
```

c) Descripción e VHDL de sumadores y retrasos del filtro del ejemplo 5.1

```
FiltrosE1:process(rst, clk, contM1)
begin
  if rst = '0' then
    RE1_02<=(others =>'0'); SE1_01<=(others =>'0');
    SE1_P0<=(others =>'0');
    RE1_11<=(others =>'0'); SE1_P1<=(others =>'0');
    RE1_21<=(others =>'0'); SE1_P2<=(others =>'0');
    RE1_32<=(others =>'0'); SE1_31<=(others =>'0');
    SE1_P3<=(others =>'0');
    SE1_POP3<=(others =>'0'); SE1_P1P2<=(others =>'0');
    SE1_F<=(others =>'0');
  elsif rising_edge(clk) then
    if contM1=to_unsigned(0, contM1'length) then
      RE1_02<=RE1_0_3;
      SE1_01<=RE1_02+RE1_0_12;
      SE1_P3<=SE1_31+RE1_3_3;
      SE1_P1P2<=resize(SE1_P1, SE1_P1P2'length)+SE1_P2;
    end if;
    if contM1=to_unsigned(1, contM1'length) then
      RE1_11<=RE1_1_6;
      SE1_P0<=SE1_01+RE1(0);
    end if;
    if contM1=to_unsigned(2, contM1'length) then
      RE1_21<=RE1_2_10;
      SE1_P1<=RE1_11+RE1_1_10;
      SE1_POP3<=resize(SE1_P0, SE1_POP3'length)+SE1_P3;
    end if;
    if contM1=to_unsigned(3, contM1'length) then
      RE1_32<=RE1(3);
      SE1_31<=RE1_32+RE1_3_12;
      SE1_P2<=RE1_21+RE1_2_6;
      SE1_F<=SE1_POP3+SE1_P1P2;
    end if;
  end if;
end process FiltrosE1;
```

d) Descripción e VHDL de la arquitectura de la sección de integradores

```
Integradores:process (rst, clk, contM1)
begin
  if rst = '0' then
    SI1<=(others =>'0'); SI2<=(others =>'0');
    SI3<=(others =>'0'); SI4<=(others =>'0');
    SI5<=(others =>'0');
  elsif rising_edge (clk) then
    if contM1=to_unsigned(1, contM1'length) then
      SI2<=SI1+SI2;
    end if;
    if contM1=to_unsigned(2, contM1'length) then
      SI3<=SI2+SI3;
    end if;
    if contM1=to_unsigned(3, contM1'length) then
      SI4<=SI3+SI4;
    end if;
    if contM1=to_unsigned(0, contM1'length) then
      SI1<=SE1_F+SI1;
      SI5<=SI4+SI5;
    end if;
  end if;
end process Integradores;
```

e) Descripción en VHDL de la arquitectura del demultiplexor correspondiente a la segunda etapa de decimación

```
DEMUX2:process(rst, clk, contM1)
begin
  if rst = '0' then
    RE2_0<=(others =>'0'); RE2_N2<=(others =>'0');
    RE2_N1<=(others =>'0'); contM2<=(others =>'0');
  elsif rising_edge(clk) then
    if contM1=to_unsigned(M1-1, contM1'length) then
      if contM2=to_unsigned(M2+1, contM2'length) then
        contM2<=to_unsigned(2, contM2'length);
      else
        contM2<=contM2 + 1;
      end if;
    end if;
    if contM1=to_unsigned(1, contM1'length) then
      if contM2=to_unsigned(2, contM2'length) then
        RE2_0<=SI5;
      else
        RE2_0<=RE2_0;
      end if;
    end if;
    if contM1=to_unsigned(1, contM1'length) then
      if contM2=to_unsigned(2+3, contM2'length) then
        RE2_N2<=SI5;
      else
        RE2_N2<=RE2_N2;
      end if;
    end if;
    if contM1=to_unsigned(1, contM1'length) then
      if contM2=to_unsigned(2+5, contM2'length) then
        RE2_N1<=SI5;
      else
        RE2_N1<=RE2_N1;
      end if;
    end if;
  end if;
end process DEMUX2;
```

f) Descripción en VHDL de la arquitectura de los sumadores y registros de los filtros coseno

```
FiltrosPolyE2:process(rst, clk, contM1, contM2)
begin
  if rst ='0' then
    RE2_01<=(others =>'0'); SE2_P0<=(others =>'0');
    SE2_PN1PN2<=(others =>'0'); SE2_F<=(others =>'0');
  elsif rising_edge(clk) then
    if contM1=to_unsigned(1, contM1'length) then
      if contM2=to_unsigned(2, contM2'length) then
        RE2_01<=RE2_0;
      end if;
    end if;
    if contM1=to_unsigned(2, contM1'length) then
      if contM2=to_unsigned(2+5, contM2'length) then
        SE2_PN1PN2<=RE2_N1+RE2_N2;
      end if;
    end if;
    if contM1=to_unsigned(2, contM1'length) then
      if contM2=to_unsigned(2, contM2'length) then
        SE2_P0<=RE2_0+RE2_01;
      end if;
    end if;
    if contM1=to_unsigned(3, contM1'length) then
      if contM2=to_unsigned(2+0, contM2'length) then
        SE2_F<=SE2_P0+SE2_PN1PN2;
      end if;
    end if;
  end if;
end process FiltrosPolyE2;
```

g) Descripción en VHDL de la arquitectura del filtro compensador

```
Compensador:process(rst, clk, contM1, contM2, SD5)
begin
  SD5_2<=SD5&'0';
  if rst ='0' then
    RC_2<=(others =>'0'); SC_1<=(others =>'0');
    RC_1<=(others =>'0'); SC_2<=(others =>'0');
    B<=(others =>'0'); SC_F<=(others =>'0');
  elsif rising_edge(clk) then
    if contM1=to_unsigned(0, contM1'length) then
      if contM2=to_unsigned(2+2, contM2'length) then
        RC_2<=SD5;
        SC_1<=SD5_2-RC_2;
        RC_1<=SD5_2;
      end if;
    end if;
    if contM1=to_unsigned(1, contM1'length) then
      if contM2=to_unsigned(2+2, contM2'length) then
        SC_2<=SC_1-SD5;
      else
        SC_2<=SC_2;
      end if;
    end if;
    if contM1=to_unsigned(2, contM1'length) then
      if contM2=to_unsigned(2+2, contM2'length) then
        B<=resize((SC_2&"000")-SC_2, B'length);
      end if;
    end if;
    if contM1=to_unsigned(3, contM1'length) then
      if contM2=to_unsigned(2+2, contM2'length) then
        SC_F<=B+(RC_1&"000");
      end if;
    end if;
  end if;
end process Compensador;
```


2. Artículos Adjuntos

a) “Using Cosine Filters to Improve Alias Rejection in Comb Decimation Filter”

Using Cosine Filters to Improve Alias Rejection in Comb Decimation Filter

Gordana Jovanovic Dolecek and Angel Garcia Robles
Department of Electronics,
Institute INAOE, Puebla, Mexico
gordana@ieee.org, angel_garcia_rob@hotmail.com

Abstract—This paper presents efficient use of cosine filters to improve the alias rejection in the comb decimation filters for Sigma-Delta-Analog/Digital (SD A/D) converters application. Like comb filters, the cosine filters are also multiplierless, thus resulting in the overall multiplierless structure. The goal was to decrease the decimator complexity, compared with the original cosine based decimator, and to increase the number of design parameters. To this end, we propose a two-stage structure, where the cosine filters are in the second stage. The novel structure is proposed, in which the polyphase decomposition is applied to the cascade of cosine filters and the comb filter from a CIC (Cascaded-Integrator-Comb) structure. The method is compared with the equivalent comb and the original cosine-based method.

Keywords—comb filter; cosine filter; aliasing; Sigma-Delta A/D converter.

I. INTRODUCTION

The oversampled Sigma-Delta Analog/Digital (SD A/D) converters consist of analog part (modulator) and a digital part (decimation stage). In modulator the analog signal is sampled with a frequency much larger than the Nyquist frequency, which is the minimum sampling frequency required for preserving the information in analog signal. The rate of the oversampled signal must be decreased to the Nyquist frequency, to be efficiently processed by the DSP (Digital Signal Processing). This process is performed in a digital form in a decimation stage. The decreasing of the sampling rate may introduce the aliasing effects that deteriorate the signal and have to be eliminated by the decimation, or antialiasing filter. The decimation is usually performed in more stages. The decimation filter at the first stage works at the high input rate and consequently must be simple in order to avoid high power consumption. The most simple decimation filter, usually used in the first decimation stage, is the comb filter proposed by Hogenauer [1]. Nevertheless, its magnitude response presents a high passband droop and low attenuations in the folding bands (bands around the zeros of the comb filter), which may deteriorate the decimated signal. In this paper we consider only the improvement of the comb aliasing rejection based on cosine filters. The methods in [2-4] are for example devoted to improvement of comb alias rejection. The cosine-based method [4] introduced two cascaded cosine filters to improve the aliasing rejection. However, the cosine filters work at high input rate. The novelty of this paper is the proposal of two-stage comb structure, where the cosine filters are moved to the second stage. The rest of the paper is organized in the following way. Next section presents comb and cosine filters. Section III describes the proposed two-stage structure. Section

IV presents the choice of the design parameters and comparisons with the original cosine-based decimator.

II. COMB AND COSINE FILTERS

A. Comb filter

The system function of the of the K -th order comb filter can be given in a recursive or nonrecursive form, as shown in Eq.(1):

$$H(z) = \left[\frac{1}{M} \frac{1-z^{-M}}{1-z^{-1}} \right]^K = \left[\frac{1}{M} \sum_{k=0}^{M-1} z^{-k} \right]^K, \quad (1)$$

where M is the decimation factor.

