



**I
N
A
O
E**

Mejora de Estructura Comb Basada en Rotación de Ceros No Recursiva

por

Erika Yazmin Teran Bahena

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)**

Agosto de 2014
Santa María Tonantzintla, Puebla

Supervisada por:

Dra. Gordana Jovanovic Dolecek, INAOE

© INAOE 2014

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



Resumen

La reducción de la tasa de muestreo de una señal digital se conoce como decimación. Este proceso introduce aliasing si la señal original no tiene un ancho de banda limitado. Por lo tanto, es necesario acotar la señal de entrada usando un filtro pasa-bajas antes de disminuir su frecuencia de muestreo. En la literatura, este filtro es llamado filtro anti-aliasing o decimador. Generalmente, las especificaciones en su diseño son rigurosas, lo que provoca el inconveniente de requerir en su implementación, una estructura compleja con un alto consumo de hardware y potencia. Una manera eficiente de realizar este proceso consiste en dividir el filtrado en varias etapas.

En este trabajo se estudian las características del filtro decimador más popular conocido como comb. Se dan a conocer las razones que hacen a este filtro ideal para ser usado en las primeras etapas del proceso de decimación. También, se exponen las desventajas que presenta su respuesta en magnitud, las cuales, han motivado la necesidad de desarrollar métodos de diseño que mejoren dicha característica. Posteriormente, se revisan las diversas técnicas que dan solución a las limitantes del filtro comb.

Considerando todo lo anterior, en este trabajo se presenta una propuesta para el diseño de filtros decimadores no recursivos basados en filtros comb. Estos poseen características en magnitud mejoradas. En el diseño de su estructura, se incluye el uso de un filtro compensador y se introduce un filtro coseno rotado. Este método es aplicable cuando factor de decimación puede ser expresado como potencia de dos. Su funcionamiento es ilustrado a través de ejemplos, y se realizan comparaciones con otros métodos existentes.

Finalmente, se expone la metodología usada para la descripción en VHDL y posterior implementación en FPGA de los filtros propuestos. La comprobación de su correcto diseño se lleva a cabo por medio de simulaciones, y los recursos requeridos en su implementación son dados a conocer.

Abstract

The process of reducing the sampling rate of a digital signal is called decimation. This process introduces aliasing if the original signal is not bandlimited. Therefore, it is necessary to limit the input signal with a low-pass filter before reducing its sampling frequency. In literature, this filter is known as anti-aliasing filter or decimation filter. Generally, the specifications for its design are rigorous. This results in the disadvantage of requiring for its implementation a complex structure with high cost in hardware and power consumption. An efficient way to perform this process is made by dividing the filtering in various stages.

In this work the characteristics of the most popular decimation filter are studied. This filter is called comb filter. The reasons why comb filter is used in the first stages of the decimation process are presented. The disadvantages of its magnitude response are also illustrated. These disadvantages have motivated the development of methods to design decimation filters with improved magnitude characteristics. Subsequently, it is presented a general review of methods for improving the limitations of the comb filter.

A method to design comb-based non-recursive decimation filters, with improved magnitude characteristics is presented. In the design of the structure of the proposed filters, a compensation filter is considered and a rotated cosine filter is introduced. This method can be applied if the decimation factor is a power of two. The functionality of this method is illustrated by examples, and comparisons with other methods are provided.

Finally, the methodology to describe in VHDL and implement the proposed filters in FPGA is given. In order to prove the correct design, simulations are made and the resources required in its implementation are listed.

Agradecimientos

Al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) por todas las facilidades otorgadas durante la duración de mis estudios de maestría.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado a través de la beca de maestría.

A la Dra. Gordana Jovanovic Dolecek por haber confiado en mí, por los valiosos consejos y conocimientos que compartió conmigo, y por guiarme para culminar exitosamente este trabajo.

A mis sinodales; el Dr. Roberto Rosas Romero, Dr. Esteban Tlelo Cuautle y Dr. José de Jesús Rangel Magdaleno, por invertir su valioso tiempo en la revisión de esta tesis.

A mis padres y a mi hermano, quienes han hecho todo lo posible e imposible para apoyarme en cada momento.

A mis amigas de toda la vida, a los amigos a quienes conocí gracias al gusto que compartimos por esta profesión, y a los amigos que han vivido día a día conmigo la experiencia de estudiar una maestría.

Índice General

Resumen	ii
Abstract	iv
Agradecimientos	vi
Índice General	viii
Prefacio	xii
CAPÍTULO 1. El Filtro Comb	1
1.1 Función de Transferencia.....	1
1.2 Respuesta en Magnitud.....	2
1.3 Respuesta en Fase.....	4
1.4 El Filtro Comb como Filtro Decimador.....	5
1.4.1 Decimación.....	5
1.4.2 Características del Filtro Comb como Filtro Decimador.....	8
1.5 Implementación de un Filtro Comb.....	9
1.5.1 El Filtro CIC (Cascaded-Integrator-Comb).....	9
1.5.2 El Filtro Comb No Recursivo.....	11
1.6 Resumen de las Ventajas y Desventajas del Filtro Comb como Filtro Decimador.....	13
CAPÍTULO 2. Métodos para Mejorar la Respuesta en Magnitud del Filtro Comb	14
2.1 Introducción.....	14
2.2 Métodos para Mejorar la Respuesta en Magnitud.....	15
2.2.1 Métodos para Mejorar la Banda de Paso.....	15
2.2.1.1 Filtro Compensador de Banda Ancha.....	15

2.2.2 Incremento en la Atenuación de las Bandas de Rechazo.....	18
2.2.2.1 Pre-filtro Coseno.....	18
2.2.2.2 El Filtro Sinc Rotado (RS).....	20
2.2.2.3 El Filtro Comb Generalizado (GCF).....	22
2.2.3 Mejora en la Respuesta en Magnitud en Ambas Bandas.....	24
2.2.3.1 Aplicación de la Técnica de Sharpening al Filtro Comb.....	24
2.2.3.2 Método de Optimización de la Función de Cambio de Amplitud.....	26
2.2.3.3 Aplicación de un Filtro IIR para Mejorar la Respuesta en Magnitud.....	29
2.3 Reducción en el Consumo de Potencia y Hardware.....	31
2.3.1 Multiplicadores a Base de Sumas y Corrimientos.....	31
2.3.2 Representación de Coeficientes como Dígitos con Signo Canónico.....	32
CAPÍTULO 3. Método Propuesto.....	34
3.1 Filtro Coseno Expandido.....	34
3.1.1 Filtro Coseno Expandido Rotado.....	37
3.2 Filtro Compensador.....	39
3.3 Estructura Propuesta.....	43
3.3.1 Respuesta en Magnitud de la Estructura Propuesta.....	45
3.3.2 Comparación con el Filtro Comb Generalizado (GCF).....	49
3.4 Diseño de la Estructura Propuesta sin Multiplicadores.....	50
3.4.1 Comparación con Otros Métodos Libres de Multiplicadores.....	52
3.5 Complejidad Computacional de la Estructura Propuesta.....	54
CAPÍTULO 4. Descripción en VHDL de los Filtros Decimadores Propuestos.....	56
4.1 Introducción.....	56
4.2 El Divisor de Frecuencia.....	57
4.3 Descripción de un Filtro Comb No Recursivo con $M = 2^p$	59
4.3.1 Celda Básica de un Filtro Comb No Recursivo con $M = 2^p$	59
4.3.2 Consideraciones Previas a la Descripción en VHDL del Filtro Comb No Recursivo..	60
4.3.2.1 Incremento en el Tamaño del Dato de Entrada.....	60
4.3.2.2 Frecuencias de Trabajo.....	62
4.3.2.3 Bloque para la Implementación de la Primera Etapa del Filtro Comb	
No Recursivo.....	62
4.3.2.4 Bloque para la Implementación de las Etapas Restantes del Filtro Comb	
No Recursivo.....	63

4.3.3 Descripción en VHDL del Filtro Comb No Recursivo.....	64
4.3.3.1 Descripción de la Primera Etapa del Filtro Comb No Recursivo.....	64
4.3.3.2 Descripción de las $p - 1$ Etapas Restantes del Filtro Comb No Recursivo.....	65
4.4 Estructura General para la Implementación de un Filtro Decimador Diseñado con el Método Propuesto.....	66
4.4.1 Descripción del Filtro Coseno Rotado.....	67
4.4.2 Descripción del Filtro Compensador	69
CAPÍTULO 5. Implementación y Simulación de los Filtros Decimadores Propuestos.....	71
5.1 Introducción.....	71
5.2 Implementación de un Filtro Comb No Recursivo con $M = 32$ y $K = 4$	72
5.2.1 Respuesta al Impulso del Filtro Comb No Recursivo con $M = 32$ y $K = 4$	73
5.2.2 Respuesta al Escalón del Filtro Comb No Recursivo con $M = 32$ y $K = 4$	75
5.2.3 Recursos Requeridos en la Implementación del Filtro Comb No Recursivo con $M = 32$ y $K = 4$	76
5.3 Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 1$	76
5.3.1 Respuesta al Impulso del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 1$	78
5.3.2 Respuesta al Escalón del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 1$	79
5.3.3 Recursos Requeridos en la Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 1$	80
5.4 Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 2$	80
5.4.1 Respuesta al Impulso del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 2$	81
5.4.2 Respuesta al Escalón del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 2$	82
5.4.3 Recursos Requeridos en la Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 2$	84
5.5 Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 3$	84
5.5.1 Respuesta al Impulso del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 3$	85
5.5.2 Respuesta al Escalón del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 3$	87
5.5.3 Recursos Requeridos en la Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 3$	88
5.6 Comparaciones.....	88
Conclusiones.....	90

Trabajo a Futuro.....	91
Lista de Figuras.....	92
Lista de Tablas.....	96
Referencias.....	99

Prefacio

El proceso de modificar la tasa de muestreo de una señal digital se conoce como conversión de tasa de muestreo. En la actualidad, muchos sistemas incluyen esta operación para llevar a cabo el procesamiento digital de las señales de una forma más eficiente. Los convertidores analógico-digitales sigma-delta, los sistemas de comunicaciones móviles y satelitales, aplicaciones médicas, los transmisores y receptores de la radio definida por software entre otros, son ejemplos de ello.

Esta modificación en frecuencia puede ser un aumento (interpolación) o un decremento (decimación). En el caso particular de la reducción de frecuencia de una señal digital, el bloque que requiere especial atención en su diseño es el filtro decimador, ya que este tiene un gran peso en la complejidad de la estructura necesaria para realizar este proceso.

En los últimos años los decimadores no recursivos basados en filtros comb, han recibido una especial atención debido a la sencillez de su estructura, bajo consumo de potencia y alta velocidad de procesamiento. Sin embargo, las pobres características que presenta su respuesta en magnitud, han propiciado la necesidad de encontrar métodos de diseño que permitan mejorar dicha respuesta.

El objetivo de esta tesis, consiste en desarrollar un método de diseño de estructuras de decimación no recursivas basadas en filtros comb. El principal criterio a tomar en cuenta es mejorar su respuesta en magnitud, tanto en la banda de paso como en las bandas de rechazo. Esta estructura no debe de necesitar multiplicadores, y en su implementación se considera un factor de decimación expresable como potencias de dos.

Para alcanzar este objetivo, en el capítulo 1 se estudia a detalle el funcionamiento del filtro comb y su aplicación dentro del proceso de decimación. Posteriormente, en el capítulo 2, se procede a investigar las diversas técnicas que se han propuesto para el diseño eficiente de filtros decimadores. Basándose en los resultados de estos dos capítulos, en el capítulo 3 se desarrolla un método para el diseño de filtros decimadores no recursivos, que presentan una respuesta en magnitud mejorada. De esta forma, se cumple con el objetivo de este trabajo.

En los capítulos 4 y 5 se proporcionan las herramientas necesarias para la implementación en FPGA de los filtros decimadores propuestos. También, se realizan una serie de simulaciones para verificar su correcto funcionamiento.

Finalmente, se presentan las conclusiones generales y se plantea el trabajo a futuro.

Capítulo 1

El Filtro Comb

En este capítulo se describirán a detalle la características del filtro comb. Se darán a conocer su función de transferencia, respuesta en frecuencia y parámetros de diseño. Se analizarán las ventajas y desventajas que se tienen al utilizar el filtro comb como primer bloque dentro del proceso de decimación. Finalmente, se realiza una comparación entre las configuraciones comúnmente usadas en su implementación, y se enlistan las consideraciones que se tienen que hacer previo al diseño de su hardware.

1.1. Función de Transferencia

El filtro comb es un filtro de respuesta finita al impulso con fase lineal (FIR). Se caracteriza por no requerir de multiplicadores, ni el almacenamiento de coeficientes en su implementación. Esto lo hace atractivo en aplicaciones donde es necesario trabajar con altas frecuencias de muestreo, como por ejemplo, los convertidores analógico-digitales sigma-delta (ADCs- $\Sigma\Delta$) [1].

Su función de transferencia en forma recursiva es:

$$H(z) = \left[\frac{1}{M} \frac{1 - z^{-M}}{1 - z^{-1}} \right]^K, \quad (1.1)$$

donde M indica su orden y K el número de filtros comb conectados en cascada. Cuando este filtro es aplicado en el proceso de decimación, como se explicará más adelante, M es conocido como el factor de decimación. También, es posible representar su función de transferencia en forma no recursiva. Esta se expresa como:

$$H(z) = \left[\frac{1}{M} \sum_{i=0}^{M-1} z^{-i} \right]^K. \quad (1.2)$$

1.2. Respuesta en Magnitud

La respuesta en frecuencia del filtro comb se obtiene a partir de sustituir $z = e^{j\omega}$ en la función de transferencia (1.1):

$$H(e^{j\omega}) = \left[\frac{1}{M} \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}} \right]^K = \frac{1}{M^K} \left[\frac{e^{-j\omega M/2} (e^{j\omega M/2} - e^{-j\omega M/2})}{e^{-j\omega/2} (e^{j\omega/2} - e^{-j\omega/2})} \right]^K. \quad (1.3)$$

Aplicando la identidad de Euler, $2j\sin(x) = e^{jx} - e^{-jx}$, la expresión (1.3) puede ser reescrita como:

$$H(e^{j\omega}) = \frac{1}{M^K} \left[\frac{e^{-j\omega M/2} 2j\sin(j\omega M/2)}{e^{-j\omega/2} 2j\sin(j\omega/2)} \right]^K = \left[\frac{1}{M} \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} e^{j\omega[(M-1)/2]} \right]^K, \quad (1.4)$$

donde el término exponencial $e^{j\omega[(M-1)/2]}$ define la respuesta en fase del filtro. Su correspondiente respuesta en magnitud es:

$$|H(e^{j\omega})| = \left| \frac{1}{M} \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right|^K. \quad (1.5)$$

Se observa que, los ceros del filtro comb se localizan en los múltiplos enteros de la frecuencia $\omega = 2\pi/M$:

$$\omega_k = 2\pi k/M, \quad k = 1, 2, \dots, M - 1, \quad (1.6)$$

Estos generan los llamados dobleces de banda o bandas de rechazo. Dentro del intervalo $[0, \pi]$ se distribuyen uniformemente $M/2$ ceros, si el orden del filtro es par. En caso contrario, se tendrán $(M - 1)/2$ ceros. Para ejemplificar esto, en la Fig. 1.1 se muestra la respuesta en magnitud de un filtro comb con $M = 12$ y $K = 1$. Se hace un acercamiento a la banda de paso y primera banda de rechazo del filtro comb.

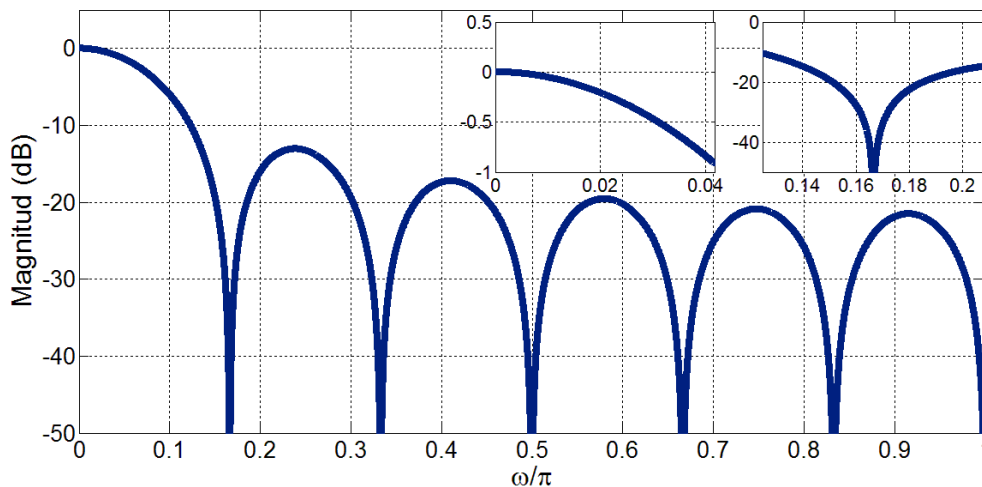


Fig. 1.1. Respuesta en magnitud del filtro comb ($M = 12$, $K = 1$).

Se observa que existen seis ceros dentro del intervalo $[0, \pi]$. Estos se ubican en las frecuencias definidas por (1.6). Una propiedad notable, es que la respuesta en magnitud del filtro comb tiene características de un filtro pasa-bajas. Sin embargo, presenta una prolongada banda de transición y una rápida caída en la banda de paso. Además, existe poca atenuación en las bandas de rechazo, principalmente en la primera de ellas, que en este caso es de aproximadamente 10.4 dB.

El parámetro de diseño que permite mejorar de forma sencilla la atenuación es K . Sin embargo, aumentar el número de filtros comb en cascada tiene como consecuencia un incremento en la caída de la banda de paso.

En la Fig. 1.2, se muestra la respuesta en magnitud del filtro comb ($M = 8$) para diferentes valores de K .

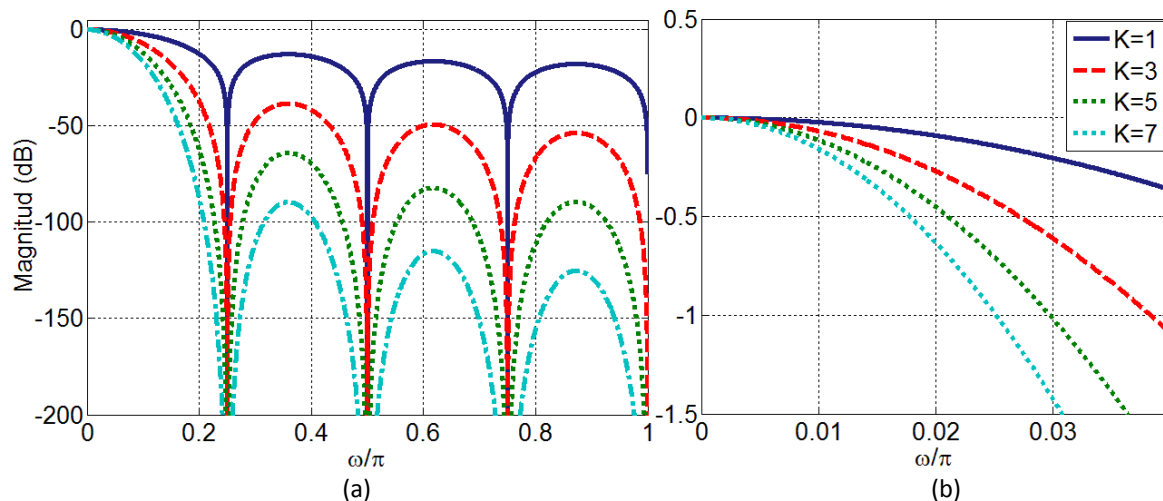


Fig. 1.2. Respuesta en magnitud de un filtro comb para diferentes valores de K . (a) Todo el intervalo de frecuencias. (b) Banda de paso.

En las Tablas 1.1 y 1.2, se incluyen los valores de la caída en la banda de paso y atenuación en la primera banda de rechazo de un filtro comb para diferentes valores de M y K .

M	Caída en la banda de paso en función del número de etapas (dB)					
	1	2	3	4	5	6
128	0.00	0.00	0.00	0.00	0.00	0.01
64	0.00	0.01	0.01	0.01	0.02	0.02
32	0.01	0.03	0.04	0.06	0.07	0.08
16	0.06	0.11	0.17	0.22	0.28	0.34
8	0.22	0.45	0.67	0.90	1.12	1.35
4	0.91	1.82	2.74	3.65	4.56	5.47

Tabla 1.1. Caída en la banda de paso.

M	Atenuación en la primera banda de rechazo en función del número de etapas (dB)					
	1	2	3	4	5	6
128	42.1	84.2	126.2	168.3	210.4	252.5
64	36.0	72.0	108.0	144.0	180.0	215.9
32	29.8	59.7	89.5	119.4	149.2	179.0
8	17.1	34.3	51.4	68.5	85.6	102.8
4	10.5	20.9	31.4	41.8	52.3	62.7

Tabla 1.2. Atenuación en la primera banda de rechazo.

Estos resultados permiten apreciar el compromiso que existe entre el incremento de atenuación y caída en la banda de paso.

1.3. Respuesta en Fase

La respuesta en fase del filtro comb está definida por el término $e^{j\omega[(M-1)/2]}$ de su respuesta en frecuencia. Esto nos indica que su fase es lineal y el retardo de grupo es constante.

$$Retardo_{grupo} = \frac{M - 1}{2}. \tag{1.7}$$

En el caso de K filtros comb conectados en cascada, el retardo de grupo está dado por la siguiente expresión:

$$(Retardo_{grupo})^K = \frac{K(M - 1)}{2}. \tag{1.8}$$

A pesar de tener una parte recursiva en su estructura, el filtro comb es estable. Esto se debe a que el polo ubicado en $z = 1$ dentro del círculo unitario, introducido por el integrador, se cancela perfectamente con el cero de la parte no recursiva. Esto se ilustra en la Fig. 1.3, donde se presenta la ubicación de polos y ceros de un filtro comb con $M = 12$ y $K = 1$.

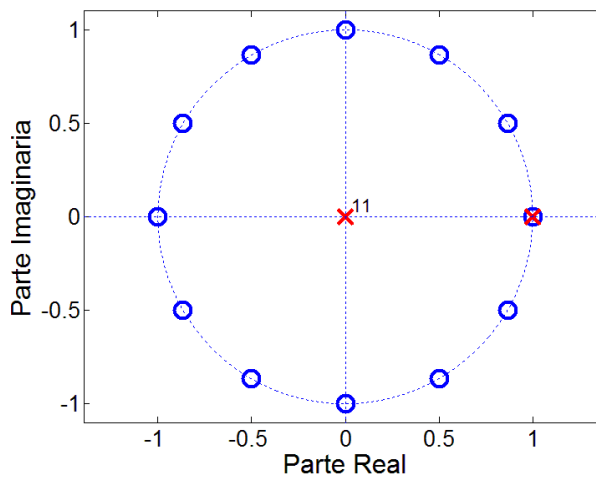


Fig. 1.3. Ubicación de los polos y ceros de un filtro comb ($M = 12, K = 1$).

1.4. El Filtro Comb como Filtro Decimador

En las secciones anteriores se dieron a conocer las características en frecuencia del filtro comb. A continuación se describe brevemente, en el dominio del tiempo y frecuencia, en que consiste el proceso de decimación. De esta forma, se mostrará la funcionalidad del filtro comb como filtro decimador.

1.4.1. Decimación

El proceso de reducir la tasa de muestreo de una señal digital, se conoce como decimación [2]. Este se divide en dos etapas: filtrado y sub-muestreo, como se muestra en la Fig. 1.4.

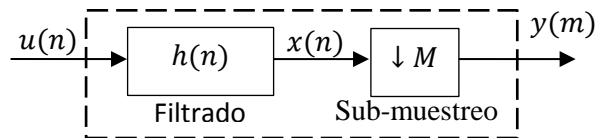


Fig. 1.4. Decimación.

En la etapa de sub-muestreo, la frecuencia de muestreo f_s de la señal de entrada es reducida por un factor entero M , que es conocido como factor de decimación. La señal de salida $y(m)$ se obtiene conservando sólo las muestras múltiplos enteros de M y descartando las demás.

$$y(m) = x(mM). \quad (1.9)$$

El proceso de sub-muestreo a su vez se divide en dos pasos. En el primero de ellos, las muestras que no son múltiplos de M se vuelven cero. Esto equivale a multiplicar, en tiempo discreto, la señal original de entrada por la secuencia $c_M(n)$.

$$x'(n) = x(n)c_M(n), \quad (1.10)$$

donde

$$c_M(n) = \begin{cases} 1 & n = mM \\ 0 & \text{otros casos} \end{cases} ; m = \dots, -1, 0, 1, \dots \quad (1.11)$$

Debido a que la secuencia $c_M(n)$ es periódica con período M , esta se puede expresar mediante la serie discreta de Fourier:

$$c_M(n) = \frac{1}{M} \sum_{k=0}^{M-1} C(k) e^{j\frac{2\pi kn}{M}}, \quad (1.12)$$

donde $C(k)$ son los coeficientes complejos de la serie de Fourier definidos como:

$$C(k) = \sum_{n=0}^{M-1} c_M(n) e^{-\frac{j2\pi kn}{M}}. \quad (1.13)$$

Sustituyendo (1.11) en (1.13) se deduce que $C(k) = 1$ para toda k . Por lo tanto:

$$c_M(n) = \frac{1}{M} \sum_{k=0}^{M-1} e^{\frac{j2\pi kn}{M}}. \quad (1.14)$$

Con el fin de analizar que sucede en el dominio de la frecuencia en este primer paso, se calcula la transformada discreta de Fourier de (1.10).

$$X'(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x'(n) e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x(n) c_M(n) e^{-j\omega n}. \quad (1.15)$$

Sustituyendo (1.14) en (1.15) se obtiene:

$$X'(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) \left(\frac{1}{M} \sum_{k=0}^{M-1} e^{\frac{j2\pi kn}{M}} \right) e^{-j\omega n}. \quad (1.16)$$

Reorganizando (1.16) esta se puede expresar como:

$$X'(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{n=-\infty}^{\infty} x(n) e^{-jn\left(\omega - \frac{2\pi k}{M}\right)}. \quad (1.17)$$

$$X'(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{j\left(\omega - \frac{2\pi k}{M}\right)}\right). \quad (1.18)$$

Se observa que el espectro de la señal $x(n)$ se repite cada $2\pi/M$ y sufre un escalamiento en magnitud por $1/M$. Por lo tanto, es necesario que la señal de entrada este limitada en frecuencia hasta π/M . En caso contrario, las réplicas de los espectros se traslapan, ocasionando la distorsión de la señal de interés. Este fenómeno es conocido como aliasing.

En el segundo paso del sub-muestreo, los ceros introducidos anteriormente, son eliminados. Es así como la señal de entrada sufre una compresión en tiempo por un factor de $1/M$. En el dominio de la frecuencia, esto equivale a una expansión del espectro por M . En consecuencia, el espectro de la señal de salida es:

$$Y(e^{jM\omega}) = X'(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{j\left(\omega - \frac{2\pi k}{M}\right)}\right). \quad (1.19)$$

En la Fig. 1.5 se presenta el proceso de sub-muestreo con $M = 4$, tanto en el dominio del tiempo como en el dominio de la frecuencia. Se observa que la señal original tiene un espectro limitado, y este no sobrepasa la frecuencia $\pi/M = \pi/4$. Por lo tanto, no se presenta el fenómeno de aliasing.

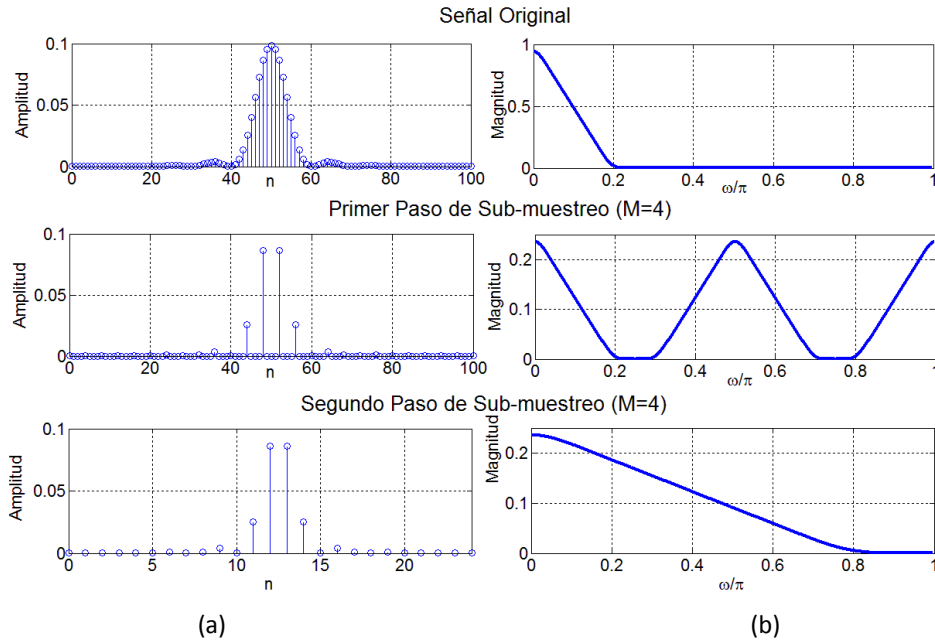


Fig. 1.5. Proceso de sub-muestreo ($M = 4$). (a) Dominio del tiempo. (b) Dominio de la frecuencia.

En la Fig. 1.6 se considera sub-muestrear la señal de entrada por un factor $M = 8$. Esta señal tiene componentes de frecuencia por encima de $\pi/M = \pi/8$. En consecuencia, se presenta el fenómeno de aliasing.

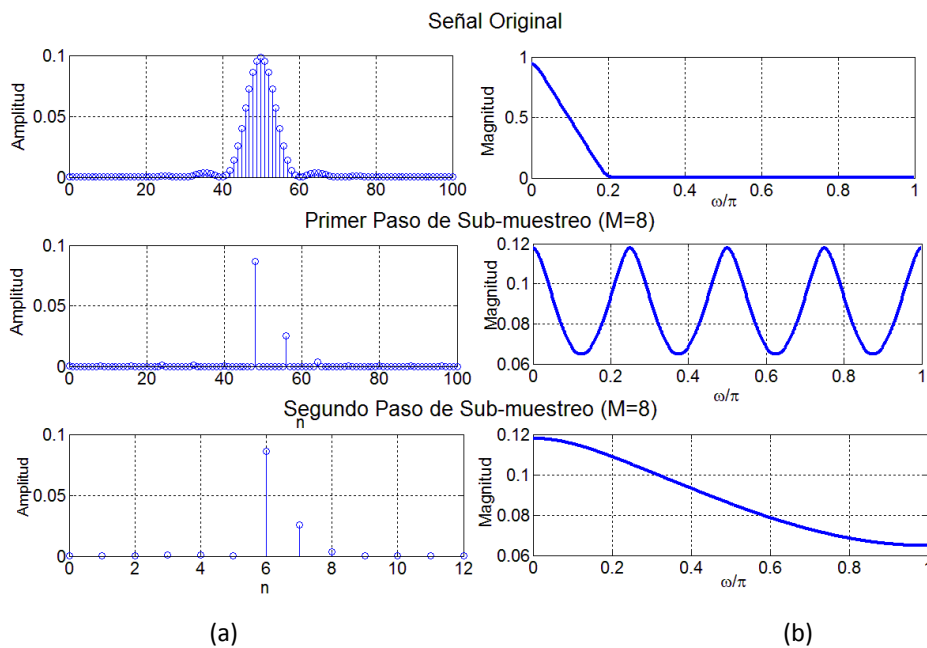


Fig. 1.6. Efecto de aliasing debido al proceso de sub-muestreo ($M = 8$). (a) Dominio del tiempo. (b) Dominio de la frecuencia.

Para evitar que la señal de entrada sea dañada por el aliasing, es necesario limitar su frecuencia con filtro pasa-bajas. Este debe de cumplir con las siguientes características:

$$|H(e^{j\omega})| = \begin{cases} 1, & |\omega| \leq \omega_c/M \\ 0, & \text{en otro caso.} \end{cases} \quad (1.20)$$

donde ω_c es la máxima frecuencia de $x(n)$ que se desea preservar.

1.4.2. Características del Filtro Comb como Filtro Decimador

El filtro comb tiene características que lo hacen idóneo para ser usado en la primera etapa del proceso de decimación. Esto se debe a que sus ceros se ubican exactamente a la mitad del ancho de banda de las réplicas de la señal de entrada, que se producen debido al submuestreo.

En la Fig. 1.7 se muestra la respuesta ideal que se desea en la primera etapa del filtro decimador, así como, la respuesta de un filtro comb con $M = 12$ y $K = 1$. Se considera usar el valor mínimo que puede tomar el factor residual, es decir $v = 2$ (este factor indica el número por el cual, las etapas posteriores al filtro comb reducen la tasa de muestreo de la señal de entrada). A partir de esta imagen se definen conceptos relacionados con el uso del filtro comb como filtro decimador.

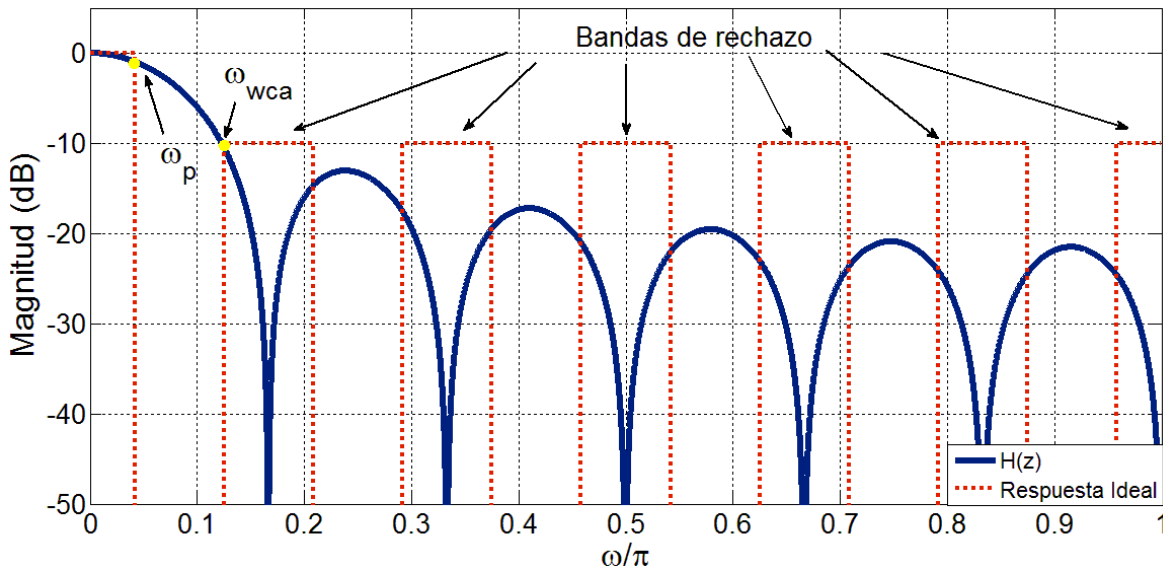


Fig. 1.7. Respuesta en magnitud ideal en la primera etapa del filtro decimador y de un filtro comb ($M = 12$, $K = 1$, $v = 2$).