The corresponding magnitude characteristic is given as:

$$\left| H(e^{j\omega}) \right| = \left| \frac{1}{M} \frac{\sin(\omega M / 2)}{\sin(\omega / 2)} \right|^K. \quad (2)$$

The comb filter is usually used in the first stage of decimation. However, it does not provide enough attenuations in the folding bands. The most critical is the first folding band which has less attenuations than other folding bands. The attenuations in the folding bands can be increased by increasing the value of the parameter K . However, this in turns, increases the droop in the passband.

B. Cosine filters

In [4] are introduced cosine filters:

$$H_c(z) = \frac{1+z^{-N_1}}{2} \frac{1+z^{-N_2}}{2}, \quad (3)$$

where the parameters N_1 and N_2 are integers chosen in the following way [4]:

$$M \text{ even: } N_1 = M / 2 - 1; N_2 = M / 2 + 1; \quad (4)$$

$$M \text{ odd: } N_1 = \lfloor M / 2 \rfloor; N_2 = N_1 + 1. \quad (5)$$

The magnitude response of the cosine filter is given as:

$$\left| H_c(e^{j\omega}) \right| = \frac{1}{4} \left| \cos(\omega N_1 / 2) \cos(\omega N_2 / 2) \right|. \quad (6)$$

This work is supported by the CONACYT grant No. 179587

This idea is illustrated in Fig.1, where the magnitude responses of the cosine-based filter [4] and comb, are shown, considering $M=12$, $K=3$, and $N_1=5$, $N_2=7$.

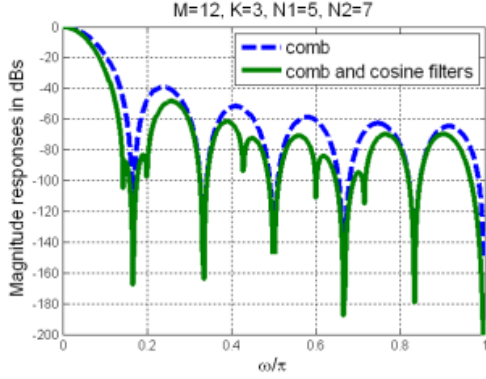


Fig. 1. Comb and cosine filters [4].

Note that the folding bands of the filter [4] are wider, especially, the first folding band, which is the most critical folding band. However, in method [4], the cosine filters work at high input rate, which in turns increases the power consumption. The goal here is to propose efficient structure, where the cosine filters work at lower rate.

III. PROPOSED TWO-STAGE STRUCTURE

Considering that the decimation factor M can be presented as a product of integers M_1 and M_2 , we propose here a two-stage comb structure where the decimation factor at first and second stages are M_1 , and M_2 , respectively. The comb filters at first and second stages are:

$$H_1(z) = \left[\frac{1 - z^{-M_1}}{M_1 - z^{-1}} \right]^{K_1}; H_2(z) = \left[\frac{1 - z^{-M_2}}{M_2 - z^{-1}} \right]^{K_2}; \quad (7)$$

The cosine filters at high input rate are given as:

$$H_{2c}(z^{M_1}) = \frac{1 + z^{-M_1 N_{21}}}{2} \frac{1 + z^{-M_1 N_{22}}}{2}, \quad (8)$$

where:

$$M_2 \text{ even: } N_{21} = M_2 / 2 - 1; N_{22} = M_2 / 2 + 1; \quad (9)$$

$$M_2 \text{ odd: } N_{21} = \lfloor M_2 / 2 \rfloor; N_{22} = N_{21} + 1; \quad (10)$$

The cosine filters (8) can be moved to the second stage becoming:

$$H_{2c}(z) = \frac{1 + z^{-N_{21}}}{2} \frac{1 + z^{-N_{22}}}{2}. \quad (11)$$

The corresponding structure is shown in Fig.2.

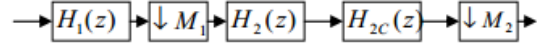
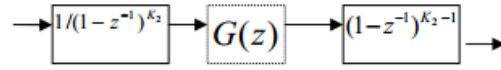
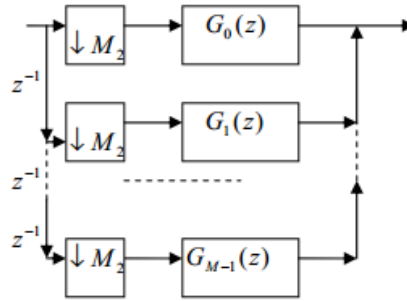


Fig. 2. Proposed structure.

The first stage can be implemented in a nonrecursive form, while comb in the second stage can be implemented as a CIC filter. The polyphase decomposition is applied to the cascade of cosine filters and comb from CIC structure, as illustrated in Fig.3, where $G_k(z)$, $k=0, \dots, M_2-1$, are the polyphase components.



a. Second stage.



b. Block $G(z)$.

Fig. 3. Second stage in structure from Fig.2.

The method is illustrated in the following examples..

Example 1: Consider the decimation factor $M=12$, and $M_1=3$, and $M_2=4$, $K_1=K_2=3$, From (9) we have: $N_{21}=1$, $N_{22}=3$. Fig.4 compares the magnitude responses of the proposed filter and the corresponding comb.

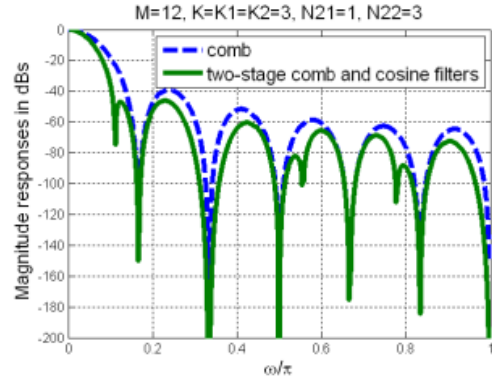
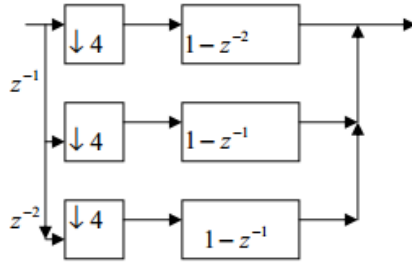
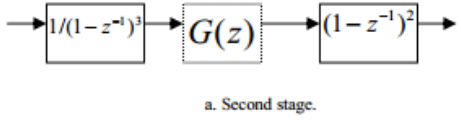


Fig.4. Magnitude responses of the proposed filter and corresponding comb.

Next, we consider the polyphase decomposition in the second stage. To this end, the comb filter, from CIC structure, is cascaded with the cosine filter $H_{2c}(z)$, resulting in:

$$\begin{aligned}
 G(z^{M_2}) &= H_{2c}(z)(1-z^{-M_2}) \\
 &= \frac{1}{4}(1-z^{-1})(1-z^{-3})(1-z^{-4}) \\
 &= \frac{1}{4}(1+z^{-1}+z^{-3}-z^{-5}-z^{-7}-z^{-8}) \\
 &= \frac{1}{4}\{(1-z^{-8})+z^{-1}(1-z^{-4})-z^{-3}(1-z^{-4})\}
 \end{aligned} \tag{12}$$

The structure of the second stage is shown in Fig. 5a, while the block $G(z)$ is given in Fig. 5b.



Example 2: In this example we have $K=K_1=K_2=3$, $M=15$ and $M_1=3$, $M_2=5$. From (10) we get $N_{21}=2$, $N_{22}=3$. The magnitude responses of the proposed filter and corresponding comb, are compared in Fig.6. The filter $(1-z^{-5})$ is cascaded with the cosine filter $H_{2c}(z)$ resulting in the following polyphase decomposition:

$$\begin{aligned}
 G(z^{M_2}) &= H_{2c}(z)(1-z^{-M_2}) \\
 &= \frac{1}{4}(1-z^{-2})(1-z^{-3})(1-z^{-5}) \\
 &= \frac{1}{4}\{(1-z^{-10})+z^{-2}(-1+z^{-5})+z^{-3}(-1+z^{-5})\}
 \end{aligned} \tag{13}$$

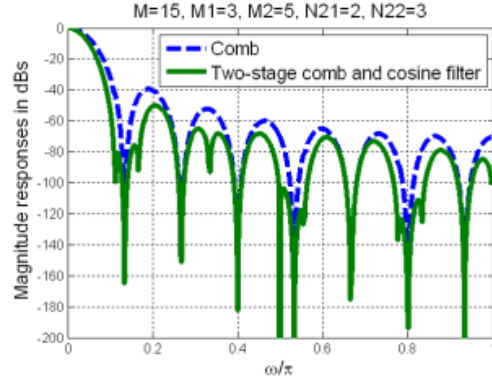


Fig. 6. Magnitude responses of the proposed filter and corresponding comb.

IV. CHOICE OF DESIGN PARAMETERS AND COMPARISONS

A. Choice of Design Parameters

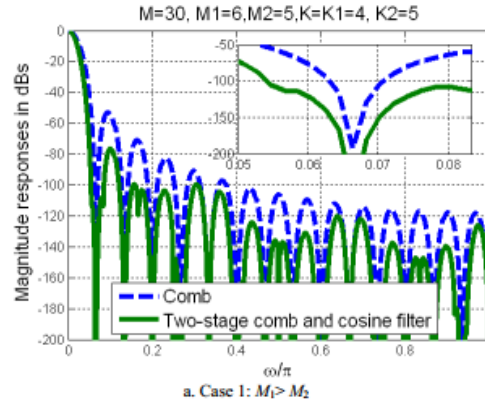
The design parameters are the decimation factors M_1 and M_2 and the numbers of the cascaded combs at the first and second stages, i.e. K_1 and K_2 .

The decimation factors are chosen in a such way that,

$$M_1 \approx M_2. \tag{14}$$

Taking, $M_1 < M_2$, while satisfying (13), slightly better attenuations are obtained, than in the case, in which $M_1 > M_2$.

Additionally, knowing that the second stage works at low rate, which is M_1 times less than the high input rate, the number of the cascaded combs at the second stage, can be slightly increased, in order to improve aliasing rejection, without significantly increase of the complexity. Next example illustrates this idea.