Estos conceptos son:

1. *Frecuencia de paso ω_p* : La frecuencia de paso del filtro comb, tiene una dependencia con el factor de decimación M y con el factor residual v . Esta se define como:

$$\omega_p = \frac{\pi}{Mv}. \quad (1.21)$$

2. *Bandas de rechazo*: Estas se localizan alrededor de los ceros del filtro comb y se definen como:

$$\left[\frac{2\pi k}{M} - \omega_p; \frac{2\pi k}{M} + \omega_p \right], \quad \text{para } k = \begin{cases} 1, \dots, M/2 & M \text{ par} \\ 1, \dots, (M-1)/2 & M \text{ impar} \end{cases} \quad (1.22)$$

3. *Peor caso de atenuación ω_{wca}* : Para muchas aplicaciones, se considera que el peor error debido al aliasing es consecuencia de la poca atenuación existente en la primera banda de rechazo. La frecuencia donde se produce esto se calcula a partir de:

$$\omega_{wca} = \frac{2\pi}{M} - \omega_p, \quad (1.23)$$

y tiene una dependencia con el factor de decimación M y el factor residual v .

1.5. Implementación de un Filtro Comb

Las configuraciones comúnmente usadas en la implementación de un filtro comb se derivan a partir de su función de transferencia. En este apartado se describen la estructura recursiva y no recursiva del filtro comb, así como, las consideraciones que se tienen que realizar previo a su implementación.

1.5.1. El Filtro CIC (Cascade-Integrator-Comb)

El filtro CIC, propuesto por E. B. Hogenauer en 1981 [3], es la forma eficiente en el consumo de recursos para implementar un filtro comb. La estructura del filtro CIC es recursiva y consiste en dos etapas. La primera de ellas es conocida como la etapa integradora. Esta se trata de una conexión en cascada de K integradores que trabajan a una alta frecuencia de muestreo f_s . Su función de transferencia se escribe como:

$$H_I(z) = \left(\frac{1}{1-z^{-1}} \right)^K. \quad (1.24)$$

La segunda etapa es conocida como la etapa diferenciadora. Al igual que la etapa anterior, es una conexión en cascada de K diferenciadores o filtros comb, que trabajan a una baja frecuencia de muestreo f_s/M . La función de transferencia de esta etapa se expresa a través de:

$$H_c(z) = (1 - z^{-1})^K. \tag{1.25}$$

Entre ambas etapas se encuentra un reductor de razón de muestreo, el cual, generalmente es un interruptor. Este disminuye la frecuencia de muestreo por M . En la Fig. 1.8 se muestra el diagrama a bloques de la estructura del filtro CIC.

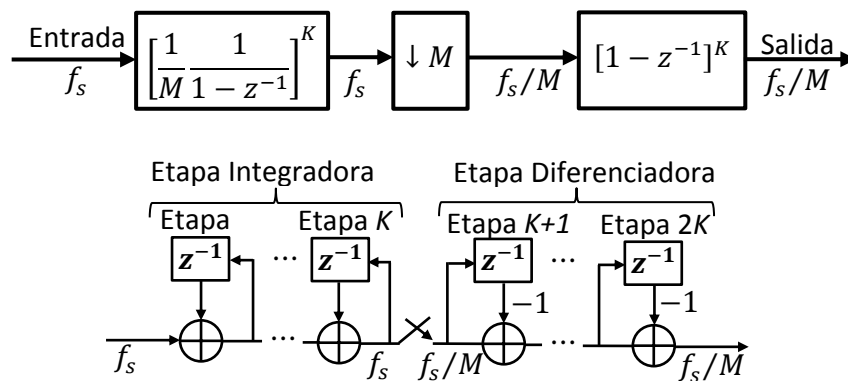


Fig. 1.8. Diagrama a bloques de la estructura del filtro CIC.

Nótese que la implementación de una etapa ($K = 1$) del filtro CIC requiere sólo de dos sumadores.

Consideraciones para la Implementación de la Estructura CIC

Las arquitecturas convencionales utilizadas para implementar un integrador y un derivador en un filtro CIC se muestran en la Fig. 1.9a y 1.9b respectivamente.

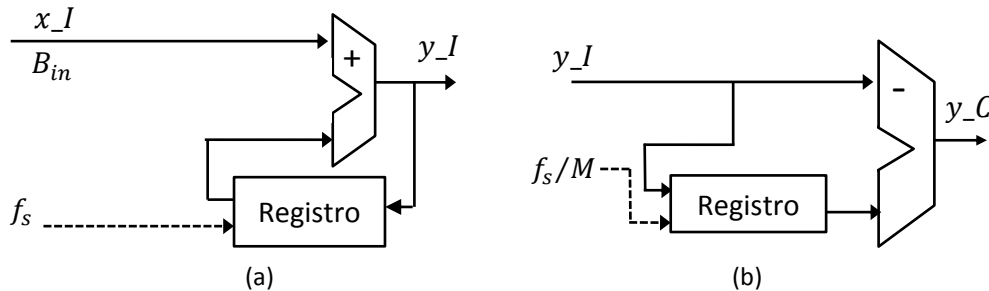


Fig. 1.9. Arquitecturas para la implementación de (a) un integrador, (b) un derivador.

E. Hogenauer [3] expone las consideraciones que se tienen que realizar en la implementación de un filtro CIC (Fig. 1.8). En primer lugar, los filtros CIC a pesar de poseer una parte recursiva, no sufren de inestabilidad debido a que el único coeficiente de los integradores es unitario.

Sin embargo, la retroalimentación que se produce en la sección de los integradores, provoca que ocurra un desbordamiento en los registros que se utilizan en su implementación. Esto puede tener como consecuencia, la pérdida de la información durante el procesamiento de la señal, si no toma en cuenta lo siguiente:

1. Se debe emplear la aritmética necesaria para poder ignorar el desbordamiento que ocurre en las operaciones intermedias. Esto puede realizarse por medio de una aritmética en base a complemento a 2 o magnitud y signo entre otras.
2. La longitud de palabra utilizada por los integradores, debe de ser lo suficientemente grande para poder representar la magnitud máxima esperada en la señal de salida.

Por lo tanto, para un filtro CIC, el tamaño de palabra requerido por los registros para evitar perder información durante los desbordamientos se encuentra dada por:

$$B_{max} = [K \log_2 M + B_{in} - 1], \quad (1.26)$$

donde B_{in} es la cantidad de bits de entrada.

La implementación física de un filtro CIC tienen desventajas y ventajas, a continuación estas son enlistadas:

Ventajas:

- Requiere de poca cantidad de recursos de hardware y por lo tanto ocupa un área de chip menor.

Desventajas:

- Existe un desbordamiento en los registros utilizados por los integradores.
- Tiene un mayor consumo de potencia, debido a que todos los integradores trabajan a una alta frecuencia de muestreo.
- Para factores de decimación grandes, el tamaño de palabra de procesamiento, así como de los registros crece considerablemente.

Una forma de evitar el desbordamiento en los registros, es implementar el filtro comb en forma no recursiva (1.2).

1.5.2. El Filtro Comb No Recursivo

La estructura no recursiva del filtro comb más popular, es aquella en la cual el factor de decimación es expresado como potencias de dos ($M = 2^p$). Su función de transferencia está dada por:

$$H(z) = \left[\frac{1}{M} \sum_{i=0}^{M-1} z^{-i} \right]^K = \left[\prod_{i=0}^{p-1} \frac{1}{2} (1 + z^{-2^i}) \right]^K. \quad (1.27)$$

También se han desarrollado estructuras no recursivas para los casos donde el factor de decimación puede expresarse en términos de potencias de tres ($M = 3^p$):

$$H(z) = \left[\frac{1}{M} \sum_{i=0}^{M-1} z^{-i} \right]^K = \left[\prod_{i=0}^{p-1} \frac{1}{3} (1 + z^{-3^i} + z^{-2 \cdot 3^i}) \right]^K. \quad (1.28)$$

Los diagramas a bloques para la implementación de estas estructuras se muestran en las Fig. 1.10a y 1.10b, respectivamente.

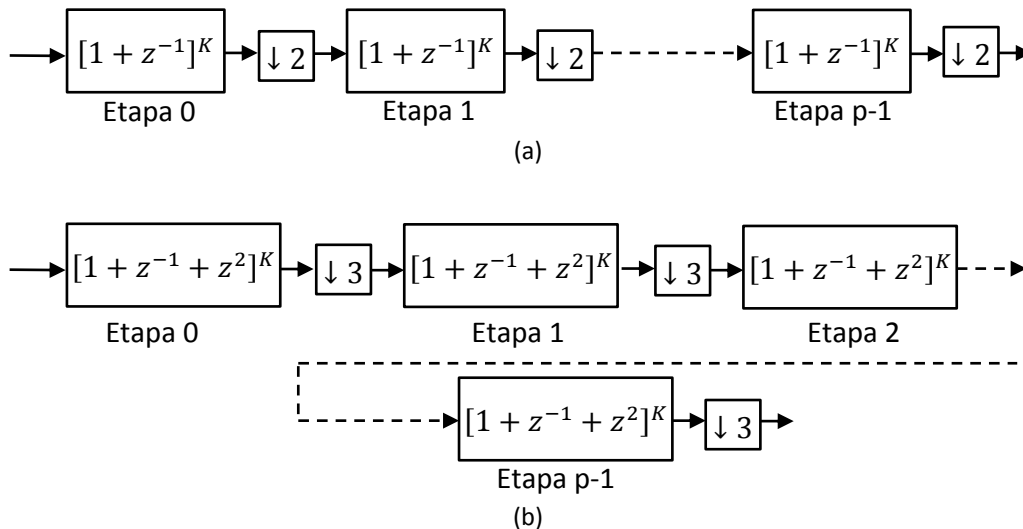


Fig. 1.10. Estructura del filtro comb no recursivo para (a) $M = 2^p$, (b) $M = 3^p$.

Consideraciones en la Implementación de un Filtro Comb No Recursivo

A diferencia de la estructura CIC, las estructuras no recursivas no sufren desbordamientos en los registros utilizados en su implementación. El tamaño de palabra de procesamiento de cada una de las etapas es diferente. La primera etapa requiere el tamaño de palabra menor. En las etapas posteriores la palabra de procesamiento aumentará su longitud. Este incremento depende del máximo valor en amplitud de la señal de salida.

La implementación física de un filtro comb en forma no recursiva tienen desventajas y desventajas. A continuación estas son enlistadas:

Ventajas:

- Se evita el problema del desbordamiento de los registros, ya que se elimina la parte recursiva del filtro CIC.
- Considerando que el factor de decimación puede representarse como potencias de dos o de tres, el filtro comb en forma no recursiva puede implementarse en varias sub-etapas, donde cada una de ellas es precedida por un factor de decimación. De esta forma se irá reduciendo de forma gradual la frecuencia de muestreo de cada sub-filtro, logrando así una reducción en el consumo de potencia de la estructura completa.

Desventajas:

- Requiere una mayor cantidad de recursos de hardware.
- Estas estructuras están limitadas a ser usadas sólo en casos donde el factor de decimación cumple con la condición de ser potencia de dos o tres.

1.6. Resumen de las Ventajas y Desventajas del Filtro Comb como Filtro Decimador

A continuación se enlistan las ventajas y desventajas que ofrece el filtro comb al ser utilizado como un filtro decimador:

Ventajas

- Estructura libre de multiplicadores.
- No requiere del almacenamiento de coeficientes.
- Posee fase lineal y un retardo de grupo constante.
- Puede ser expresado e implementado tanto en forma recursiva como no recursiva.
- Sus ceros se localizan justo a la mitad de cada una de las réplicas de espectro de la señal de entrada que se producen durante el proceso de sub-muestreo.

Desventajas

- Presenta una rápida caída en la banda de paso. Esto provoca distorsión en la señal de entrada.
- Tiene poca atenuación en las bandas de rechazo, principalmente en la primera de ellas.
- Las características de su respuesta en magnitud están determinadas sólo por dos parámetros de diseño, M y K .

Capítulo 2

Métodos para Mejorar la Respuesta en Magnitud del Filtro Comb

En este capítulo se describen algunos de los métodos que se han propuesto para mejorar la respuesta en magnitud del filtro comb. Estos se clasifican en aquellos que buscan disminuir la caída en la banda de paso, incrementar la atenuación en las bandas de rechazo, o mejorar ambas características al mismo tiempo. Con la finalidad de ilustrar las ventajas y desventajas que estos ofrecen, se presentarán ejemplos de diseño donde son aplicados. Posteriormente, se mostrarán algunas de las técnicas más utilizadas para reducir los recursos de hardware y el consumo de potencia del filtro comb. Estas consisten principalmente en buscar una representación eficientemente de los coeficientes.

2.1. Introducción

El filtro comb es idóneo para ser utilizado en las primeras etapas del proceso de decimación. Esto se debe a la simplicidad que posee su estructura. Sin embargo, las pobres características que presenta su respuesta en magnitud, son intolerables en algunas aplicaciones del procesamiento digital. Principalmente, en aquellas donde se exige una alta selectividad. Por lo tanto, se han desarrollado diversos métodos que permiten mejorar su respuesta, sin incrementar de forma considerable la complejidad de su estructura. Algunos de estos métodos son descritos a lo largo de este capítulo, y por medio de ejemplos se muestran las ventajas y desventajas que ofrecen. Finalmente, se presentan algunas de las técnicas utilizadas para reducir el consumo de potencia y el hardware requerido en la implementación del filtro comb.

2.2. Métodos para Mejorar la Respuesta en Magnitud

2.2.1. Métodos para la Mejora de la Banda de Paso

La caída en la banda de paso del filtro comb se reduce usando un filtro compensador. Este es colocado en cascada con el filtro comb y tiene que cumplir con los siguientes requerimientos:

- Trabajar en la frecuencia de muestreo más baja que sea posible.
- Su estructura no debe necesitar multiplicadores.
- Su diseño debe de ser simple e independiente de los valores de M y K . Esto permite el uso de un mismo filtro compensador para diferentes filtros comb.

Distintos métodos han sido desarrollados con el fin de obtener filtros compensadores como el anteriormente descrito. Estos se clasifican en forma general como métodos de compensación de banda angosta ($v > 2$) y de banda ancha ($v = 2$).

En [4] un filtro FIR de tercer orden es usado como compensador. Sus coeficientes se obtienen utilizando técnicas de optimización. Sin embargo, estos no son números enteros o expresables en potencias de dos. En consecuencia, se requieren multiplicadores en su implementación.

Los métodos [5] y [6] describen el diseño de compensadores de banda ancha basados en coseno. En ellos se aplica redondeo a sus coeficientes, para que estos puedan ser implementados con sumas y corrimientos. Por otro lado, en [7] se presenta un método de diseño de compensadores de segundo y cuarto orden. Este método se extiende tanto para compensadores de banda ancha como de banda angosta. Los coeficientes de estos los filtros son calculados con el método de optimización de máximo aplanamiento, y su diseño libre de multiplicadores es descrito.

El concepto de función de cambio de amplitud es aplicado por primera vez a filtros compensadores en [8]. También se han desarrollado métodos de diseño iterativos como el presentado en [9], donde los coeficientes obtenidos son implementados como dígitos con signo canónico. Finalmente en [10] se muestra un compensador basado en seno, el cual sólo requiere de tres sumadores en su implementación.

En el siguiente apartado se ilustra un método de diseño de un filtro compensador.