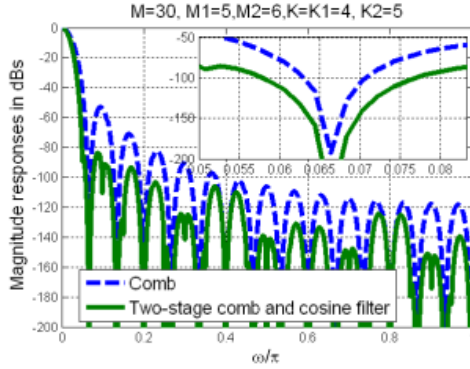


Fig.7. Illustration of Example 3.

B. Comparisons

We compare the proposed method with the method [4] using $M=9$, $K=4$ and $N_1=4$, $N_2=5$.

In the proposed method $K_1=4$, $K_2=5$, $M_1=M_2=3$, and $N_{21}=1$, $N_{22}=2$.

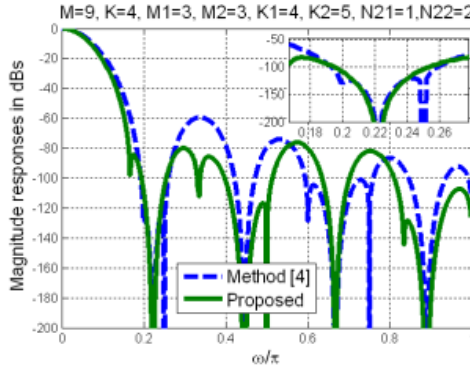


Fig.7. Comparison with Method [4].

The filter from Method [4] presents better attenuations in the third folding band, while the proposed method provides better attenuation in the second and fourth folding bands. However, the attenuations in the first folding band are similar.

In the proposed structure, the polyphase components at the first stage are:

$$H_{10}(z) = 1 + 16z^{-1} + 4z^{-2} \dots$$

$$H_{11}(z) = 4 + 19z^{-1} + z^{-2}. \quad (15)$$

$$H_{12}(z) = 10 + 10z^{-1}.$$

Similarly the polyphase components at the second stage are:

$$H_{2CC0}(z) = 1 + 2z^{-1} \dots H_{2CC1}(z) = -1 + z^{-1}.$$

$$H_{2CC2}(z) = -2 - z^{-1}. \quad (16)$$

However, the polyphase components of the filter from the Method [4] require significantly more adders than the components (13) and (14).

V. CONCLUDING REMARKS

This paper presents the modification of the cosine-based decimation filter in order to fulfill two goals: First, to move the cosine filters to a lower rate, and thus decrease the overall complexity; Second, to increase the number of the designed parameters. The proposed structure has two stages, considering that the overall decimation factor can be presented as a product of two integers. The comb at the first stage is implemented in a nonrecursive form, while the comb at the second stage is implemented as a CIC (Cascaded-Integrators-Combs) structure. The cosine filters are cascaded with the CIC structure, and thus the efficient multiplierless polyphase decomposition is obtained. Moreover, the proposed filter presents better attenuations in some folding bands, than the original cosine-based filter.

REFERENCES

- [1] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. On Acoust., Speech, Signal Processing*, vol.29, pp. 155-162, 1981.
- [2] L. Lo Presti, "Efficient modified-sin filter for sigma-delta A/D converters," *IEEE Trans. Circuits Syst. II: Analog Digit, Signal Process*, vol. 47, pp. 1204-1213, 2000.
- [3] M. Laddomada, "Generalized comb decimation filters for $\Sigma\Delta$ A/D converters," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 54, pp. 994-1005, 2007.
- [4] G. Jovanovic Dolcecek and M. Laddomada, "An improved class of multiplierless decimation filters: Analysis and design," *Elsevier Digital Signal Processing*, vol. 23, No. 5, pp. 1773-1782, 2013.

b) “Polyphase Decomposition of Non-Recursive Comb-Cosine Decimation Filter for SDR Applications”

Polyphase Decomposition of Non-Recursive Comb-Cosine Decimation Filter for SDR Applications

Angel Garcia Robles, Gordana Jovanovic Dolecek

Department of Electronics

Institute INAOE

Puebla, Mexico

agarcia@inaoep.mx, gordana@ieee.org

Abstract— This paper describes a novel, low complexity nonrecursive comb-cosine decimation filter, with an improved alias rejection. It is supposed that the overall decimation factor can be presented as a power of two. The structure is composed of different stages, each decimated by two. The polyphase decomposition is applied at each stage. The choice of the design parameters is described and illustrated with the examples. The method is compared with the classical non-recursive comb structure and the original comb-cosine structure. The FPGA implementation of the proposed filter is applied in order to evaluate its performances in terms of area and power consumption.

Keywords— Decimation; Comb filter; Cosine filters; Non recursive comb; Field programmable gate array (FPGA).

I. INTRODUCTION

A fundamental block in a Software Defined Radio (SDR) receiver is a digital decimator, which is useful to processing various wireless communications standards at low cost [1]. Decimator consists of two blocks: a downsampler which reduces the rate of a signal by an integer, and a lowpass filter which is used to limit the spectrum of the signal and thus to avoid aliasing produced in the process of downsampling. The filter is called decimation, or antialiasing filter, and precedes the downsampler.

The most simple decimation filter, suitable for the first decimation stage, is the comb filter introduced by Hogenauer [2]. Nevertheless, its magnitude response presents a high passband droop and poor attenuations in the folding bands (bands around the zeros of the comb filter), which may deteriorate the decimated signal.

Different methods have been proposed to improve the magnitude response of the comb filter. The methods [3]-[5] improve the passband, while stopband is enhanced in methods [6]-[8]. On the other hand, methods [9]-[10] improve both bands.

The design and implementation of a non-recursive comb-cosine decimation filter is described in this paper. It is supposed that the decimation factor M can be presented as a power of two. In order to reduce power consumption, cosine filters, introduced in [8], are cascaded with comb filters at

different stages of the non-recursive structure. The results are compared in terms of alias rejection improvements.

The rest of the paper is organized in the following way. The next section describes non-recursive structure and cosine filters. The proposed filter is presented in Section III. The FPGA implementation results are explained in Section IV. Finally, conclusions are given in Section V.

II. NON-RECURSIVE STRUCTURE AND COSINE FILTERS

A. Comb Non-Recursive Structure

The non-recursive comb filter for $M = 2^p$ is a p -stage filter where each stage is decimated by 2:

$$H(z) = \prod_{m=0}^{p-1} [0.5(1 + z^{-2^m})]^K. \quad (1)$$

For example taking $M = 8 = 2^3$, the comb nonrecursive structure has $p=3$ stages:

$$H(z) = \prod_{m=0}^2 [0.5(1 + z^{-2^m})]^K = [0.5(1 + z^{-1})]^K [0.5(1 + z^{-2})]^K [0.5(1 + z^{-2^2})]^K. \quad (2)$$

The power consumption of the structure (1) is much less than the power consumption of the corresponding comb recursive structure. The power consumption can be further decreased by applying the polyphase decomposition at each stage.

However, the alias rejection of this structure is low and must be improved.

B. Cosine filters

To improve the alias rejections in the folding bands, in [8] is proposed the cascade of cosine filters,

$$H_c(z) = \frac{1 + z^{-N_1}}{2} \frac{1 + z^{-N_2}}{2}, \quad (3)$$

which works at high input rate.

The corresponding magnitude response is:

$$|H_s(e^{j\omega M})| = \left| \cos\left(\frac{\omega N_1}{2}\right) \right| \left| \cos\left(\frac{\omega N_2}{2}\right) \right|. \quad (4)$$

The parameters N_1 and N_2 are defined as:

$$M \text{ even: } N_1 = \frac{M}{2} - 1; N_2 = \frac{M}{2} + 1. \quad (5)$$

In [8] is also considered a non-recursive structure where M is a power of two, and the filter (3) is applied at the first stage.

The main goal of this work is to decrease the power consumption of the nonrecursive structure [8] by placing the cosine filters (3) at lower stages. This idea is presented in next section.

III. PROPOSED STRUCTURES

A. Equivalent Two-stage Structure

The proposed filter can be presented as an equivalent two stage comb filter with decimation factors M_1 and M_2 , where $M = 2^p = M_1 M_2$, and,

$$M_1 = 2^{p_1}; M_2 = 2^{p-p_1}, \quad p_1 \geq 1, \quad p > 2. \quad (6)$$

$$H(z) = H_1(z)H_2(z^{M_1}), \quad (7)$$

where

- $H_1(z)$ is the comb filter at first stage.
- $H_2(z^{M_1})$ is the comb filter at the second stage, at high rate.

We propose here to introduce the cosine filters (2) at the second stage of the filter (7). Consequently, at high rate we have:

$$H_s(z^{M_1}) = \left[\frac{1 + z^{-M_1 N_1}}{2} \right]^{k_1} \left[\frac{1 + z^{-M_1 N_2}}{2} \right]^{k_2}. \quad (8)$$

From (5), we have for $M_2 > 2$:

$$N_1 = \frac{M_2}{2} - 1; N_2 = \frac{M_2}{2} + 1. \quad (9)$$

Using (3), the system function of the proposed filter at high input rate, is given as:

$$H_p(z) = H_1^{K_1}(z)H_2^{K_2}(z^{M_1})H_s(z^{M_1}), \quad (10)$$

where:

$H_1^{K_1}(z)$ is the equivalent comb filter at first stage cascaded K_1 times.

$H_2^{K_2}(z^{M_1})$ is the equivalent comb filter at the second stage, cascaded K_2 times.

The values of M_1 are: $2, 2^2, 2^3, \dots, 2^{p-2}$.

Knowing that the filter (8) cannot be placed in the first and last stage, there are $p-2$ possible structures.

The structure is called *Structure L*, if the filter $H_s(z)$ is placed at the stage $L+1$. As an example, Table I shows the parameters of the structures for $p=3, 4, 5$, and 6.

TABLE I

p	M_1	M_2	L	Filter $H_s(z)$ at high input rate	N_1, N_2
3	2	4	1	$H_s(z^2)$	1, 3
4	2	8	1	$H_s(z^2)$	3, 5
4	4	4	2	$H_s(z^4)$	1, 3
5	2	16	1	$H_s(z^2)$	7, 9
5	4	8	2	$H_s(z^4)$	3, 5
5	8	4	3	$H_s(z^8)$	1, 3
6	2	32	1	$H_s(z^2)$	15, 17
6	4	16	2	$H_s(z^4)$	7, 9
6	8	8	3	$H_s(z^8)$	3, 5
6	16	4	4	$H_s(z^{16})$	1, 3

Next example illustrates the proposed structures.

Example 1:

Consider $M = 16 = 2^4$, and $K = 4$. The non-recursive structure has $p = 4$ stages. We have:

$$H(z) = \frac{1}{16^4} [1 + z^{-1}]^4 [1 + z^{-2}]^4 [1 + z^{-4}]^4 [1 + z^{-8}]^4. \quad (11)$$

Using Table I, it follows that we have two structures:

Structure 1 ($L=1$):

The filter (8) is placed in the second stage, resulting in: $M_1 = 2, M_2 = 8, N_1 = 3, N_2 = 5$. The magnitude responses of the *Structure 1*, and the corresponding comb filter, are given in Fig.1.

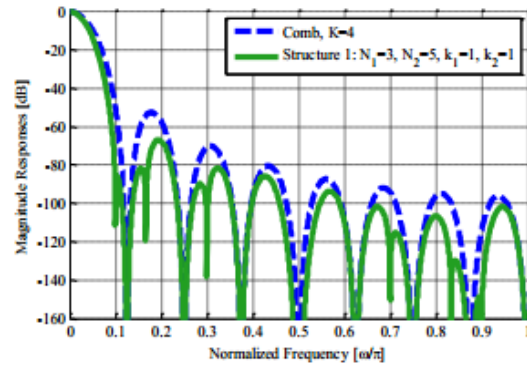


Fig. . Structure 1: $M_1 = 2, M_2 = 8$.

Structure 2: ($L=2$)

The filter (8) is placed at the third stage, resulting in: $M_1 = M_2 = 4, N_1 = 1, N_2 = 3$. The corresponding magnitude responses are shown in Fig. 2. Note that the attenuations in the

folding bands are slightly decreased in the *Structure 2*, in comparison with the *Structure 1*.

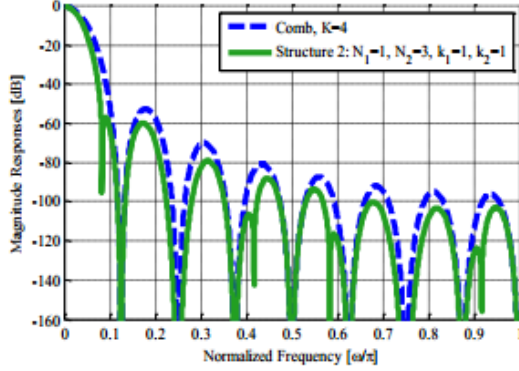


Fig. 2. Structure 2: $M_1 = 4, M_2 = 4$.

B. Polyphase decomposition

The combs in all stages $m = 1, 2, \dots, k$, can be presented by its polyphase components, $H_0^m(z)$ and $H_1^m(z)$:

$$H^m(z) = H_0^m(z) + z^{-1}H_1^m(z). \quad (12)$$

Note that the polyphase components are equal at all stages except at the stage $L+1$.

Example 2:

We consider the polyphase components in Example 1. The structure with the polyphase components is shown in Fig. 3.

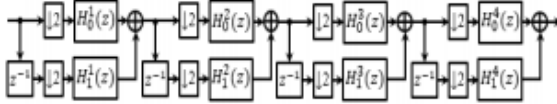


Fig. 3. Polyphase decomposition for Filters in Example 1.

For the *Structure 1*, it holds:

$$H_0^1(z) = H_0^3(z) = H_0^4(z) = 1 + 6z^{-1} + z^{-2}. \quad (13)$$

$$H_1^1(z) = H_1^3(z) = H_1^4(z) = 4(1 + z^{-1}). \quad (14)$$

$$H_0^2(z) = 1 + 6z^{-1} + 5z^{-2} + 8z^{-3} + 5z^{-4} + 6z^{-5} + z^{-6}. \quad (15)$$

$$H_1^2(z) = 4 + 5z^{-1} + 7z^{-2} + 7z^{-3} + 5z^{-4} + 4z^{-5}. \quad (16)$$

Similarly for *Structure 2* we have:

$$H_0^1(z) = H_0^2(z) = H_0^4(z) = 1 + 6z^{-1} + z^{-2}. \quad (17)$$

$$H_1^1(z) = H_1^2(z) = H_1^4(z) = 4(1 + z^{-1}). \quad (18)$$

$$H_0^3(z) = 1 + 10z^{-1} + 10z^{-2} + 10z^{-3} + z^{-4}. \quad (19)$$

$$H_1^3(z) = 5 + 11z^{-1} + 11z^{-2} + 5z^{-3}. \quad (20)$$

C. Comparison with Method [8]

In Method [8], the filter (3) works at high input rate, and has parameters: $N_1 = 7, N_2 = 9$.

The magnitude response of structure [8] is compared with magnitude responses of the proposed structures 1, and 2, at Fig. 4. a, and b, respectively.

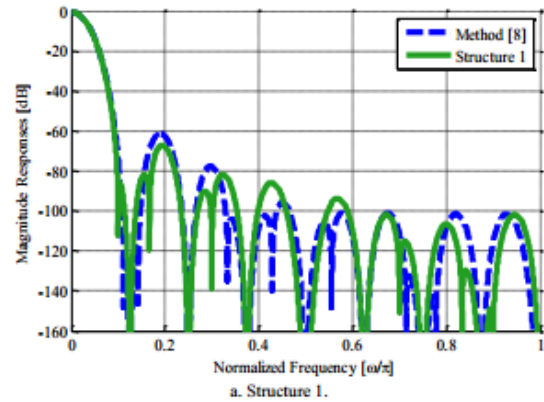
The polyphase components of the nonrecursive structure in [8] are:

$$H_0^2(z) = 1 + 6z^{-1} + z^{-2} + 4z^{-4} + 8z^{-5} + 4z^{-6} + z^{-8} + 6z^{-9} + z^{-10}. \quad (21)$$

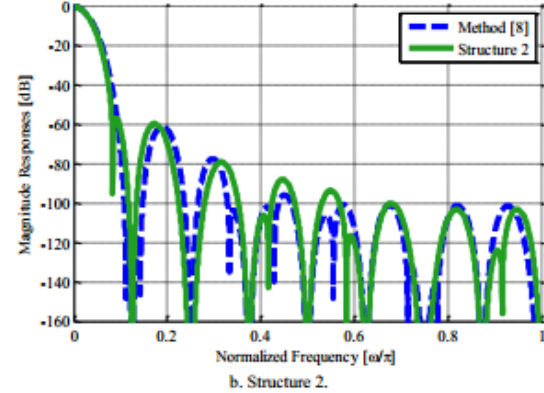
$$H_1^2(z) = 4 + 4z^{-1} + z^{-3} + 7z^{-4} + 7z^{-5} + z^{-6} + 4z^{-8} + 4z^{-9}. \quad (22)$$

$$H_0^3(z) = H_0^4(z) = H_0^5(z) = 1 + 6z^{-1} + z^{-2}. \quad (23)$$

$$H_1^3(z) = H_1^4(z) = 4(z) = 4(1 + z^{-1}). \quad (24)$$



a. Structure 1.



b. Structure 2.

Fig. 4. Comparison with Method [8].

IV. FPGA IMPLEMENTATION AND RESULTS

In this section the FPGA implementation of the proposed filter is described, and its performance is evaluated in terms of area and power consumption, using the Example 2. The corresponding filters are described using VHDL at the register transfer level (RTL). Since in all cases there are the same three identical stages (they are different only in word length), generic models were created to facilitate the design. Fig. 5 shows the model that describes a single stage. The transpose structure is used to implement the subfilters. Because the filter's coefficients are integers the multipliers have been implemented using shifts and additions. To improve the maximum operation frequency, the pipeline is used. Similarly, it has been implemented the stage containing the cosine filters.

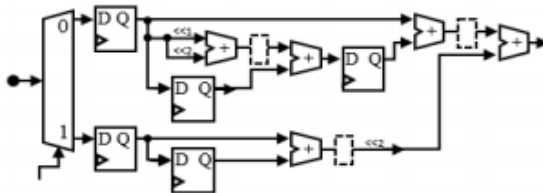


Fig. 5. RTL circuit of the m-th stage, (without cosine filters).

The filters are synthesized using ISE Design Suite 14.7, using the Artix7 (XC7A100T) FPGA device from Xilinx. The synthesis report indicates that the designs have about 460 MHz registered performance. The FPGA resource utilization is summarized in Table II. It can be observed that the requirements are low for the filters addressed in this paper. Because the *Structure 2* requires the fewest number of adders and registers, this structure uses the lowest area.

The power consumption of the implemented filters has been estimated using the tool XPower Analyzer. We considered here the dynamic power associated with design activity and switching events. The results of implementations are also summarized in Table II. As expected, the placing of the cosine filters at later stages, their operating frequency is decreased, and thus their power consumption is decreased. As a consequence, the structures proposed in this paper, have less power consumption than the structure, proposed in [8].