2.2.1.1. Filtro Compensador de Banda Ancha

En [10] se define la respuesta en magnitud del filtro compensador como:

$$|G(e^{j\omega M})| = 1 + B\sin^2(\omega M/2). \quad (2.1)$$

Su correspondiente función de transferencia es:

$$G(z^M) = 2^{-2}[-B + (2^2 + 2B)z^{-M} - Bz^{-2M}], \quad (2.2)$$

donde B es una constante positiva. Utilizando propiedades de estructuras multi-tasa el compensador se coloca en baja frecuencia:

$$G(z) = 2^{-2}[-B + (2^2 + 2B)z^{-1} - Bz^{-2}]. \quad (2.3)$$

Reorganizando (2.3) se obtiene (2.4), que es la forma eficiente de implementar el compensador.

$$G(z) = 2^{-2}[(-1 + 2z^{-1} - z^{-2})B + 2^2z^{-1}]. \quad (2.4)$$

Usando (2.2), la función de transferencia de un filtro comb compensado es:

$$H_c(z) = H(z)G(z^M). \quad (2.5)$$

En este método de diseño se considera que la banda de paso puede dividirse en N intervalos:

$$\omega_k = \frac{\omega_p}{N} k = \frac{\pi}{2MN} k; \quad k = 1, \dots, N. \quad (2.6)$$

Sustituyendo (2.6) en (2.5), y recordando que $\sin \alpha \approx \alpha$ para valores pequeños de α . Se obtiene:

$$|H_c(e^{j\omega_k})| \approx \left[\frac{\sin(\pi k/4N)}{\pi k/4N} \right]^K \left[1 + B \sin^2 \left(\frac{\pi k}{4N} \right) \right]. \quad (2.7)$$

De (2.7) se observa que el diseño de $H_c(z)$ no depende de M pero si de K . El valor B se escoge de tal forma que el valor absoluto de la caída en la banda de paso, sea menor a un determinado valor δ (dB), es decir:

$$20 \log_{10} |H_c(e^{j\omega_k})| \leq \delta [dB], \quad \omega_k \in [0, \dots, \omega_p], \quad (2.8)$$

donde $\omega_p = \pi/(2M)$. Despejando B en (2.8):

$$B \leq \frac{10^{\delta/20} (\pi k/(4N))^K - [\sin(\pi k/(4N))]^K}{[\sin(\pi k/(4N))]^K \sin^2(\pi k/4N)}, \quad (2.9)$$

y definiendo B_k como:

$$B_k \leq \frac{10^{\delta/20} (\pi k/(4N))^K - [\sin(\pi k/(4N))]^K}{[\sin(\pi k/(4N))]^K \sin^2(\pi k/4N)}. \quad (2.10)$$

El objetivo es encontrar el valor de B_k que cumpla con (2.8). Por lo tanto, en el diseño de este filtro compensador se deben de conocer previamente K , N y δ . En la Tabla 2.1 se incluyen los valores de B para diferentes valores de K con $\delta = 0.4 \text{ dB}$. Estos fueron obtenidos realizando simulaciones en *MATLAB*® con el fin de garantizar la mejor compensación en la banda de paso.

K	$B = \min_k \{B_k\}, k = 1, \dots, N; N = 50$
1	0.321
2	0.5677
3	0.8035
4	1.0372
5	1.2696

Tabla 2.1. Coeficiente B para diferentes valores de K .

El diseño libre de multiplicadores del filtro compensador, es posible si los valores de B son redondeados tal y como se muestra en la Tabla 2.2.

K	$B = \min_k \{B_k\}, k = 1, \dots, N; N = 50$	Redondeo
1	0.321	$0.25 < 0.321$
2	0.5677	$0.5 < 0.5677$
3	0.8035	$0.75 < 0.8035$
4	1.0372	$1 < 1.0372$
5	1.2696	$1.25 < 1.2696$

Tabla 2.2. Coeficiente B redondeado para diferentes valores de K .

Finalmente, los valores de B se expresan como potencias de dos para facilitar su implementación a base de sumas y corrimientos (Véase Tabla 2.3).

K	B	Total de sumas
1	2^{-2}	3
2	2^{-1}	3
3	$2^{-1} + 2^{-2}$	4
4	2^0	3
5	$2^0 + 2^{-2}$	4

Tabla 2.3. Valores de B en términos de potencias de dos.

Ejemplo 2.1. Considérese el diseño de dos filtros comb compensados. El primero de ellos tiene como parámetros de diseño $M = 12$, $K = 4$ y $B = 1$. El segundo $M = 32$, $K = 4$ y $B = 1$. En las Fig. 2.1a y 2.2b se observan sus respuestas en magnitud. Se realiza una comparación con la respuesta de un filtro comb sin compensar.

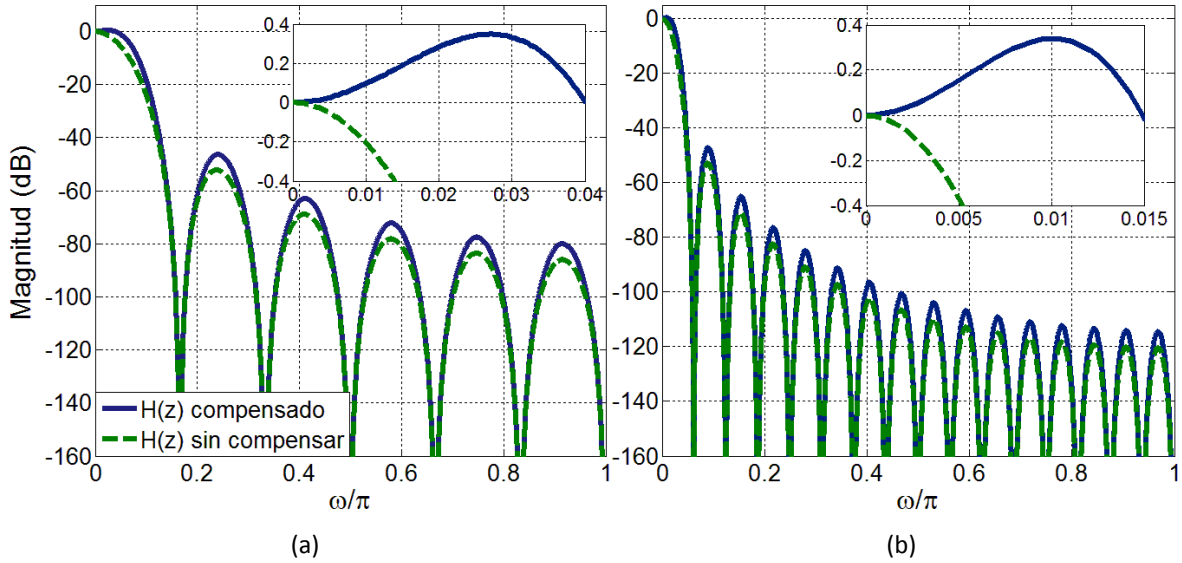


Fig. 2.1. Respuestas en magnitud del Ejemplo 2.1. (a) $M = 12, K = 4, B = 1$, (b) $M = 32, K = 4, B = 1$.

Es posible ver, que existe una disminución en la caída de la banda de paso. Además, se comprueba que el diseño del filtro compensador no depende del factor de decimación.

2.2.2. Incremento en la Atenuación de las Bandas de Rechazo

La atenuación de las bandas de rechazo del filtro comb, no sólo se mejora incrementando el número de filtros en cascada. Existen diversos métodos con los que se obtienen mejores resultados. Estos agregan determinadas modificaciones en la estructura del filtro comb. A continuación se describen algunos de ellos.

2.2.2.1. Pre-filtro Coseno [11]

El pre-filtro coseno ha sido ampliamente utilizado en diversas estructuras basadas en filtros comb [12]-[15]. En general, estas se caracterizan por tener una considerable atenuación en las bandas de rechazo. La función de transferencia del pre-filtro coseno es:

$$H_{COS}(z^N) = 0.125(1 + z^{-2N})(1 + z^{-N})^2, \tag{2.11}$$

donde N es un número entero positivo. Los coeficientes de (2.11) son unitarios, por lo tanto, no requiere de multiplicadores en su implementación. Su correspondiente respuesta en magnitud es:

$$|H_{COS}(e^{j\omega})| = 0.5|\cos(N\omega) + \cos^2(N\omega)|. \tag{2.12}$$

En la Fig. 2.2 se muestra la respuesta del pre-filtro coseno para diferentes valores de N .

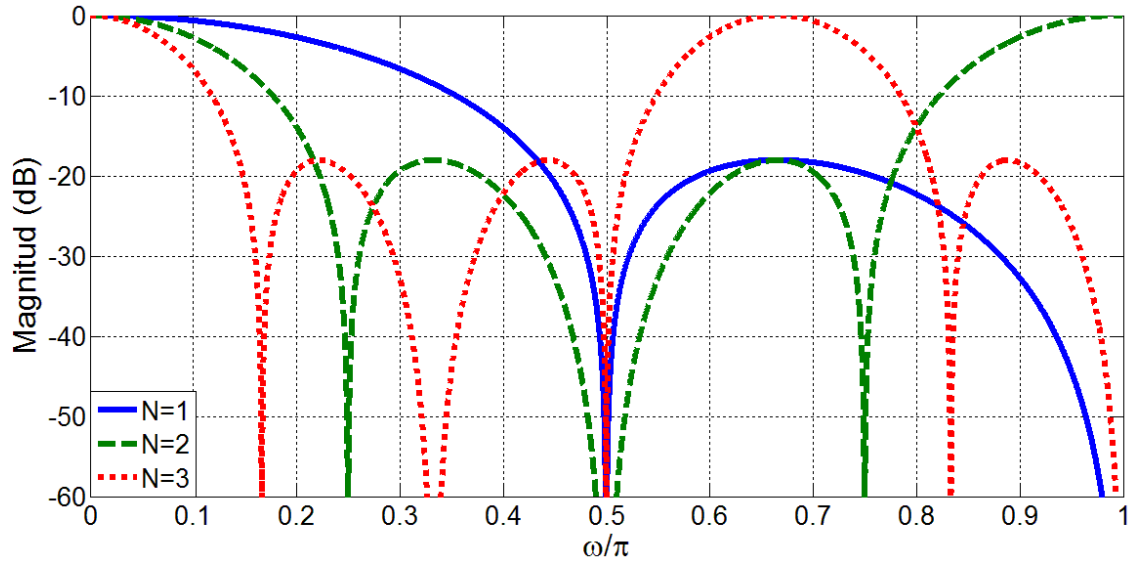


Fig. 2.2. Respuesta en magnitud del pre-filtro coseno para diferentes valores de N .

Si un filtro comb es seguido por un pre-filtro coseno, ciertas bandas de rechazo tendrán mejor atenuación cuando la ubicación de sus ceros coincida. Si se colocan varios pre-filtro coseno en cascada, es posible incrementar la atenuación en todas las bandas de rechazo, por lo tanto:

$$H_{CCos}(z) = \prod_{i=1}^K H_{Cos}(z^{N_i}). \quad (2.13)$$

donde el valor de cada N está dado por:

$$N_i = \frac{M}{2^{i+1}}. \quad (2.14)$$

La estructura eficiente para la implementación en cascada de (2.13) y (1.1), se obtiene considerando que M puede ser expresado como el producto de N números enteros:

$$M = M_1 M_2 M_3 \cdots M_N. \quad (2.15)$$

De esta forma la función (1.1) se reescribe como:

$$H_m(z) = \left[\prod_{i=1}^N H_i \left(z^{\prod_{j=0}^{i-1} M_j} \right) \right]^K. \quad (2.16)$$

En forma general, la función de transferencia de un filtro Comb-Coseno es:

$$H_{mCCos}(z) = H_m(z) \prod_{i=1}^K H_{CCos}(z^{N_i}). \quad (2.17)$$

Ejemplo 2.2. Los parámetros de diseño de un filtro comb son $M = 32$ y $K = 1$. Se considera dividir M en tres sub-factores, donde $M_1 = 2, M_2 = 4$ y $M_3 = 4$. Usando (2.14), se obtienen tres valores para N , que son $N_1 = 8, N_2 = 4$ y $N_3 = 2$. En la Fig. 2.3 se muestran las respuestas en magnitud de los filtros $H(z), H_{ccos}(z)$ y $H_{mccos}(z)$, diseñados con los parámetros previamente mencionados.

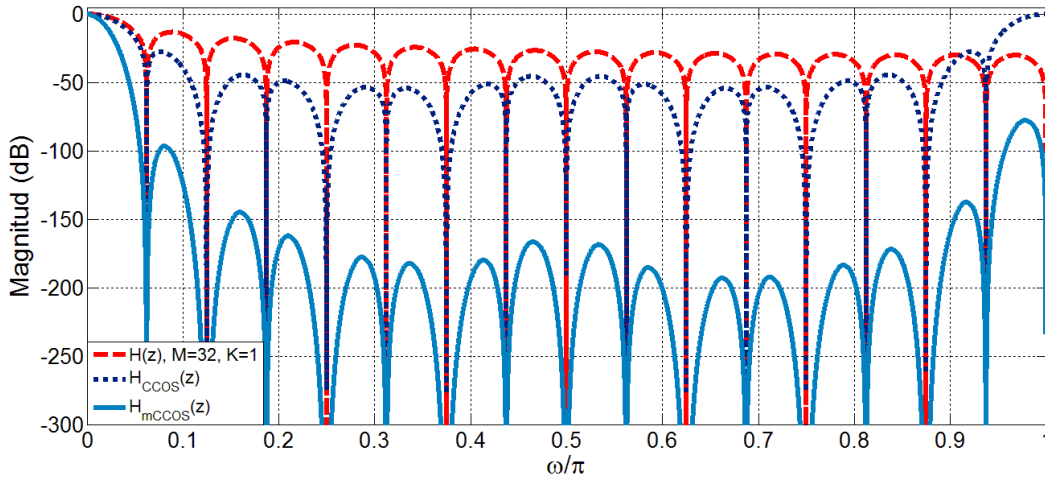


Fig. 2.3. Respuesta en magnitud del Ejemplo 2.2.

Se observa que la respuesta en magnitud del filtro Comb-Coseno presenta una notable mejoría en atenuación. Sin embargo, tiene como desventaja que la caída en la banda de paso se incrementa considerablemente. La solución a este inconveniente se obtiene aplicando filtros compensadores. Algunas de estas propuestas se han presentado en [14]-[17].

2.2.2.2. El Filtro Sinc Rotado (RS) [18]

La función de transferencia del *Filtro Sinc Rotado* o *Filtro RS* se obtiene aplicando la técnica de rotación de ceros. En esta los ceros del filtro comb recursivo son rotados α radianes, tanto en sentido de las manecillas del reloj, como en sentido contrario. Deduciendo así las siguientes dos funciones:

$$H_u(z) = \frac{1}{M} \frac{1 - z^{-M} e^{j\alpha M}}{1 - z^{-1} e^{j\alpha}} \quad (2.18)$$

$$H_d(z) = \frac{1}{M} \frac{1 - z^{-M} e^{-j\alpha M}}{1 - z^{-1} e^{-j\alpha}} \quad (2.19)$$

Una conveniente elección de α es:

$$\alpha = q2\pi\omega_p, \quad (2.20)$$

donde $q \in [-1, 1]$ y su valor se elige de acuerdo a la selectividad deseada en las bandas de rechazo.

Los filtros (2.18) y (2.19) tienen coeficientes complejos. Sin embargo, si estos se conectan en cascada el resultado es un filtro con coeficientes reales:

$$H_r(z) = H_u(z)H_d(z) = \frac{1}{M^2} \frac{1 - 2 \cos(\alpha M) z^{-M} + z^{-2M}}{1 - 2 \cos \alpha z^{-1} + z^{-2}}. \quad (2.21)$$

Al colocar un filtro $H(z)$ seguido por un filtro $H_r(z)$, se obtiene la función de transferencia del filtro RS:

$$H_{RS}(z) = H(z)H_r(z), \quad (2.22)$$

donde $H(z)$ y $H_r(z)$ están dadas por (1.1) y (2.21) respectivamente.

La respuesta en magnitud del filtro (2.22) es:

$$H_{RS}(e^{j\omega}) = \left| \frac{1}{M^{3K}} \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right|^K \left| \frac{\sin\left[\frac{(\omega + \alpha)M}{2}\right]}{\sin\left[\frac{(\omega + \alpha)}{2}\right]} \right|^K \left| \frac{\sin\left[\frac{(\omega - \alpha)M}{2}\right]}{\sin\left[\frac{(\omega - \alpha)}{2}\right]} \right|^K. \quad (2.23)$$

Ejemplo 2.3. Aplicando la técnica de rotación de ceros, se diseña un filtro RS con $M = 16$, $K = 1$ y $\alpha = 0.1841$. En la Fig. 2.4 se muestra la respuesta en magnitud del filtro RS, y se realiza una comparación con la respuesta de un filtro comb con $M = 16$ y $K = 1$.

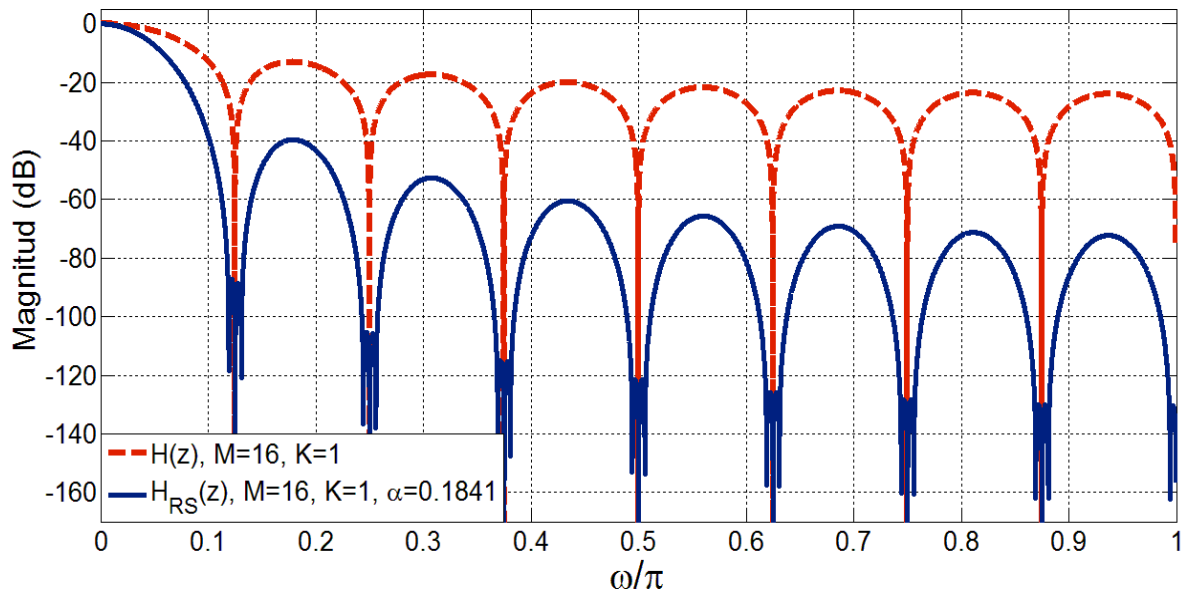


Fig. 2.4. Respuestas en magnitud del Ejemplo 2.3.

De la Fig. 2.4 y de la función de transferencia (2.22) se concluye lo siguiente:

- La respuesta en magnitud de un filtro RS se caracteriza, no sólo por tener una mejor atenuación en las bandas de rechazo, sino también porque estas son más anchas. Sin embargo, presenta un incremento en la caída de la banda de paso.
- La implementación de un filtro RS requiere de dos multiplicadores, uno de ellos trabajando a la frecuencia de muestreo de entrada. Esto provoca un incremento en el consumo de potencia.
- Al utilizar precisión finita en la representación de los coeficientes del filtro (2.21), puede existir inestabilidad en el filtro. Esto a consecuencia de la no cancelación perfecta de los polos y ceros.

Algunas propuestas para solucionar los inconvenientes del filtro RS se han presentado en [19]-[21].

2.2.2.3. El Filtro Comb Generalizado (GCF) [22]

El Filtro Comb Generalizado o GCF (por sus siglas en inglés) es una generalización del Filtro RS. Su método de diseño está enfocado en obtener decimadores aplicables en ADCs- $\Sigma\Delta$. Este método resuelve dos de las problemáticas presentes en [18]:

- Proporciona los valores óptimos de α .
- Elimina la necesidad de utilizar multiplicadores.

La función de transferencia de un GCF de orden K está dada por:

$$H_{GCF_K}(z) = \prod_{n=1}^K \frac{\sin(\alpha_n/2)}{\sin(\alpha_n M/2)} \prod_{n=1}^K \frac{1 - z^{-M} e^{-j\alpha_n M}}{1 - z^{-1} e^{-j\alpha_n}}. \quad (2.24)$$

Su correspondiente respuesta en magnitud es:

$$H_{GCF_K}(\omega) = \prod_{n=1}^K \frac{\sin(\alpha_n/2)}{\sin(\alpha_n M/2)} \prod_{n=1}^K \frac{\sin((\omega + \alpha_n)M/2)}{\sin((\omega - \alpha_n)/2)}, \quad (2.25)$$

donde α_n , con $n = 1, \dots, K$, son los parámetros de rotación. Cada uno de ellos está definido por (2.20). Los valores óptimos de q_n son aquellos que reducen la potencia del ruido de cuantización, introducido por el ADC- $\Sigma\Delta$, en las bandas de rechazo. Estos son calculados a

través de un análisis presentado en [22]. La Tabla 2.4 contiene los resultados obtenidos para diferentes órdenes de CGF.

K	3	4	5	6
q_1	-0.79	-0.35	+0.55	+0.95
q_2	0.0	+0.35	+0.93	+0.675
q_3	+0.79	-0.88	-0.55	+0.25
q_4	+0.79	+0.88	-0.93	-0.25
q_5	-	+0.88	0.0	-0.675
q_6	-	+0.35	+0.55	-0.95
q_7	-	-	+0.93	+0.675
q_8	-	-	-	+0.25
q_9	-	-	-	-

Tabla 2.4. Valores óptimos de q .

La estructura libre de multiplicadores de un GCF se obtiene aproximando los valores de α a través de potencias de dos. Otra alternativa para evitar el uso de multiplicadores en un GCF se ha propuesto en [23]. También se han considerado diseños de bajo consumo de potencia como el presentado en [24].

Ejemplo 2.4. Considérese el diseño de un GCF con $M = 16$ y $K = 3$. De la Tabla 2.4 se obtienen los valores óptimos $q_1 = -0.79$, $q_2 = 0$ y $q_3 = 0.79$. En la Fig. 2.5 se muestra la respuesta en magnitud del filtro diseñado. También, para fines de comparación se incluye la respuesta de un filtro comb con $M = 16$ y $K = 3$.

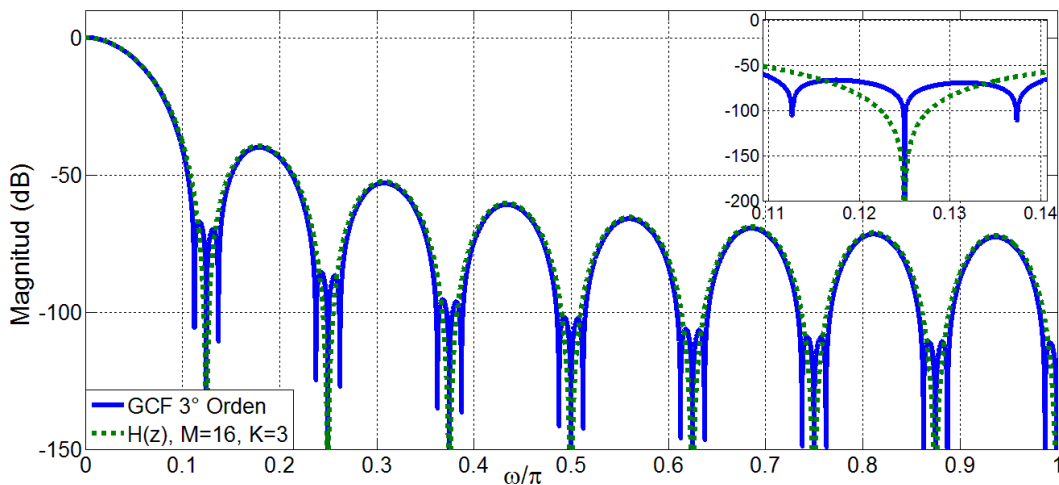


Fig. 2.5. Respuestas en magnitud del Ejemplo 2.4.

La respuesta en magnitud del GCF se caracteriza por tener mayor anchura y atenuación en las bandas de rechazo. Sin embargo, su respuesta en la banda de paso sigue siendo igual a la de un filtro comb convencional. Algunos métodos para solucionar este inconveniente se han presentado en [25]-[27].

2.2.3. Mejora en la Respuesta en Magnitud de Ambas Bandas

2.2.3.1. Aplicación de la Técnica de Sharpening al Filtro Comb [28]-[37]

El método de Sharpening, presentado por Kaiser y Hamming en [28], establece que la respuesta en magnitud de un filtro FIR se mejora si este es usado repetidamente. La forma en que se realizan las interconexiones de los filtros, está dictada por la llamada función de cambio de amplitud (ACF). En [28] Kaiser y Hamming definen una familia de ACFs. Estas están dadas por:

$$H_{out} = H^{n+1} \sum_{k=0}^m \frac{(n+k)!}{n!k!} [1-H]^k = H^{n+1} \sum_{k=0}^m C(n+k, k)H^k, \quad (2.26)$$

donde m y n son números enteros positivos, H la función de transferencia del filtro FIR y $C(n+k, k)$ los coeficientes del polinomio de la ACF.

En la Tabla 2.5 se presentan las ACF considerando que $m = n$ y definiendo $\bar{H} = 1 - H$.

$m = n$	H_{out}	Forma alternativa
0	H	H
1	$H^2[1 - 2\bar{H}]$	$H^2[3 - 2H]$
2	$H^3[1 + 3\bar{H} + 6\bar{H}^2]$	$H^3(z)[10 - 15H + 6H^2]$
3	$H^4[1 + 4\bar{H} + 10\bar{H}^2 + 20\bar{H}^3]$	$H^4[35 - 84H + 70H^2 - 20H^3]$

Tabla 2.5. Funciones de cambio de amplitud.

El valor de n está asociado con la mejora que habrá en la banda de paso del filtro FIR, y el de m con la mejora de la banda de rechazo. Sin embargo, un valor grande en n o m , implica el uso de un polinomio de orden mayor. Esto tiene como consecuencia un incremento de recursos en la implementación del filtro deseado.

La aplicación del método de Sharpening en el diseño de filtros comb fue presentado por primera vez en [29], y proporciona filtros con mejoras tanto en la banda de paso como en las bandas de rechazo. Con la finalidad de mantener una baja complejidad, comúnmente se utiliza la ACF con $m = n = 1$. Por lo tanto, sustituyendo $H(z)$ en (2.26), la función de transferencia de un filtro Sharpened-Comb es:

$$H_{Sh}(z) = 3H^2(z) - 2H^3(z). \quad (2.27)$$

Su correspondiente respuesta en magnitud es:

$$|H_{Sh}(e^{j\omega})| = \left| 2 \left(\frac{1 \sin(\omega M/2)}{M \sin(\omega/2)} \right)^{2K} - 2 \left(\frac{1 \sin(\omega M/2)}{M \sin(\omega/2)} \right)^{3K} \right|. \quad (2.28)$$

Ejemplo 2.5. Considere un filtro comb cuyos parámetros de diseño son $M = 16$ y $K = 3$. Su respuesta en magnitud es comparada con la de un filtro Sharpened-Comb con $M = 16$ y $K_1 = 2$. En la Fig. 2.6 se muestra dicha comparación.

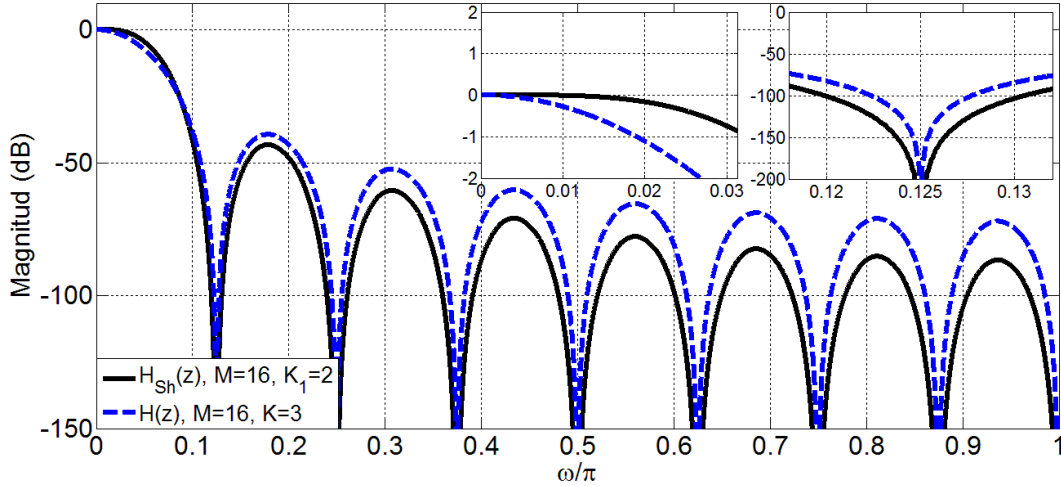


Fig. 2.6. Respuestas en magnitud del Ejemplo 2.5.

El concepto de Sharpening ha sido aplicado en el desarrollo de varias estructuras decimadoras basadas en filtros comb. En [30] se considera que M puede ser expresado como:

$$M = 2M_1. \tag{2.29}$$

De esta forma, la función de transferencia de un filtro comb se reescribe de la siguiente forma:

$$H(z) = [H_1(z^2)H_2(z)]^K, \tag{2.30}$$

donde:

$$H_1(z^2) = \left[\frac{1}{M_1} \frac{1 - z^{-2M_1}}{1 - z^{-2}} \right]^K; \quad H_2(z) = \left[\frac{1}{2} (1 + z^{-1}) \right]^K. \tag{2.31}$$

Con el objetivo de reducir el consumo de potencia, la técnica de Sharpening sólo es aplicada al filtro $H_1(z^2)$. Teniendo como resultado la siguiente función de transferencia:

$$H_n(z) = H_2(z)Sh\{H_1(z^2)\}, \tag{2.32}$$

donde

$$Sh\{H_1(z^2)\} = 3[H_1(z^2)]^{2K} - 2[H_1(z^2)]^{3K}. \tag{2.33}$$

Otras variantes del filtro (2.32) son presentadas en [31]-[32]. En [31] se hace una generalización de la estructura, considerando que M puede ser expresado como el producto

de dos números enteros $M = M_1 M_2$. Cada sub-etapa es implementada en forma recursiva y tienen diferentes números de filtros comb en cascada. En [32] se introduce la idea de implementar la primera sub-etapa en forma no recursiva. Esto permite incrementar la velocidad de procesamiento.

La técnica de Sharpening generalizada [32] es aplicada a una estructura de dos etapas en [34] y a una estructura completamente no recursiva en [35]. Finalmente, también se han considerado el desarrollo de métodos que permitan encontrar los valores óptimos de los coeficientes de la ACF. Algunos de estos se introducen en [36]-[37]. A continuación se describe brevemente en que consiste el método presentado en [37].

2.2.3.2. Método de Optimización de la Función de Cambio de Amplitud [37]

El objetivo principal de este método, es reducir el valor de la integral definida por el criterio del mínimo cuadrado ponderado, a partir de encontrar los valores óptimos de los coeficientes de la ACF, es decir.

$$\varepsilon_{wls}(\mathbf{a}) = \int_B w(s) [H_{SCIC}(\omega, \mathbf{a}) - H_d(\omega)]^2 d\omega, \quad (2.34)$$

donde:

La respuesta en magnitud del filtro H_{SCIC} es:

$$H_{SCIC}(\omega) = \sum_{n=0}^N a_n H_{CIC}^n(\omega). \quad (2.35)$$

donde:

N : Determina el orden de la ACF.

B : Corresponde a banda de frecuencias a mejorar, ya sea de paso o de rechazo.

$w(s)$: Es una función positiva de ponderación.

\mathbf{a} : Es el vector de coeficientes óptimos de la ACF.

$$\mathbf{a} = [a_1 \ a_2 \ \dots \ a_N]^T, \quad (2.36)$$

$H_d(\omega)$: Se define como la respuesta en frecuencia deseada para el filtro SCIC.

$$H_d(\omega) = \begin{cases} 1; & \omega \leq \omega_p \\ 0; & \frac{2k\pi}{M} - \omega_p \leq \omega \leq \frac{2k\pi}{M} + \omega_p, \ n = 1, \dots, \left\lfloor \frac{M}{2} \right\rfloor. \end{cases} \quad (2.37)$$

La solución del problema anteriormente planteado, está dada por:

$$\hat{\mathbf{a}} = \arg \min_{\mathbf{a}} \varepsilon_{wls}(\mathbf{a}). \quad (2.38)$$

El procedimiento para resolver la ecuación (2.38) consta de varios pasos. El primero de ellos consiste en dividir en ancho de banda B en I puntos de frecuencias distribuidos uniformemente.

$$\{\omega_i: i = 0, 1, \dots, I\}. \quad (2.39)$$

De esta forma, la integral (2.35) es aproximada a través de una suma finita de términos:

$$\varepsilon(\mathbf{a}) = \sum_{i=0}^{I-1} w(\omega_i) [H_{SCIC}(\omega_i, \mathbf{a}) - H_d(\omega_i)]^2. \quad (2.40)$$

Finalmente, la ecuación (2.40) puede ser expresada en forma matricial:

$$\varepsilon(\mathbf{a}) = \|\mathbf{W}(\mathbf{H}\mathbf{a} - \mathbf{d})\|^2, \quad (2.41)$$

donde cada una de las matrices se encuentra definida por:

$$\mathbf{W} = \begin{bmatrix} \omega_0 & 0 & \dots & 0 \\ 0 & \omega_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \omega_{I-1} \end{bmatrix}, \quad (2.42)$$

$$\mathbf{H} = \begin{bmatrix} H_{CIC}(\omega_0) & H_{CIC}^2(\omega_0) & \dots & H_{CIC}^K(\omega_0) \\ H_{CIC}(\omega_1) & H_{CIC}^2(\omega_1) & \dots & H_{CIC}^K(\omega_1) \\ \vdots & \vdots & \ddots & \vdots \\ H_{CIC}(\omega_{I-1}) & H_{CIC}^2(\omega_{I-1}) & 0 & H_{CIC}^K(\omega_{I-1}) \end{bmatrix}, \quad (2.43)$$

$$\mathbf{d} = [H_d(\omega_0) \quad H_d(\omega_1) \quad \dots \quad H_d(\omega_{I-1})]^T. \quad (2.44)$$

Aplicando propiedades de operaciones con matrices, y derivando la función $\varepsilon(\mathbf{a})$ con respecto a \mathbf{a} , la expresión (2.41) puede ser reducida a:

$$\hat{\mathbf{a}} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} (\mathbf{H}^T \mathbf{W} \mathbf{d}). \quad (2.45)$$

Realizando los cálculos indicados por (2.45), es posible determinar el valor óptimo de los coeficientes de la ACF. A continuación se presentan ejemplos de diseños de filtros (2.35) a partir de la aplicación del método anteriormente descrito.