TABLE II

Structure	Logic utilization		Dynamic Power (mW)
	Number of slice registers	Number of slice LUTs	
1	440 (0.35 %)	323 (0.51 %)	4.25
2	412 (0.32 %)	303 (0.48 %)	3.97
[8]	439 (0.35 %)	307 (0.48 %)	4.53

V. CONCLUSIONS

A method for improving the alias rejection of the comb filter has been presented. By using the multistage non-recursive structure and cascading cosine filters at later stages, it has increased the attenuation between the folding bands. The design and implementation of the proposed filter are simple, since only three design parameters need to be determined and some stages have the same structure. The FPGA implementation confirms low resource usage. The power consumption estimations have shown that it is best to insert the cosine filters in the later stages.

ACKNOWLEDGMENT

This work is supported by the CONACYT grant No. 179587.

REFERENCES

- [1] T. Hentschel and G. Fettweis, "Sample Rate Conversion for Software Radio," *IEEE Communications Magazine*, vol. 38, no. 8, pp. 142-150, August 2000.
- [2] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, No. 2, pp. 155-162, April 1981.
- [3] G.J. Dolecek, and S. K. Mitra, "Simple method for compensation of CIC decimation filter," *Electronics Letters*, vol. 44, , No. 19, pp. 1162-1163, September 2008.
- [4] G. Molnar, and M. Vucic, "Closed-form design of CIC compensators based on maximally flat error criterion," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 58, pp. 926-930, 2011.
- [5] G. Jovanovic Dolecek and A. Fernandez-Vazquez, "Trigonometrical approach to design a simple wideband comb compensator," *Int. Journal of Electronics and Communications (AEÜ)*, vol. 68, pp. 437-441, May 2014.
- [6] L. Lo Presti, "Efficient modified sinc-filters for sigma-delta AD converters," *IEEE Trans. Circ. Syst. -II*, vol. 47, No. 11, pp. 1204-1213, Nov. 2000.
- [7] M. Laddomada, "Generalized comb decimation filters for sigma-delta A/D converters: analysis and design," *IEEE Trans. Circ. Syst. -I: Reg. Papers*, vol. 54, No. 5, pp. 994-1005, May 2007.
- [8] G. Jovanovic Dolecek and M. Laddomada, "An Improved Class of Multiplierless Decimation Filters: Analysis and Design," *Elsevier Digital Signal Processing*, vol. 23, No. 5, pp. 1773-1782, 2013.
- [9] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Transactions on Signal Processing*, vol. 45, pp. 457-467, February 1997.
- [10] G. Jovanovic Dolecek and A. Fernandez, "Novel droop-compensated comb decimation filter with improved alias rejections," *International Journal for Electronics and Communication (AEUE)*, vol.67, pp.387-396, May 2013.

c) “FPGA Implementation of Comb-Based Decimation Filter with Improved Frequency Characteristic for SD A/D Converters Application”

FPGA Implementation of Comb-Based Decimation Filter with Improved Frequency Characteristic for SD A/D Converters Application

Angel Garcia Robles, Gordana Jovanovic Dolecek *and* Jose de Jesus Rangel Magdaleno
Department of Electronics,
Institute INAOE,
Puebla, Mexico

Abstract— This paper describes a FPGA implementation of a low-complexity comb-based decimation filter with improved frequency characteristics in both, the passband and the folding bands. The attenuation across the folding bands is increased by two cosine filters in cascade, while the simple compensator, which work at low rate, decrease the resulting filter passband droop. The overall structure is multiplierless and was fitted into the Spartan 3A (3s700afg484-4) FPGA device from Xilinx, to prove a low resource implementation.

Keywords— *comb-based decimator; compensator; field programmable gate array (FPGA); Sigma-Delta A/D converter.*

I. INTRODUCTION

The oversampling Sigma-Delta-Analog/Digital (SD A/D) converters consist of analog part (modulator) and a digital part (decimation stage). In modulator the analog signal is sampled with a frequency much larger than the Nyquist frequency, which is the minimum sampling frequency required for preserving the information in analog signal. The rate of the oversampled signal must be decreased to the Nyquist frequency to be efficiently processed by the DSP (Digital Signal Processing). This process is performed in a digital format in decimation stage. The decreasing of the sampling rate may introduce the aliasing effects that deteriorate the signal and have to be eliminated by the decimation or antialiasing filter. The decimation is usually performed in more stages. The decimation filter at the first stage works at the high input rate and consequently must be simple in order to avoid a high power consumption.

The most simple decimation filter, usually used in the first decimation stage, is the Comb filter proposed by Hogenauer [1]. Nevertheless, its magnitude response presents a high passband droop and poor attenuation in the folding bands (bands around the zeros of the comb filter), which may deteriorate the decimated signal.

Different methods have been proposed to improve the magnitude response of the comb filter. Existing methods can be classified based on the band (passband or stopband or both) in which they improve the response of the comb filter. For example, the methods [2]-[4] improve the passband by cascading an additional FIR filter, commonly called compensator, with the comb filter. A simple second-order sine-based compensator was proposed in [2] and [4], while in [3] authors proposed a method for the design of compensators

which is based on maximally flat error criterion. On the other hand, the stopband is improved in [5]-[7] by introducing rotation of comb zeros. Similarly, methods [8]-[10] improve both bands. The methods [8] and [9] use sharpening technique, while method [10] uses simple corrector filters.

Nowadays, Field Programmable Gate Arrays (FPGAs) offer a very attractive solution to implement many front-end digital signal processing algorithms. FPGAs exhibit many features in common with ASICs, like reduction in area and power dissipation, higher performance and speed, and low device cost. However, FPGAs have some advantages over ASICs, such as in-system programming (ISP) and reduction in development time and costs.

The goal of this paper is two fold: Improve the frequency characteristic of comb filter in both passband and stopband and perform the FPGA implementation of the resulting comb-based decimation filter to prove a low implementation complexity. The attenuation in the comb folding bands is increased by the cascade of two cosine filters [10]. In order to reduce power consumption, downsampling and filtering are interchanged by the use of polyphase decomposition. Additionally, the passband of the overall filter is improved in the wideband by cascading simple compensator [4] at the low rate after the downsampling.

The rest of the paper is organized in the following way. The next section presents the structure of the filter. Section III describes the design of the filter. The FPGA implementation results are explained in Section IV. Finally conclusions are given in section V.

II. DESCRIPTION OF THE FILTER

The system function of the K -th order comb filter is given as:

$$H_C(z) = \left(\frac{1 - z^{-M}}{M(1 - z^{-1})} \right)^K = \left(\frac{1}{M} \sum_{i=0}^{M-1} z^{-i} \right)^K, \quad (1)$$

where M is the decimation factor. The corresponding frequency response is:

$$|H_C(e^{j\omega})| = \left| \frac{1}{M^K} \frac{\sin^K\left(\frac{\omega M}{2}\right)}{\sin^K\left(\frac{\omega}{2}\right)} \right|. \quad (2)$$

As mentioned above, this filter is very simple, but its frequency characteristic has low attenuation in the folding

This work is supported by the CONACYT grant 179587

bands and a high droop in the passband of the interest. The latter is due to the fact that for a given factor, the frequency response is fully determined by only one parameter K . In this way, the attenuation in the folding bands can be improved by increasing the value of K . However, this leads to an increase in the passband droop. In order to overcome these limitations, a simple method is presented below.

Improvement of alias rejection in the folding bands

We adopt here the result from [10] where the attenuation in the folding bands is increased by placing two additional zeros in proximity of the first zero of the comb filter. This idea can be performed by cascading two cosine filters with comb:

$$H_s(z) = \frac{1+z^{-N_1}}{2} \frac{1+z^{-N_2}}{2}, \quad (3)$$

where the parameters N_1 and N_2 are defined as:

$$N_1 = \left\lfloor \frac{M-1}{2} \right\rfloor, \quad (4)$$

$$N_2 = \left\lceil \frac{M+1}{2} \right\rceil. \quad (5)$$

Decreasing comb passband droop

The cascade of the cosine filters (3) with the comb filter results in an increased passband droop which must be compensated. The compensation filter proposed in [10] provides the passband compensation in the narrowband ($|\omega| < \pi/2M$), and provides the passband deviation of about 0.6 dB. Our goal here is to provide the compensation in the wideband region defined by the passband frequency $\pi/2M$, such that the resulting passband deviation is below 0.5 dB. To this end, we modified the compensator for the comb filter, recently proposed in [4], whose magnitude response approximates the inverse magnitude response of the comb filter in the wide passband region, and it is given by:

$$G_C(e^{j\omega}) = 1 + B \sin^2\left(\frac{\omega M}{2}\right) \approx \frac{M^K \sin^K\left(\frac{\omega}{2}\right)}{\sin^K\left(\frac{\omega M}{2}\right)}. \quad (6)$$

Knowing that in our case we have the comb (1) cascaded with the cosine filters (3), we need two compensators [4] at low rate (after downsampling by M):

$$G_1(z) = 2^{-2} [(-1 + 2z^{-1} - z^{-2})B_1 + 2^2 z^{-1}]. \quad (7)$$

$$G_2(z) = 2^{-2} [(-1 + 2z^{-1} - z^{-2})B_2 + 2^2 z^{-1}]. \quad (8)$$

The overall compensator is the cascade of filters (7) and (8):

$$G(z) = G_1(z)G_2(z). \quad (9)$$

The compensator parameters B_1 and B_2 given in (7) and (8) depend on the comb parameter K and do not depend on the decimation factor, for $M > 5$. Using the MATLAB simulations we get the parameters shown in Table I, which provide the passband deviation less than 0.5 dB.

TABLE I
THE VALUES OF B_1 and B_2 .

K	B_1	B_2
2	2^{-2}	2^{-1}
3	2^{-1}	2^{-1}
4	$\frac{3}{4} = 2^{-1} + 2^{-2}$	2^{-1}
5	$\frac{3}{4} = 2^{-1} + 2^{-2}$	$\frac{3}{4} = 2^{-1} + 2^{-2}$

Desired worst-case attenuation

In general, the Worst Case Attenuation (WCA) occurs at the frequency:

$$\omega_s = \frac{2\pi}{M} - \frac{\pi}{2M}. \quad (10)$$

In the following we present how to choose the comb parameter K to get the desired WCA. Placing (10) into the magnitude response of the proposed filter:

$$|H(e^{j\omega})| = \left| \frac{1}{M^K} \frac{\sin^K\left(\frac{\omega M}{2}\right)}{\sin^K\left(\frac{\omega}{2}\right)} \cos\left(\frac{\omega N_1}{2}\right) \cos\left(\frac{\omega N_2}{2}\right) \right| \left[1 + B_1 \sin^2\left(\frac{\omega M}{2}\right) \right] \left[1 + B_2 \sin^2\left(\frac{\omega M}{2}\right) \right] \quad (11)$$

we obtain the relation of the WCA and K depicted in Fig. 1, which is practically independent of M , for $M \geq 5$. The obtained value is rounded to the next integer because K can take only integer values.

Efficient implementation

Taking (1), (3) and (9), the system function of the overall filter, at high input rate is given as:

$$H(z) = H_C(z)H_s(z)G(z^M). \quad (12)$$

This system function can be implemented as a low-power structure. To this end, we propose using the prime factorization technique introduced in [11], and polyphase technique. The former technique can be used to factorize the decimation factor into $M = 2^p 3^q 5^r 7^s$, and thus obtaining a nonrecursive low-power structure. Using latter technique, one can obtain more power efficient structure by moving the

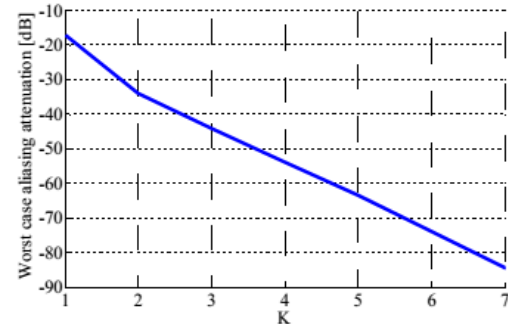


Fig. 1. Worst case attenuation of proposed decimation filter, for $M \geq 5$.

filtering after downsampling [10]. We consider here the case in which the decimation factor is an even number. We can distinguish between two cases depending of M . First case occurs for $M = 2 \cdot 3^{Q_1} 5^R = 2 \cdot (2l - 1)$. As a consequence N_1 and N_2 are even numbers, and using the multirate identity, $H_5(z)$ can be moved after downsampling by 2. The resulting power efficient structure is shown in Fig. 2(a). On the other hand, when $M = 2 \cdot (2l) = 2^p 3^{Q_1} 5^R$, N_1 and N_2 are odd numbers, and the implementation results in the structure depicted in Fig. 2(b). A particular case occurs when M is expressed as power of two, this results in the power efficient structure shown in Fig. 2(c).

III. FILTER DESIGN METHOD

The main goal of the filter proposed in this paper is advances a low complexity, multiplierless, low-power consumption decimator structure that provides improved response magnitude when compared to the comb filter.

In this sense, given the specifications of a desired antialiasing filter, namely the decimation factor $M > 5$, the minimum aliasing attenuation and the overall passband deviation in the wideband region less than 0.5dB, a simple procedure for the design of the proposed decimation filter is given in the following:

1. Choose the value of K according with Fig. 1.
2. Use (4) and (5) to compute the values of N_1 and N_2 .
3. Choose the values of B_1 and B_2 from Table I.
4. Obtain efficient structure as described in Section 2.

The method is illustrated in the following examples.

Example 1. Consider the decimation factor $M=16$, and the WCA of at least 50 dB.

According with Fig. 1, the desired attenuation can be achieved by choosing $K = 4$. From (4) and (5) we get $N_1=7$ and $N_2=9$. Similarly, from Table I, we obtain $B_1=2^{-1}+2^{-2}$, and $B_2=2^{-1}$. The magnitude responses of the improved filter and comb filter are shown in Fig. 3, along with the passband zoom. Note that the filter proposed in this paper achieves a much better magnitude response. First, the WCA has been increased about 13 dB, with respect to the WCA of comb filter. Second, the proposed filter has a maximum passband deviation of 0.35 dB, which is less than that of the comb filter (3.6 dB).

The power efficient architecture from Fig. 2(c) has 4 stages with the decimation by 2, three of which are identical. The polyphase filters have the following system functions:

$$\begin{aligned} H_0^1(z) &= (1 + z^{-8})(1 + 6z^{-1} + z^{-2}) \\ &\quad + z^{-1}[(z^{-3} + z^{-4})[4(1 + z^{-1})]], \\ H_1^1(z) &= (z^{-3} + z^{-4})(1 + 6z^{-1} + z^{-2}) \\ &\quad + [(1 + z^{-8})[4(1 + z^{-1})]], \\ H_0(z) &= 1 + 6z^{-1} + z^{-2}, \\ H_1(z) &= 4(1 + z^{-1}). \end{aligned} \quad (13)$$

Example 2. Consider the decimation factor $M=24$, and the WCA of at least 60 dB.

As a result, we obtain the following parameters: $K = 5$, $N_1=11$, $N_2=13$, and $B_1 = B_2 = 2^{-1}+2^{-2}$. In Fig. 4 is shown a comparison

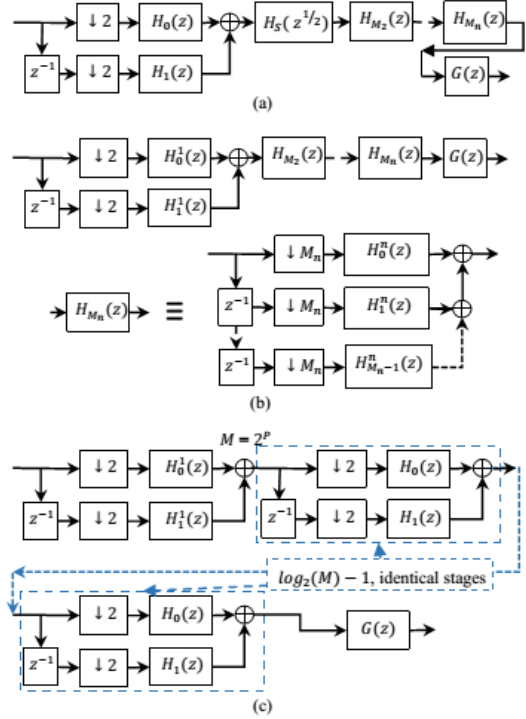


Fig. 2. Efficient architecture of decimation filter $H(z)$ in (12), (a) $M = 2 \cdot (2l - 1)$, (b) $M = 2 \cdot (2l)$, and (c) M is a power of two.

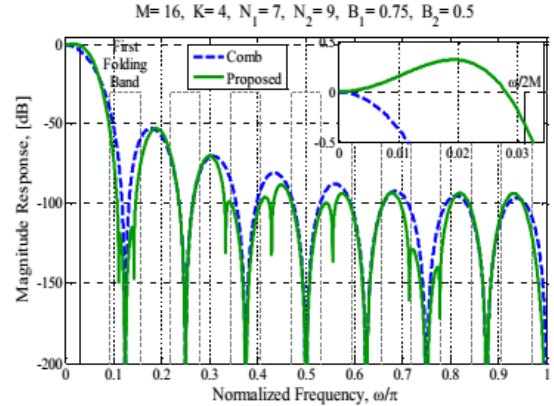


Fig. 3. Comparison of the Magnitude response of proposed filter and Comb filter.

between the magnitude responses of the proposed filter and comb filter along with the passband zoom. It can be seen that proposed filter has better magnitude characteristics. Notice that the proposed filter introduces a maximum passband deviation of 0.41 dB, while the comb filter presents a passband droop of about 4.5 dB. Likewise, the WCA has been improved for 12 dB.

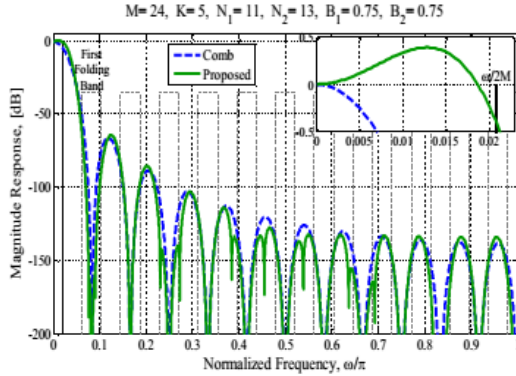


Fig. 4. Comparison of the Magnitude response of proposed filter and Comb filter.

For this example, the polyphase filters have the following system functions:

$$\begin{aligned}
 H_0^1(z) &= (1 + z^{-12})(1 + 10z^{-1} + 5z^{-2}) \\
 &\quad + z^{-1}[(z^{-5} + z^{-6})[5 + 10z^{-1} + z^{-2}]], \\
 H_1^1(z) &= (1 + z^{-12})(5 + 10z^{-1} + z^{-2}) \\
 &\quad + (z^{-5} + z^{-6})(1 + 10z^{-1} + 5z^{-2}), \\
 H_0^2(z) &= H_0^3(z) = 1 + 10z^{-1} + 5z^{-2}, \\
 H_1^2(z) &= H_1^3(z) = 5 + 10z^{-1} + z^{-2}, \\
 H_0^4(z) &= 1 + 30z^{-1} + 45z^{-2} + 5z^{-3}, \\
 H_1^4(z) &= 5 + 45z^{-1} + 30z^{-2} + z^{-3}, \\
 H_2^4(z) &= 15 + 51z^{-1} + 15z^{-2}. \quad (14)
 \end{aligned}$$

IV. FPGA IMPLEMENTATION AND RESULTS

The decimation filters from previous examples have been implemented at the register transfer level in VHDL, taking as reference the Spartan 3A (3s700afg484-4) FPGA device from Xilinx. The use of the FPGA allows a real-time low-cost implementation with the advantage of rapid prototyping. The FPGA implementation reconfigurability provides constant updating to accomplish new requirements, an open architecture for future module integration or improvements in the methodology, and a parallel structure for a fast and efficient processing that permits online implementation.