Ejemplo 2.6. Diseño de un filtro de banda angosta: Considérese en el diseño del filtro $H_{SCIC}(\omega)$ la función de ponderación dada por:

$$w(\omega) = \begin{cases} 1; & \omega \leq \omega_p \\ 5; & \frac{\pi}{5} - \omega_p \leq \omega \leq \frac{\pi}{5} + \omega_p \\ 0.1; & \frac{k\pi}{5} - \omega_p \leq \omega \leq \frac{k\pi}{5} + \omega_p, \quad k = 2, 3 \text{ y } 5 \\ 0.1; & \pi - \omega_p \leq \omega < \pi \end{cases}, \quad (2.46)$$

donde se selecciona $\omega_p = 0.2\pi$. Se tiene un filtro CIC de orden $K = 2$ con un factor $M = 10$, el orden del polinomio de Sharpening es $N = 3$ y se considera una división $I = 2000$ puntos. Realizando las operaciones indicadas por (2.45) se obtiene como vector de coeficientes:

$$\hat{\mathbf{a}} = \begin{bmatrix} -0.024199 \\ 3.134081 \\ -2.110156 \end{bmatrix}. \quad (2.47)$$

En la Fig. 2.7 se muestra la respuesta en magnitud del Filtro $H_{SCIC}(\omega)$ en comparación con la respuesta de un filtro CIC de orden $K = 2$ y $M = 10$. Se observa claramente las mejoras que proporciona la aplicación del método, tanto en la banda de paso como en las bandas de rechazo.

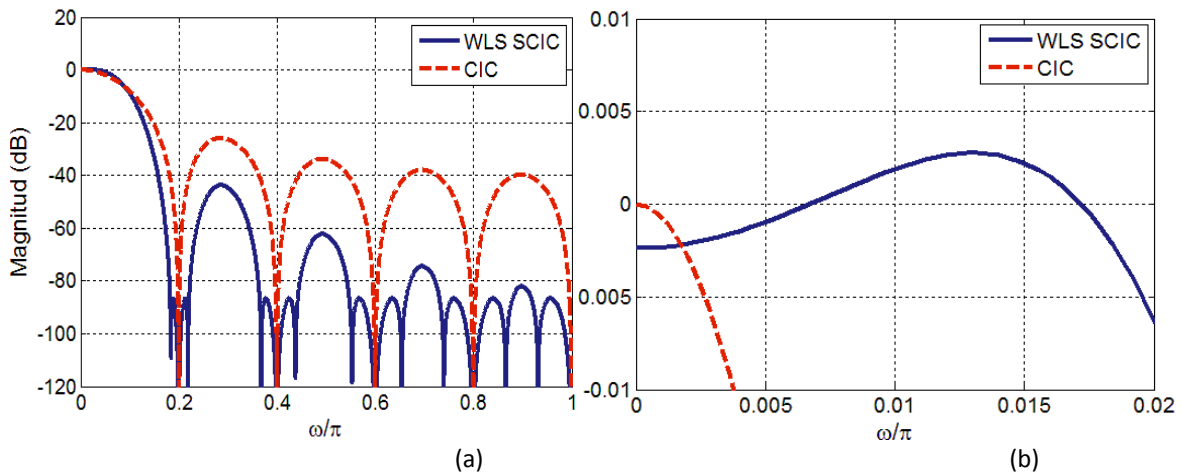


Fig. 2.7. Respuestas en magnitud del Ejemplo 2.6. (a) Toda la banda de frecuencias. (b) Banda de paso.

Ejemplo 2.7. Diseño de un filtro de banda ancha: En este ejemplo se considera la función de ponderación (2.46) y los mismos parámetros de diseño del filtro CIC. Se selecciona $\omega_p = 0.05\pi$ y el orden del polinomio de Sharpening es de $N = 7$, obteniendo así el siguiente vector de coeficientes.

$$\hat{\mathbf{a}} = \begin{bmatrix} 0.001694 \\ -0.342322 \\ 12.10196 \\ -18.21024 \\ 7.448995 \end{bmatrix}. \quad (2.48)$$

En la Fig. 2.8 se presenta la respuesta en magnitud del filtro $H_{SCIC}(\omega)$ en comparación con un filtro CIC de orden $K = 2$ y factor $M = 10$. Se observa claramente la respuesta del filtro $H_{SCIC}(\omega)$ tiene mejores características que las del filtro CIC, lo que lo hace más selectivo.

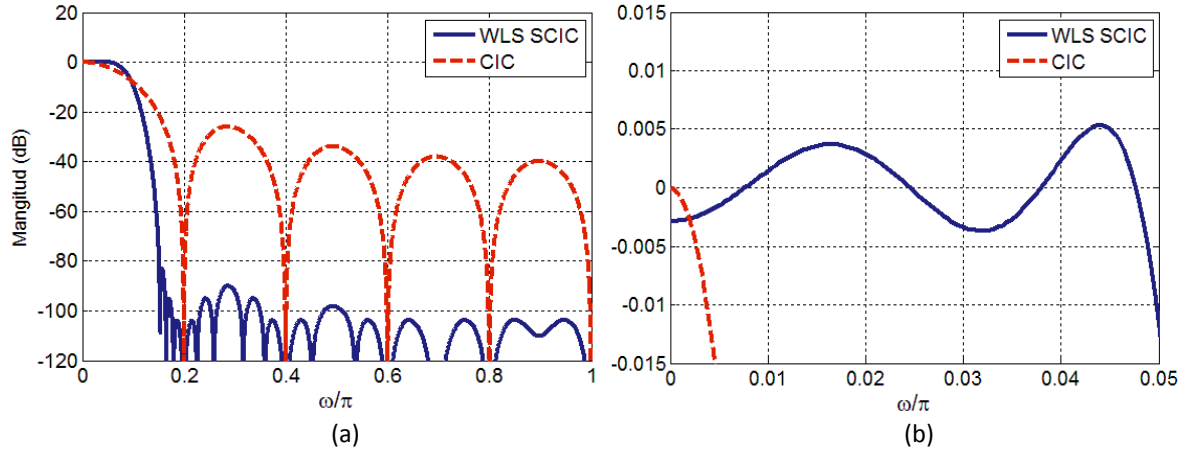


Fig. 2.8. Respuesta en magnitud del Ejemplo 2.7. (a) Toda la banda de frecuencias. (b) Banda de paso.

2.2.3.3. Aplicación de un Filtro IIR para Mejorar la Respuesta en Magnitud [38]

Con el objetivo de mejorar la respuesta en magnitud del filtro comb, se propone el uso de un filtro IIR de segundo orden. Al aplicarlo, se produce una mejora tanto en la banda de paso como en las bandas de rechazo. La función de transferencia del filtro IIR está dada por:

$$P(z) = A \frac{(1 - c_1 z^{-1})(1 - c_2 z^{-1})}{(1 - d_1 z^{-1})(1 - d_2 z^{-1})} \quad (2.49)$$

donde:

c_1, c_2 : Par de ceros complejos conjugados. $c_{1,2} = e^{\mp j\omega_0}$.

d_1, d_2 : Par de polos complejos conjugados. $d_{1,2} = e^{\mp j\omega_1}$.

A : Constante de normalización.

Sustituyendo las expresiones de los polos y ceros complejo conjugados en la ecuación (2.49). Esta puede ser reducida a:

$$P(z) = A \frac{1 - 2 \cos \omega_0 z^{-1} + z^{-2}}{1 - 2r \cos \omega_1 z^{-1} + r^2 z^{-2}} \quad (2.50)$$

Posteriormente, si se conecta en cascada el filtro IIR (2.50) con un filtro CIC. La función de transferencia de la estructura resultante así como su respuesta en magnitud están dadas por:

$$H_{CIC-IIR}(z) = P(z)H(z), \quad (2.51)$$

$$H_{CIC-IIR}(e^{j\omega}) = \left(\frac{\sin(M\omega/2)}{M \sin(\omega/2)} \right)^K \times \frac{A(2 \cos \omega - 2 \cos \omega_0) e^{j(1-D)\omega/2}}{(1+r^2) \cos \omega_0 - 2r \cos \omega_1 + j(1-r^2) \sin \omega_0}. \quad (2.52)$$

En forma general, la respuesta en fase de (2.51) no es lineal, ya que se trata de un filtro IIR. Sin embargo, para valores pequeños de ω_p se puede considerar que dentro del intervalo $[0, \omega_p]$ la respuesta en fase puede ser aproximadamente:

$$\phi(\omega) \approx \frac{(1-M)\omega}{2}. \quad (2.53)$$

De la expresión (2.52) se puede ver que se requiere calcular los valores óptimos de ω_0 , ω_1 y r para mejorar la respuesta del filtro CIC. El valor de r tiene que ser menor a uno para garantizar la estabilidad del filtro IIR, haciendo que todos sus polos se encuentren dentro del círculo unitario. Los valores de ω_0 y ω_1 se obtiene definiendo una función de error dentro de la banda de paso como:

$$\varepsilon_p(\omega) = |G(\omega)| - 1 = |H_{CIC-IIR}(\omega)| - 1. \quad (2.54)$$

Adicionalmente, la función de error dentro de la banda de rechazo está dada por:

$$\varepsilon_s(\omega) = |G(\omega)|. \quad (2.55)$$

Por lo tanto, considerando las expresiones (2.54) y (2.55), la función de error total de la respuesta en magnitud del filtro $H_{CIC-IIR}(z)$ es:

$$\epsilon = \alpha \int_0^{\omega_p} \varepsilon_p^2(\omega) d\omega + (1-\alpha) \int_{\omega_p}^{\pi} \varepsilon_s^2(\omega) d\omega, \quad (2.56)$$

donde α es un parámetro de peso y $\omega_s = \frac{3\pi}{2M}$.

Finalmente, considerado lo anterior, el problema de optimización se resuelve aplicando el siguiente criterio de optimización:

$$\min_{\omega_0, \omega_1, r} \epsilon, \quad (2.57)$$

Ejemplo 2.8. Considérese un filtro CIC de orden $K = 3$ y factor de decimación $M = 16$. La frecuencia de corte es igual a $0.85\pi/M$, y se tiene como parámetro de peso $\alpha = 0.6$. Resolviendo (2.57) con los parámetros dados, se obtienen como valores óptimos para el diseño del filtro IIR: $\omega_0 = 0.49087385$, $\omega_1 = 0.17220228$ y $r = 0.96793194$.

En la Fig. 2.9 se muestra la respuesta en magnitud del filtro $H_{CIC-IIR}(z)$ en comparación con la respuesta de un filtro CIC de tercer orden. Se observa claramente que existe una buena

compensación en la banda de paso y un incremento en la atenuación de las bandas de rechazo.

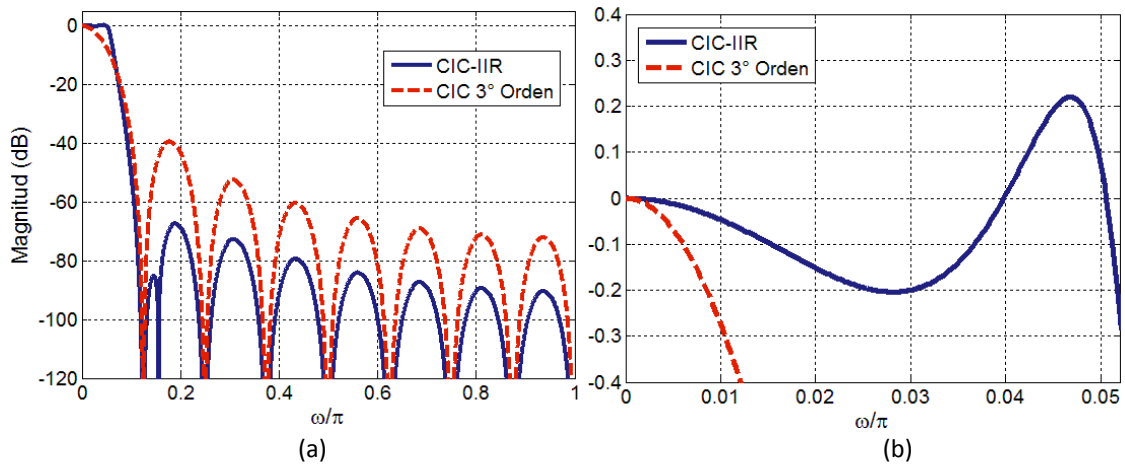


Fig. 2.9. Respuestas en magnitud del Ejemplo 2.8. (a) Toda la banda de frecuencias. (b) Banda de paso.

2.3. Reducción en el Consumo de Potencia y Hardware

Hasta el momento se han revisado algunos de los métodos utilizados para mejorar la respuesta en magnitud del filtro comb. Sin embargo, también existen métodos enfocados a reducir el consumo de potencia y de hardware requerido en su implementación. A continuación se describen algunas de las ideas introducidas para conseguir este objetivo.

2.3.1. Multiplicadores a Base de Sumas y Corrimientos [39]-[41]

La implementación de los coeficientes de un filtro digital requiere de multiplicadores. Estos son los responsables del mayor consumo de potencia y área de chip [39]. Por lo tanto, se han desarrollado varios algoritmos para realizar la operación de multiplicación en forma eficiente. El algoritmo de Höner [40] establece que si la base usada para la realización de las operaciones es 2 (números binarios), la operación de multiplicación puede hacerse utilizando sólo sumas y corrimientos. Los pasos de este algoritmo son los siguientes:

- Encontrar la representación del coeficiente como suma de potencias de dos, y representarlo como un número binario.
- Cada elemento 2^n , presente en la representación binaria del coeficiente, aplicará un corrimiento de n -bits a la izquierda sobre el multiplicando. En caso de tener un elemento 2^{-n} , el corrimiento se llevará a cabo hacia la derecha.
- Finalmente, se deben de sumar todos los corrimientos obtenidos en el paso anterior. Las sumas requeridas en este paso serán $b - 1$, donde b es el número de bits distintos de cero de la representación binaria del coeficiente.

Ejemplo 2.9. Considérese que se tiene un dato x de entrada. Este requiere ser multiplicado por un coeficiente con valor 22. La forma eficiente de implementar esta multiplicación se obtiene partiendo de la representación a base de potencias de dos del coeficiente. Es decir:

$$22 = 2^4 + 2^2 + 2^1. \quad (2.58)$$

Posteriormente, el dato x es multiplicado por cada elemento que conforma la representación en suma de potencias de dos del coeficiente.

$$22x = 2^4x + 2^2x + 2^1x. \quad (2.59)$$

Los términos 2^4 , 2^2 y 2^1 indican que se aplicarán corrimientos de 4, 2 y 1 bit al dato x de entrada. Finalmente, se suman todos los corrimientos. Efectuar la operación (2.59) necesita la implementación de dos sumadores.

2.3.2. Representación de Coeficientes como Dígitos con Signo Canónico (CSD) [42]

En el apartado anterior se mostró que es posible implementar una multiplicación por medio de sumas y corrimientos. Los sumadores requeridos en esta operación, depende del número de términos en potencias de dos en los que se puede representar el coeficiente. Por lo tanto, para reducir el consumo de hardware, se han desarrollado formas eficientes de representar dichos coeficientes y reducir así el número de bits requeridos.

Una de estas formas es la representación en formato *CSD* (*Canonical Signed Digit*). El formato *CSD* está conformado por un el conjunto de dígitos $\{1, 0, \bar{1}\}$, donde $\bar{1} = -1$, y tiene las siguientes propiedades:

- El número de bits diferentes de cero es mínimo. Para un número representado por n bits en formato *CSD* existen como máximo $\lfloor n/2 \rfloor$ bits diferentes de cero.
- El producto de dos bits adyacentes es cero. Es decir $b_n * b_{n-1} = 0$.

Los pasos para convertir un coeficiente al formato *CSD* son:

1. Obtener la representación binaria en complemento a dos con signo del coeficiente.
2. Conocer el valor del bit ubicado en la posición más significativa de la cadena de bits (MSB). Si el bit MSB es -1 se cambia por $\bar{1}$. En caso contrario no se realiza ningún cambio.
3. Analizar las parejas de bits adyacentes de derecha a izquierda y realizar la conversión utilizando las siguientes transformaciones:

$$\begin{array}{cccccccccc}
 \dots & 0 & 0 & 1 & 1 & 1 & \dots & 1 & 1 & 0 & \dots \\
 & & i+k & & & & & & i & & \\
 & & & & & \downarrow & & & & & \\
 \dots & 1 & 0 & 0 & 0 & 0 & \dots & & \bar{1} & 0 & \dots \\
 & & i+k & & & & & & i & &
 \end{array} \tag{2.60a}$$

$$\begin{array}{cccccccccc}
 \dots & \bar{1} & 1 & 1 & \dots & 1 & 1 & 0 & \dots \\
 & & i+k & & & & & & i & & \\
 & & & & & \downarrow & & & & & \\
 \dots & 0 & 0 & 0 & \dots & 0 & \bar{1} & 0 & \dots \\
 & & i+k & & & & & & i & &
 \end{array} \tag{2.60b}$$

Ejemplo 2.10. Se desea obtener la representación en formato CSD de los coeficientes 14 y -81. Por lo tanto, primero es necesario obtener su representación como números binarios en complemento a dos:

$$14|_{10} = 01110|_b \tag{2.61}$$

$$-81|_{10} = 10101111|_b \tag{2.62}$$

Posteriormente, aplicando el paso dos, los números (2.61) y (2.62) quedan expresados como:

$$01110 \tag{2.63}$$

$$\bar{1}0101111 \tag{2.64}$$

Finalmente, usando las propiedades del paso tres, la representación en CDS de los coeficientes es:

$$100\bar{1}0|_{CDS} \tag{2.65}$$

$$0\bar{1}0\bar{1}000\bar{1}|_{CDS} \tag{2.66}$$

Considerando que se tiene un dato de entrada x , la multiplicación de x a base de sumas y corrimientos con los coeficientes (2.65) y (2.66) se implementa como:

$$14x = 2^4 - 2^1 \tag{2.67}$$

$$-81x = -2^6 - 2^4 - 2^0 \tag{2.68}$$

Finalmente, se observa que tanto en el formato CSD como de la implementación de multiplicadores mediante sumas y corrimientos, se requiere que los coeficientes sean constantes y enteros. Sin embargo, los métodos de diseño proporcionan coeficientes fraccionarios. Por lo tanto, se requiere usar un factor de redondeo apropiado para volverlos enteros los coeficientes.