The implemented RTL circuit from Example 1 is shown in Fig. 5. This structure is implemented with fixed point and two's complement arithmetic. Additionally, in order to reduce the resources usage the multiplications are implemented using adds and shifts. The FPGA resource utilization for the FPGA Spartan 3A 3s700afg484-4 is summarized in Table II. Since the implementations do not use embedded multipliers, the registered performance is improved. From Table II is observed that the filters have the maximum frequency of 135.674 MHz. In these examples, the input of the filters is a bit stream which comes from a 1-bit sigma-delta modulator. In the Example 1, the output of the filter has a word width of 27 bits (format 2.25), while in the Example 2 the output of the filter has a word width of 34 bits (format 2.32).

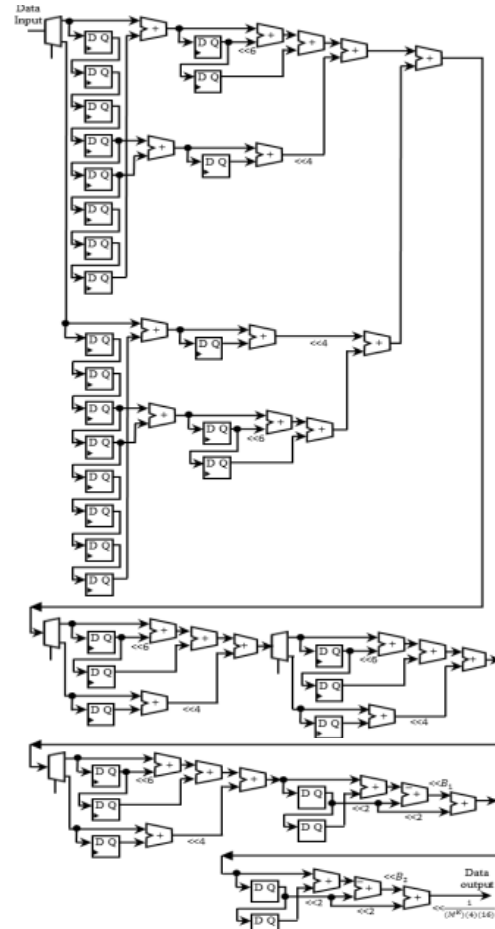


Fig. 5. RTL circuit of FPGA implementation for filter from Example 1.

TABLE II

Device utilization summary for Target Device xc3s700a-4-fg484					
Logic utilization	Used		Available	Utilization	
	Ex.1	Ex.2		Ex.1	Ex.2
Number of Slices	447	873	5888	7%	14%
Number of Slice Flip Flop	795	1569	11776	6%	13%
Number of 4 input LUTs	408	1169	11776	3%	9%
Clock period	7.37ns (Maximum Frequency: 135.67 MHz)				

In order to evaluate the implemented filter two tests were performed. First test uses simulations in MATLAB and Active-HDL. The filter performance was evaluated by a comparison between the outputs of the simulations, using as a test signal the unit-step sequence. Since neither rounding nor

truncation were used in the implementation, and neglecting the delay caused by the implemented filter, the obtained results are identical and they can be observed in Fig. 6.

In the second test, a 1-bit, 10 MHz, 2nd order sigma-delta modulator (SDM) is used for sampling an analog sinusoidal signal of 156 kHz, (the oversampling ratio is equal to 32). This analog signal is depicted in Fig. 7(a). In Fig. 7(b) is shown the output of the SDM, which is connected to the filter input of *Example 1*. The output of the filter can be observed in Fig. 7(c). This result is satisfactory, since the decimated digital signal reliably represents the analog sinusoidal signal.

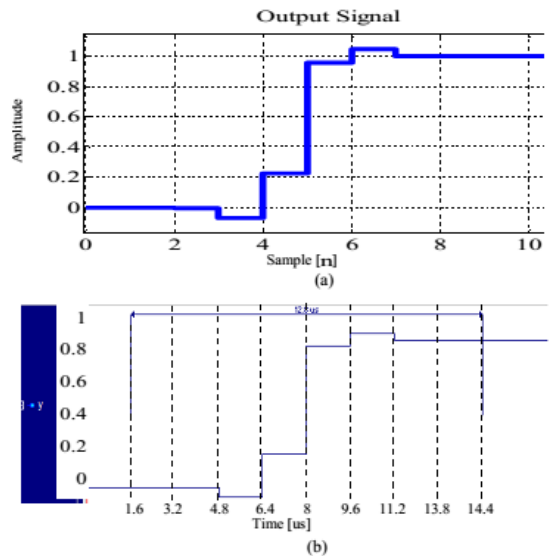


Fig. 6. Output signal of decimation filter (a) using MATLAB, (b) using Active-HDL. Test signal is a unit-step sequence.

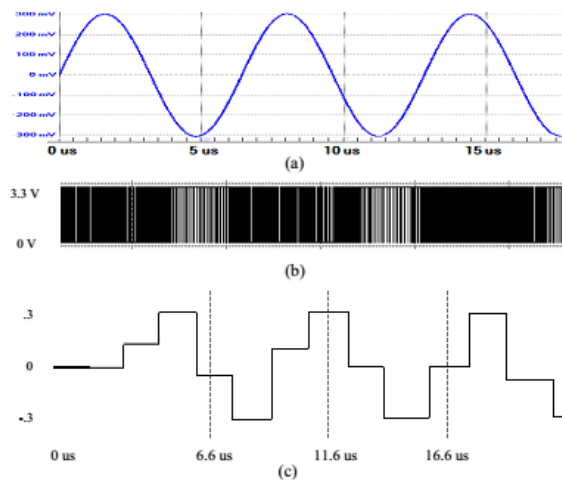


Fig. 7. Signals comparison: (a) analog signal, (b) SDM output, (c) proposed decimation filter output.

V. CONCLUSIONS

In this paper the design and implementation of a low-complexity decimation filter has been presented. The main feature of the implemented filter is its improved magnitude response in both the passband and the stopband, compared with a classical comb filter. The comb stopband is improved by introducing two cosine filters in cascade with the comb filter. Similarly, the comb passband droop is decreased by two simple compensators, both working at low rate. In addition, the design of the filter is simple, since only four design parameters need to be determined. In the FPGA implementation, the device utilization summary points out low FPGA resource usage and an optimal registered performance. The tests carried out show the proper behavior of the filter.

ACKNOWLEDGMENT

This work is supported by the CONACYT grant No. 179587.

REFERENCES

- [1] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, No. 2, pp. 155-162, April 1981.
- [2] G.J. Dolecek, and S. K. Mitra, "Simple method for compensation of CIC decimation filter," *Electronics Letters*, vol. 44, , No. 19, pp. 1162-1163, September 2008.
- [3] G. Molnar, and M. Vucic, "Closed-form design of CIC compensators based on maximally flat error criterion," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 58, pp. 926-930, 2011.
- [4] G. Jovanovic Dolecek and A. Fernandez-Vazquez, "Trigonometrical approach to design a simple wideband comb compensator," *Int. Journal of Electronics and Communications (AEÜ)*, vol. 68, pp. 437-441, May 2014.
- [5] L. Lo Presti, "Efficient modified sinc-filters for sigma-delta AD converters," *IEEE Trans. Circ. Syst. -II*, vol. 47, No. 11, pp. 1204-1213, Nov. 2000.
- [6] M. Laddomada, "Generalized comb decimation filters for sigma-delta A/D converters: analysis and design," *IEEE Trans. Circ. Syst. -I: Reg. Papers*, vol. 54, No. 5, pp. 994-1005, May 2007.
- [7] F. Daneshgaran and M. Laddomada, "A novel class of decimation filters for sigma delta A/D converters," *Wireless Communications and Mobile Computing*, 2002, vol. 2, pp. 867-882.
- [8] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Transactions on Signal Processing*, vol. 45, pp. 457-467, February 1997.
- [9] Q. Liu, and J. Gao, "Efficient comb decimation filter with sharpened magnitude response," In Proc. 5th International Conference on Wireless Communications, Networking and Mobile Computing WICOM, pp. 1-4, 2011.
- [10] G. Jovanovic Dolecek and M. Laddomada, "An Improved Class of Multiplierless Decimation Filters: Analysis and Design," *Elsevier Digital Signal Processing*, vol. 23, No. 5, pp. 1773-1782, 2013.
- [11] Jang, Y., Yang, S.: 'Non-recursive cascaded integrator-comb decimation filters with integer multiple factors'. Proc. 44th IEEE Midwest Symp. On Circuits and Systems, Dayton, OH, August 2001, pp. 130-133.

Lista de figuras

Figura 1.1 Diagrama típico de un decimador.....	1
Figura 1.2 Ilustración en el dominio de la frecuencia de los efectos del submuestreo. a) Espectro original de la señal. b) Espectro de la señal submuestreada.	3
Figura 1.3 Decimador multietapa.....	4
Figura 1.4 Diagrama a bloques del SD-ADC.....	5
Figura 1.5 Modelo del Modulador Sigma Delta en tiempo discreto.	7
Figura 1.6 Gráficas de la función de transferencia de ruido de un SDM.	7
Figura 2.1 Localización de polos y ceros del filtro comb.	10
Figura 2.2 Respuestas de magnitud de filtros Comb de diferente orden.....	11
Figura 2.3 Características que exhibe el filtro comb al ser utilizado en el proceso de decimación.	13
Figura 2.4 Estructura de un filtro decimador CIC de K etapas.	15
Figura 2.5 Diagrama de bloques de un filtro comb no recursivo multietapas.....	17
Figura 2.6 Estructura de un decimador multietapas basado en comb no recursivo.	17
Figura 2.7 Estructura de un decimador polifásico.....	18
Figura 3.1 Respuesta de magnitud de filtros coseno.	22
Figura 3.2 Ilustración del método para mejorar la respuesta de magnitud del filtro comb a través de la conexión en cascada de filtros coseno.....	24
Figura 3.3 Respuesta de magnitud de pre-filtros coseno.	25
Figura 3.4 Ilustración del método para incrementar la atenuación en las folding bands utilizando un pre-filtro coseno.	26
Figura 3.5 Ilustración del método para incrementar la atenuación en las folding bands utilizando dos pre-filtros coseno.....	27
Figura 3.6 Ilustración del método para incrementar la atenuación en las folding bands utilizando filtros coseno rotado.	29
Figura 3.7 Respuestas de magnitud del filtro comb y del filtro HCSz del ejemplo 3.5.	30
Figura 3.8 Compensación del filtro comb utilizando método [22].	33
Figura 3.9 Respuestas de magnitud de filtro comb no compensado y compensado con diferentes longitudes del compensador (L) y números de términos (P), utilizando método [23].	35
Figura 3.10 Filtro comb compensado utilizando el método [24].	37
Figura 4.1 Respuestas de magnitud de los filtros del ejemplo 4.1.	41
Figura 4.2 Comparación entre las respuestas magnitud del filtro propuesto y el filtro comb.....	44
Figura 4.3 Estructura del filtro propuesto.	45
Figura 4.4 Arquitectura propuesta.....	47
Figura 4.5 Área estimada del filtro propuesto ante diferentes valores de M y modos de factorizarlo.	50
Figura 4.6 Potencia estimada del filtro propuesto ante diferentes valores de M y modos de factorizarlo.....	50
Figura 4.7 Estructura multietapa de un filtro comb con $M = 2P$	51

Figura 4.8 Arquitectura del filtro propuesto para $M = 2P$.	52
Figura 4.9 Arquitectura eficiente del filtro propuesto para $M = 2P$.	53
Figura 4.10 Arquitectura general para los filtros del ejemplo 4.3.	54
Figura 4.11 Respuestas de magnitud de los filtros del ejemplo 4.3.	55
Figura 4.12 Respuestas de magnitud de los filtros del ejemplo 4.4.	56
Figura 4.13 Estructura del decimador propuesto	58
Figura 4.14 Arquitectura del filtro del ejemplo 4.4.	59
Figura 4.15 Magnitud del filtro del ejemplo 4.4.	59
Figura 4.16 Comparación entre el filtro propuesto con método [18].	60
Figura 4.17 Comparación entre el filtro propuesto con método [19].	61
Figura 4.18 Comparación entre el filtro propuesto con método [20].	62
Figura 5.1 (a)Arquitectura de la primera etapa del decimador propuesto. (b)Modelo equivalente RTL usando un demultiplexor en la entrada.	64
Figura 5.2 Realización de un filtro no recursivo en su forma directa transpuesta.	65
Figura 5.3 Diagrama RTL de la primera etapa del decimador propuesto considerando $M1 = 4$ y $K1 = 3$.	66
Figura 5.4 (a)Multiplicadores necesarios para el subfiltro polifásico $H01(z)$ y su modelo equivalente utilizando sumas y corrimientos. (b)Multiplicadores necesarios para el subfiltro polifásico $H21(z)$ y su modelo equivalente utilizando sumas y corrimientos.	67
Figura 5.5 Estructura de la segunda etapa del decimador propuesto.	68
Figura 5.6 Arquitectura de la descomposición polifásica de los filtros coseno.	68
Figura 5.7 Arquitectura de la descomposición polifásica de los filtros coseno.	69
Figura 5.8 Arquitectura del filtro compensador.	70
Figura 5.9 Arquitectura del decimador implementado en el Ejemplo 5.2.	71
Figura 5.10Comparación de respuestas al impulso obtenidas con Matlab y con Active.	72
Figura 5.11 Comparación de respuestas al escalón obtenidas con Matlab y con Active.	73
Figura 5.12 Arquitectura del decimador implementado en el Ejemplo 5.3.	74

Lista de tablas

Tabla 2.1 Relación entre el orden del filtro comb, K , con la DBP y el PCA	14
Tabla 3.1 Valores de B que satisfacen $\delta \leq 0.4$ dB.	32
Tabla 4.1 Número mínimo de sumadores necesarios para implementar los coeficientes del filtro comb con parámetros $M1$ y $K1$	48
Tabla 4.2 Diferentes formas de elegir $M1$ y $M2$, para $M = 24$	51
Tabla 4.3 Parámetros correspondientes a las diferentes estructuras para $P=3, 4, 5$ y 6	53
Tabla 4.4 Valores de B	57
Tabla 5.1 Resumen de utilización de recursos del decimador implementado en el Ejemplo 5.2. El dispositivo destino es el FPGA Artix 7 XC7A100T de Xilinx.	73
Tabla 5.2 Resumen de utilización de recursos del decimador implementado en el Ejemplo 5.3. El dispositivo destino es el FPGA Artix 7 XC7A100T de Xilinx.	74

Referencias

- [1] N. J. Fliege, *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Waveltes*, John Wiley & Sons, England, 1995.
- [2] G. Jovanovic Dolecek, *Multirate Systems: Design and Applications*, Idea Group Publishing, Hershey USA, 2002.
- [3] S. K. Mitra, *Digital Signal Processing: A Computer Based Approach*. 3rd edition, McGraw-Hill, New York, 2006.
- [4] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, John Wiley & Sons, USA, 2005.
- [5] B. Leung, *The Oversampling Technique For Analog to Digital Conversion: A tutorial Overview*. *Analog Integrated Circuits and Signal Processing*, Kluwer Academi Publisher, January 1991.
- [6] R. Gaggl, *Delta-Sigma A/D-Converters*, Springer-Verlag Berlin Heidelberg, 2013.
- [7] L. Milić, *Multirate Filtering for Digital Signal Processing: MATLAB Applications*, IGI Global, USA, 2009.
- [8] M. Laddomada, "Generalized Comb Decimation Filters for $\Sigma\Delta$ A/D Converters: Analysis and Design," *IEEE Transactions on Circuits and Systems*, vol. 54, No. 5, pp. 994–1005, May 2007.
- [9] E. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, No. 2, pp. 155-162, April 1981.
- [10] H. Aboushady, Y. Dumonteix, M.M. Louerat, and H. Mehrez. "Efficient Polyphase Decomposition of Comb Decimation Filters in Sigma-Delta Analog-to-Digital Converters," *IEEE Transactions on Circuits and Systems-II*, vol. 48, No. 10, pp. 898-903, 2001.
- [11] S. Mitra, M. Tchobanou and M. Bryukhanov, "A General Method for Designing Sparse Antenna Arrays," *Proc. 17th European Conf. On Circuit Theory and Design*, Cork, Ireland, vol. II, pp. 263–266, August 2005

- [12] G. Jovanovic Dolecek and A. Fernandez-Vazquez, "Novel droop-compensated comb decimation filter with improved alias rejections," *Int. Journal of Electronics and Communications (AEÜ)*, vol.67, pp.387-396, 2013.
- [13] M. G. Pecotic, G. Molnar and M. Vucic, "Design of CIC Compensators with SPT Coefficients Based on Interval Analysis," in *MIPRO, 2012 Proceedings of the 35th International Convention*, pp.123-128, 21-25 May 2012.
- [14] G. J. Dolecek, "Simple Wideband CIC Compensator," *Electronics Letters*, vol. 45, no. 24, pp. 1270–1272, November 2009.
- [15] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of Filter Sharpening to Cascaded Integrator-Comb Decimation Filters," *IEEE Transactions on Signal Processing*, vol. 45, pp. 457-467, February 1997.
- [16] L. Lo Presti, "Efficient Modified Sinc-Filters for Sigma-Delta AD Converters," *IEEE Trans. Circ. Syst. –II*, vol. 47, No. 11, pp. 1204-1213, November 2000.
- [17] Y. Lian and Y. C. Lim, "New Prefilter Structure for Designing FIR Filters," *Electronic Letters*, vol. 29, pp. 1034-1036, May 1993.
- [18] G. Dolecek and M. Laddomada, "Comb-Cosine Prefilter Based Decimation Filter," *Proc. IEEE Int. Conf. Industrial Technology*, pp. 205–210, March 2010.
- [19] G. Dolecek and J. Carmona, "A New Cascaded Modified CIC-Cosine Decimation Filter," *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 4, pp. 3733–3736, May 2005.
- [20] Teran Bahena, E.Y.; Jovanovic Dolecek, G., "Zero-Rotation-Based Nonrecursive Comb Structure," *Proc. IEEE 57th International Midwest Symposium*, pp.797-800, 3-6 Aug. 2014.
- [21] G. Jovanovic Dolecek, and M. Laddomada, "An Improved Class of Multiplierless Decimation Filters: Analysis and Design," *Elsevier Digital Signal Processing*, vol. 23, pp. 1773-1782, September 2013.
- [22] G. Jovanovic Dolecek and A. Fernandez-Vazquez, "Trigonometrical Approach to Design a Simple Wideband Comb Compensator," *Int. Journal of Electronics and Communications (AEÜ)*, vol. 68, pp. 437-441, May 2014.

- [23] G. Molnar, and M. Vucic, "Closed-Form Design of CIC Compensators Based on Maximally Flat Error Criterion," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 58, pp. 926-930, 2011.
- [24] D. E. Troncoso Romero and G. Jovanovic Dolecek, "Application of Amplitude Transformation for Compensation of Comb Decimation Filters," *Electronic Letters*, vol. 49 No. 16, pp. 985-987, August 2013.
- [25] G. M. Salgado, et al., "Power and Area Efficient Comb-based Decimator for $\Sigma\Delta$ ADCs with High Decimation Factors," *Proc. of the 2013 Int. Symp. on Circ. and Syst. (ISCAS)*, pp.1260-1263, May 2013.
- [26] M. Abbas, O. Gustafsson, and L. Wanhammar, "Power Estimation of Recursive and Non-recursive CIC Filters Implemented in Deep-Submicron Technology," *Proc. First IEEE Conference on Green Circuits and Systems, ICGCS2010*, June 2010, pp. 221-225.
- [27] Y. Voronenko and M. Püschel, "Multiplierless Multiple Constant Multiplication," *ACM Transactions on Algorithms*, vol. 3, No. 2, pp. 1-39, May 2007.