Capítulo 3

Método Propuesto

En este capítulo se propone una estructura de decimación no recursiva basada en filtros comb. Esta posee mejores características en magnitud que un filtro comb convencional. En su diseño se considera que el factor de decimación tiene que ser expresado como potencias de dos. Se introduce un filtro coseno rotado para incrementar la atenuación en las bandas de rechazo. Adicionalmente, la caída en la banda de paso es disminuida usando un filtro compensador que trabaja en baja frecuencia. A su vez, el diseño libre de multiplicadores de la estructura propuesta es considerado. Finalmente, su funcionamiento es ilustrado por medio de ejemplos y se realiza una comparación con otros métodos existentes.

3.1. Filtro Coseno Expandido

La función de transferencia del filtro coseno expandido se escribe como:

$$H_{\text{COS}}(z) = 0.5(1 + z^{-N}), \quad (3.1)$$

donde N es un número entero positivo. Se conoce así debido a que su respuesta en magnitud tiene la forma de la función coseno. Esta se encuentra dada por la siguiente expresión:

$$|H_{\text{COS}}(e^{j\omega N})| = |\cos(\omega N/2)|. \quad (3.2)$$

En la Fig. 3.1 se muestra la respuesta en magnitud del filtro coseno expandido para diferentes valores de N . Obsérvese que los ceros de este filtro se encuentran localizados en las frecuencias:

$$\frac{(2l-1)\pi}{N}; \quad \text{para } l = \begin{cases} 1, \dots, \frac{N}{2} & N \text{ par} \\ 1, \dots, \frac{N+1}{2} & N \text{ impar} \end{cases} \quad (3.3)$$

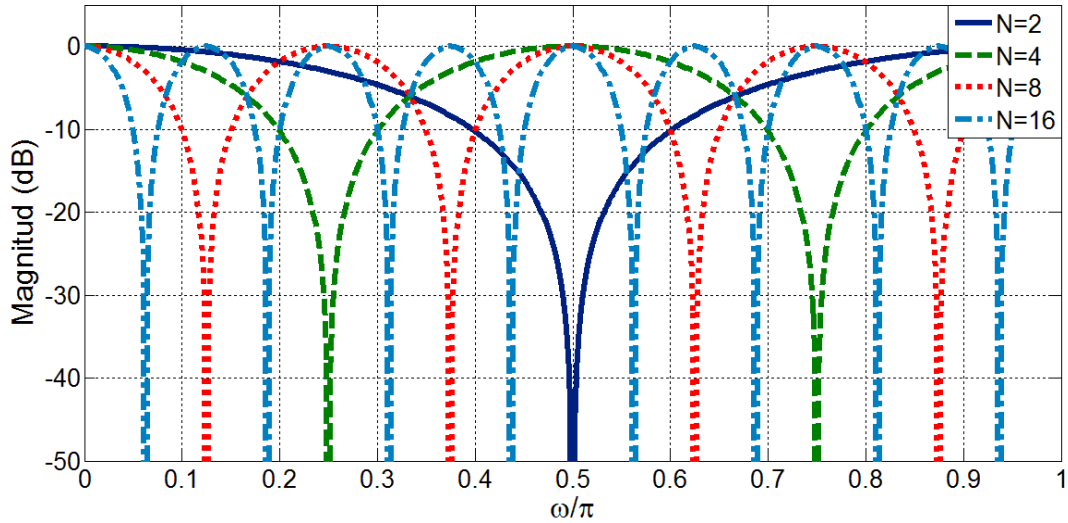


Fig. 3.1. Respuesta en magnitud del filtro coseno expandido para diferentes valores de N .

Por consiguiente, el primer cero del filtro coseno expandido ($l = 1$) está ubicado en π/N y los restantes, en los múltiplos enteros de dicha frecuencia. Por otro lado, el filtro comb tiene sus ceros localizados en los múltiplos enteros de $2\pi/M$. Entonces, si un filtro comb es seguido por un filtro coseno expandido, es posible incrementar la atenuación en ciertas bandas de rechazo si se hace coincidir la ubicación de sus ceros. Por ejemplo, considérese una estructura formada por un filtro comb conectado en cascada con un filtro coseno con $N = M/2$, la respuesta en magnitud de esta estructura presenta mayor atenuación en todas las bandas de rechazo impares. De forma similar, usando $N = M/4$, ceros adicionales se colocan en las bandas 2, 6, 10 y así sucesivamente.

En forma general, usando:

$$N = \frac{M}{2^k}, \tag{3.4}$$

donde k es un número entero positivo con valor $k = 1, 2, \dots, (\log_2 M) - 1$ y M es expresable en potencias de dos. La atenuación de las siguientes bandas de rechazo se incrementa:

$$2^{k-1}(2l - 1), \quad \text{para } l = 1, 2, \dots, \frac{M}{2^{k+1}}. \tag{3.5}$$

Esto se ilustra a continuación a través de un ejemplo.

Ejemplo 3.1. Se consideran los factores de decimación $M = 8, 16, 32$ y 64 . De acuerdo a la expresión (3.5) y usando (3.4), las bandas de rechazo mejoradas para $k = 1, 2$ y 3 son enlistadas en la Tabla 3.1.

M	k	N	Bandas de Rechazo Mejoradas
8	1	4	1,3
	2	2	2
	3	-	-
16	1	8	1, 3, 5, 7
	2	4	2, 6
	3	2	4
32	1	16	1, 3, 5, 7, 9, 11, 13, 15
	2	8	2, 6, 10, 14
	3	4	4, 12
64	1	32	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31
	2	16	2, 6, 10, 14, 18, 22, 26, 30
	3	8	4, 12, 20, 28

Tabla 3.1. Bandas de rechazo mejoradas.

En la Fig. 3.2 se muestran las respuestas de magnitud de un filtro comb ($K = 3$), y de los correspondientes filtros coseno, tomando los valores de la Tabla 3.1 para $M = 32$.

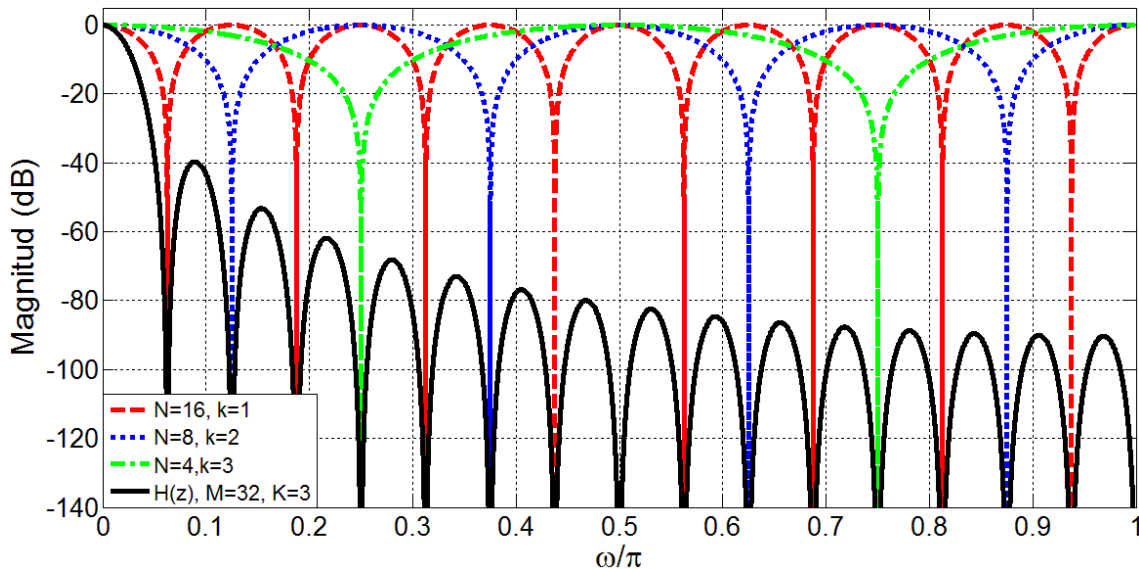


Fig. 3.2. Respuestas en magnitud de un filtro comb ($M = 32, K = 3$) y de filtros coseno con diferentes valores de N .

Se observa que, efectivamente, si el valor de N es submúltiplo del factor de decimación y expresable en potencia de dos, se hace coincidir la ubicación de los ceros de los filtros. En la Fig. 3.3 se muestra la respuesta en magnitud del filtro resultante de conectar en cascada los filtros del ejemplo anterior. También, se realiza una comparación con un filtro comb con $M = 32$ y $K = 3$. Se aprecia claramente la mejora en la atenuación en todas las bandas de rechazo a excepción de las bandas 8 y 16.

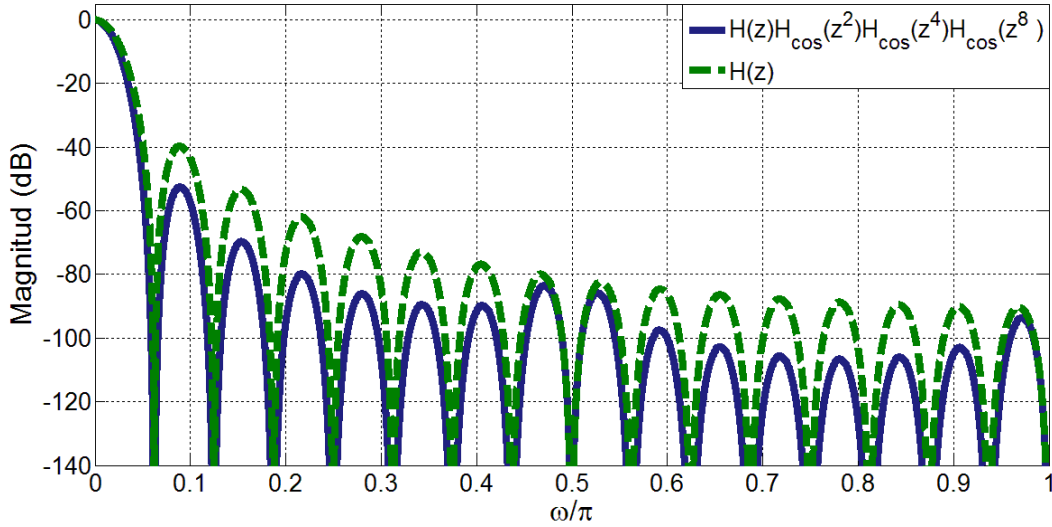


Fig. 3.3. Respuesta en magnitud de un filtro comb ($M = 32, K = 3$) conectado en cascada con filtros coseno con diferentes valores de N .

3.1.1. Filtro Coseno Expandido Rotado

Considérese que la técnica de rotación de ceros es aplicada al filtro coseno expandido. Por lo tanto, los ceros de (3.1) son rotados un ángulo α en sentido de las manecillas del reloj, teniendo como resultado la siguiente función de transferencia:

$$H_u(z^N) = 0.5(1 + z^{-N}e^{j\alpha N}). \tag{3.6}$$

De forma similar, una expresión equivalente es obtenida al aplicar la misma rotación pero en sentido contrario. Para este caso la función de transferencia resultante es:

$$H_d(z^N) = 0.5(1 + z^{-N}e^{-j\alpha N}). \tag{3.7}$$

Las filtros (3.6) y (3.7) tienen coeficientes complejos. Sin embargo, si estos son conectados en cascada se obtiene un filtro con coeficientes reales. Su función de transferencia está dada por:

$$H_r(z^N) = H_u(z^N)H_d(z^N) = 0.25[1 + (2 \cos \alpha N)z^{-N} + z^{-2N}]. \tag{3.8}$$

Su correspondiente respuesta en magnitud es:

$$|H_r(e^{j\omega})| = |\cos((\omega + \alpha)N/2) \cos((\omega - \alpha)N/2)|. \tag{3.9}$$

El filtro (3.8) puede ser introducido en diferentes etapas de la estructura de un filtro comb no recursivo, particularmente en la estructura donde el factor de decimación es expresado como potencias de dos. Por lo tanto, usando (3.4), el filtro resultante en forma general tiene como función de transferencia:

$$H_c(z) = H(z) \prod_{k=1}^R H_r(z^{M/2^k}), \quad (3.10)$$

donde R indica el número de filtros (3.8) que son introducidos. Se propone que el número máximo de filtros coseno rotados usados en dicha estructura sea de tres, con el fin de mantener un equilibrio entre la complejidad del filtro (3.10) y la mejora de la atenuación en las bandas de rechazo. En el siguiente ejemplo se ilustran las características en magnitud que presenta el filtro (3.10).

Ejemplo 3.2. La Fig. 3.4 muestra la respuesta en magnitud del filtro (3.10) para $R = 2$. También se incluyen, tanto la respuesta del filtro comb correspondiente con $M = 16$ y $K = 3$, como las respuestas de los filtros coseno rotado $H_r(z^8)$ y $H_r(z^4)$. Ambos filtros tienen el mismo valor de $\alpha = 0.0982$.

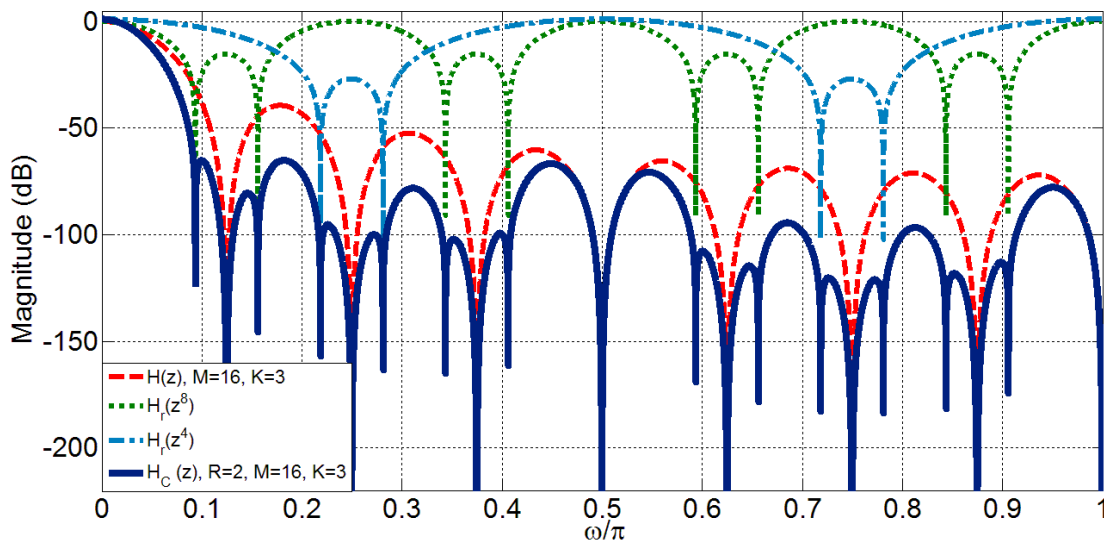


Fig. 3.4. Respuestas en magnitud del Ejemplo 3.2.

Se observa que la atenuación en todas las bandas impares, y adicionalmente, en las bandas 2 y 6 se incrementa. Sin embargo, realizando un acercamiento a la banda de paso de los filtros $H(z)$ y $H_c(z)$, como se muestra en la Fig. 3.5, es posible notar que la caída en la banda de paso del filtro $H_c(z)$ es más rápida que la del filtro comb convencional. Esto es consecuencia de la adición de los ceros de los filtros $H_r(z^8)$ y $H_r(z^4)$ a la respuesta total del filtro. Principalmente, de la contribución de $H_r(z^8)$ que agrega ceros adicionales en la primera banda de rechazo del filtro comb.

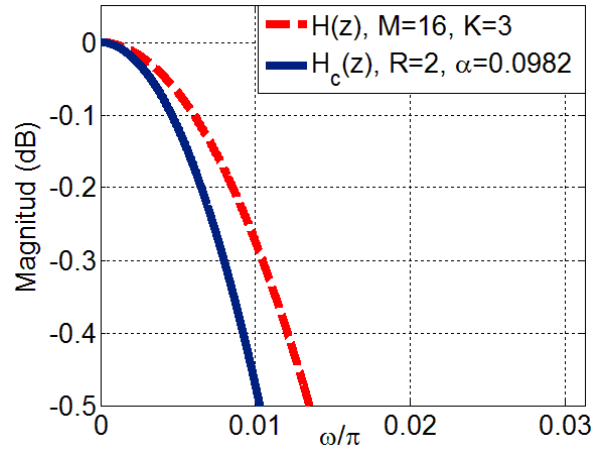


Fig. 3.5. Caída en la banda de paso de los filtros del Ejemplo 3.2.

La solución de este problema se discute en la siguiente sección.

3.2. Filtro Compensador

Con la finalidad de disminuir la caída en la banda de paso, se adoptó el filtro compensador descrito en [10]. Su función de transferencia se escribe como:

$$G(z^M) = 2^{-2}[(-1 + 2z^{-M} - z^{-2M})B + 2^2z^{-M}]. \tag{3.11}$$

Usando identidades de estructuras multi-tasa, el filtro compensador puede ser colocado en baja frecuencia, convirtiéndose así en:

$$G(z) = 2^{-2}[(-1 + 2z^{-1} - z^{-2})B + 2^2z^{-1}], \tag{3.12}$$

Su respuesta en magnitud se expresa a través de:

$$|G(e^{j\omega M})| = |1 + B \sin^2(\omega M/2)|, \tag{3.13}$$

donde B es una constante positiva. Su valor depende únicamente del número de filtros comb colocados en cascada. Sin embargo, en este caso se requiere compensar la caída en la banda de paso del filtro $H_c(z)$, en lugar de la caída de un filtro comb. Por este motivo, con el fin de adaptar el método [10] y encontrar los valores adecuados para el coeficiente B , se analizó la forma en que contribuyen los filtros coseno rotado en la caída de la banda de paso del filtro $H_c(z)$. Realizando una serie de simulaciones en MATLAB®, se buscó aproximar la caída de uno, dos y tres filtros coseno rotado conectados en cascada con la caída de filtros comb con diferentes valores de K . En las Fig.3.6 y 3.7 se muestran los resultados obtenidos.

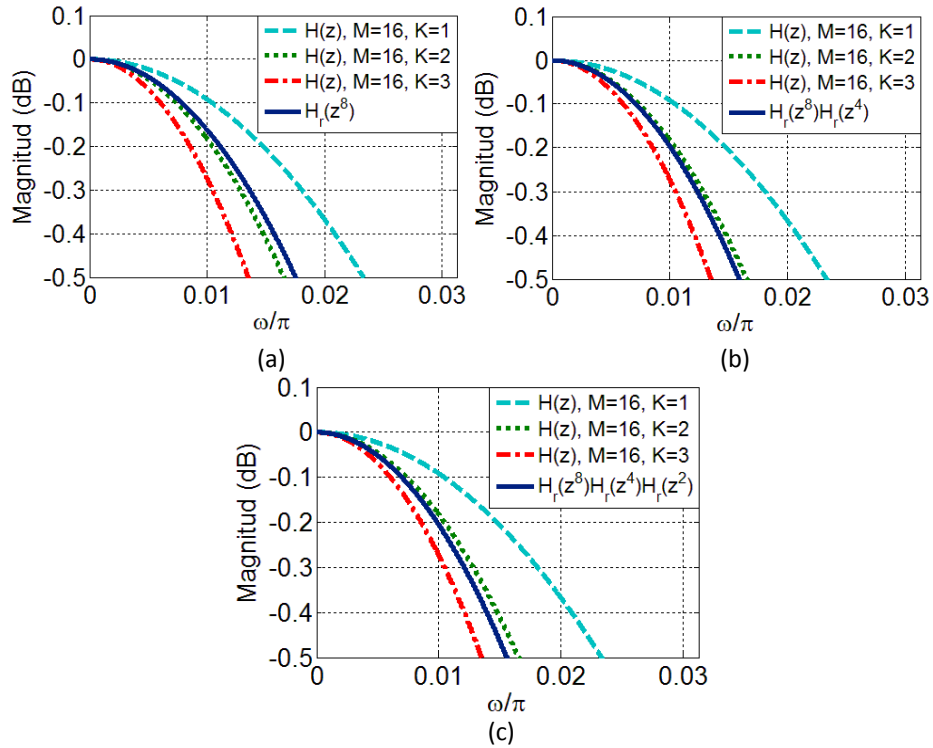


Fig. 3.6. Comparación entre la caída de la banda de paso de un filtro comb ($M = 16, K = 1, 2, y 3$) y del filtro $H_r(z^{M/2^k})$ para (a) $k = 1$, (b) $k = 1 y 2$ (c) $k = 1, 2 y 3$.

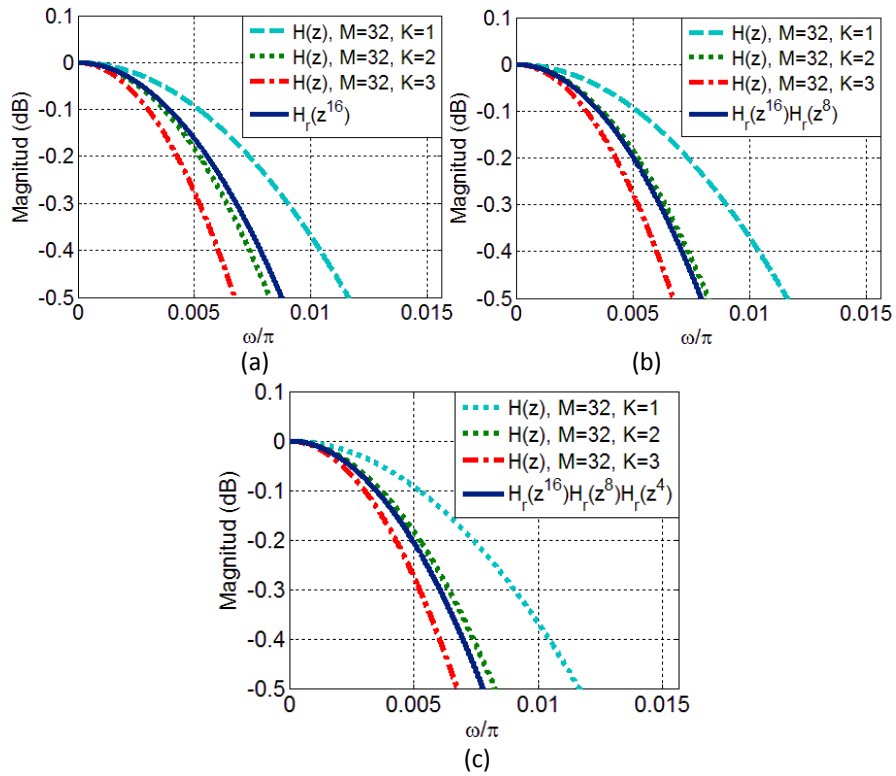


Fig. 3.7. Comparación entre la caída de la banda de paso de un filtro comb ($M = 32, K = 1, 2, y 3$) y del filtro $H_r(z^{M/2^k})$ para (a) $k = 1$, (b) $k = 1 y 2$ (c) $k = 1, 2 y 3$.

En las Fig. 3.6 y 3.7 se observa que la caída del filtro $H_r(z^{M/z^k})$, no importando el número de filtros coseno rotados que lo integren ni el valor del factor de decimación, es aproximadamente igual a la caída de un filtro comb con $K = 2$. En consecuencia, el valor de B se obtiene para el parámetro $K + 2$. De esta forma, aplicando el método de diseño [10] y considerando diferentes factores residuales v , se encontraron valores adicionales de B . Estos se presentan en la Tabla 3.2.

$K + 2$	B		
	$N = 50, \delta = 0.4, v = 2$	$N = 50, \delta = 0.04, v = 4$	$N = 50, \delta = 0.004, v = 8$
3	0.75	0.5	0.5
4	1	0.75	0.75
5	1.25	0.75	0.75
6	1.5	1	1
7	1.75	1.25	1.25

Tabla 3.2. Valores de B calculados considerando el parámetro $K + 2$.

Ejemplo 3.3. En este ejemplo se consideran tres filtros $H_c(z)$ con el mismo factor de decimación $M = 32$ y orden $K = 5$. En el primero de ellos se introduce sólo un filtro coseno rotado ($R = 1$), en el segundo dos ($R = 2$) y en el tercero tres ($R = 3$). A todos los filtro se les aplica una rotación por $\alpha = 0.0491$ y el valor del factor residual es $v = 2$. Conociendo lo anterior, cada filtro es conectado en cascada con un filtro compensador cuyo coeficiente $B = 1.75$. En la Fig. 3.8 se muestra la respuesta en magnitud en la banda de paso de los filtros anteriores.

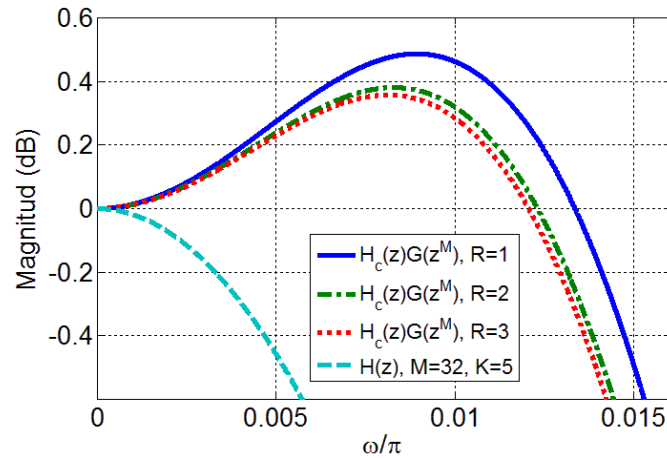


Fig. 3.8. Banda de paso de los filtros del Ejemplo 3.3.

La respuesta ideal que se espera en la banda de paso del filtro comb, es que presente una magnitud constante de 0 dB a lo largo del intervalo de frecuencias $[0, \omega_p]$. Entonces, la caída en la banda de paso se mide como la máxima desviación que existe en relación a 0 dB en $[0, \omega_p]$. Considerando lo anterior, se observa en la Fig. 3.8 que la caída en la banda de paso de cada filtro $H_c(z)$ compensado es menor a la caída en el filtro comb. Sin embargo, existe

todavía una desviación considerable, principalmente en la frecuencia $\omega_p = 0.0491$. En la Tabla 3.3 se incluyen los valores de la máxima desviación presente en la banda de paso de cada uno de filtros del ejemplo anterior. Esta se encuentra en la frecuencia ω_p .

Filtro	Magnitud (dB) en ω_p
$H(z), M = 32, K = 3$	-4.541
$H_c(z)G(z^M), R = 1$	-0.720
$H_c(z)G(z^M), R = 2$	-1.070
$H_c(z)G(z^M), R = 3$	-1.155

Tabla 3.3. Caída en la banda de paso de los filtros del Ejemplo 3.3.

Con la finalidad de solucionar este inconveniente, se analizó la respuesta del filtro $H_c(z)$ compensado para cada uno de los valores de B previamente calculados. Al igual que en el ejemplo anterior, en algunos casos la banda de paso no se compensaba adecuadamente.

También se observó que, a pesar de que la caída en los filtros coseno rotado es similar a un filtro comb con $K = 2$, el filtro $H_c(z)$ no presenta una misma respuesta para diferentes valores de R . Por lo tanto, fue necesario en reajustar el valor de B en dependencia con R para obtener una mejor compensación en la banda de paso. Los valores fueron obtenidos a partir de la realización de numerosas simulaciones en MATLAB®. Estos se muestran en la Tabla 3.3.

K	B								
	v = 2			v = 4			v = 8		
	R = 1	R = 2	R = 3	R = 1	R = 2	R = 3	R = 1	R = 2	R = 3
1	0.68	0.79	0.83	0.46	0.54	0.55	0.42	0.49	0.50
2	0.98	1.1	1.13	0.65	0.73	0.75	0.60	0.66	0.68
3	1.30	1.45	1.45	0.85	0.92	0.94	0.77	0.83	0.85
4	1.65	1.8	1.85	1.05	1.13	1.15	0.94	1.01	1.03
5	2.00	2.23	2.23	1.26	1.33	1.35	1.12	1.18	1.20

Tabla 3.4. Valores ajustados de B.

En las Tablas 3.5, 3.6 y 3.7 se presenta una comparación entre la caída de la banda de paso del filtro $H_c(z)$ compensado y la caída de un filtro comb. Se consideran diferentes valores en M, K, R y v . Los valores de B se eligen de acuerdo a lo mostrado en la Tabla 3.4.

K	H(z)			H _p (z)								
				Máxima Desviación (dB)								
	M=16	M=32	M=64	M=16			M=32			M=64		
				R=1	R=2	R=3	R=1	R=2	R=3	R=1	R=2	R=3
1	-0.911	-0.908	-0.908	0.255	0.301	0.337	0.254	0.300	0.336	0.253	0.300	0.335
2	-1.823	-1.816	-1.818	0.395	0.447	0.460	0.393	0.444	0.458	0.392	0.444	0.457
3	-2.735	-2.725	-2.727	0.537	0.624	0.587	0.533	0.621	0.584	0.533	0.620	0.583
4	-3.646	-3.633	-3.636	0.694	0.767	0.803	0.689	0.762	0.799	0.688	0.761	0.798
5	-4.558	-4.541	-4.545	0.821	0.990	0.954	0.815	0.984	0.949	0.814	0.983	0.948

Tabla 3.5. Comparación de la caída en la banda de paso de $H(z)$ y $H_c(z)$ compensado para $v = 2$.

K	H(z)			H _p (z)								
	Máxima Desviación (dB)											
	M=16	M=32	M=64	M=16			M=32			M=64		
			R=1	R=2	R=3	R=1	R=2	R=3	R=1	R=2	R=3	
1	-0.222	0.223	-0.168	0.012	0.017	0.016	0.012	0.016	0.015	0.012	0.016	0.015
2	-0.445	-0.446	-0.447	0.019	0.024	0.025	0.019	0.023	0.025	0.019	0.023	0.025
3	-0.668	-0.670	-0.670	0.028	0.033	0.034	0.027	0.032	0.033	0.027	0.032	0.033
4	-0.891	-0.889	-0.894	0.039	0.045	0.046	0.038	0.043	0.045	0.037	0.043	0.045
5	-1.114	-1.119	-1.118	0.053	0.056	0.057	0.052	0.054	0.055	0.051	0.054	0.055

Tabla 3.6. Comparación de la caída en la banda de paso de H(z) y H_c(z) compensado para v = 4.

K	H(z)			H _p (z)								
	Máxima Desviación (dB)											
	M=16	M=32	M=64	M=16			M=32			M=64		
			R=1	R=2	R=3	R=1	R=2	R=3	R=1	R=2	R=3	
1	-0.055	-0.055	-0.058	0.0007	0.001	0.001	0.0007	0.0009	0.0009	0.0006	0.0009	0.0009
2	-0.111	-0.111	-0.117	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
3	-0.166	-0.167	-0.175	0.001	0.002	0.002	0.001	0.001	0.001	0.001	0.001	0.001
4	-0.221	-0.222	-0.234	0.002	0.002	0.003	0.002	0.002	0.002	0.002	0.002	0.002
5	-0.277	-0.278	-0.292	0.003	0.003	0.003	0.002	0.003	0.003	0.002	0.003	0.003

Tabla 3.7. Comparación de la caída en la banda de paso de H(z) y H_c(z) compensado para v = 8.

De acuerdo a los resultados anteriores, la mayor desviación en la banda de paso del filtro H_c(z) compensado, se presenta para el caso de K = 5 y v = 2. Esta desviación es de aproximadamente 0.9 dB. Sin embargo, en comparación con la desviación presente de un filtro comb del mismo orden, que es de -4.5 dB, se logra disminuir considerablemente. Por último, el inconveniente de usar de los valores de B de la Tabla 3.4, es que estos requieren de multiplicadores para su implementación. La solución a este problema se presenta en la sección 3.4.

3.3. Estructura Propuesta

Considere que M = 2^p, N = M/2^k y usando (3.10), la función de transferencia del filtro propuesto es:

$$H_p(z) = H(z) \prod_{k=1}^R H_r(z^{M/2^k}) G(z^M), \tag{3.14}$$

donde H(z), H_r(z^{M/2^k) y G(z^M) están dadas en (1.2), (3.8) y (3.12) respectivamente.}

Su correspondiente respuesta en magnitud se escribe como:

$$H_p(e^{j\omega}) = \left[\frac{1 \sin(\omega M/2)}{M \sin(\omega/2)} \right] \left[\prod_{k=1}^R [\cos([\omega + \alpha]M/2^{k+1})] \times [\cos([\omega - \alpha]M/2^{k+1})] \right] \times [1 + B \sin^2(\omega M/2)]. \tag{3.15}$$

El valor del parámetro α se elige de tal forma que sea igual al valor de la frecuencia de paso ω_p . Es así como, las bandas de rechazo del filtro propuesto son ensanchadas para abarcar por completo el ancho de banda de las réplicas de la señal de entrada. Esto tiene como objetivo que el filtro propuesto presente mayor rechazo al fenómeno de aliasing. Por lo tanto, considerando un factor residual ν , el valor de α es igual a:

$$\alpha = \omega_p = \frac{\pi}{\nu M}. \tag{3.16}$$

Finalmente, como se mencionó, este método de diseño sólo es aplicable en la estructura del filtro comb no recursivo con $M = 2^p$. En la Fig. 3.9 se muestra la forma general del diagrama a bloques del filtro propuesto.

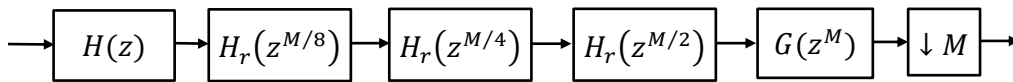
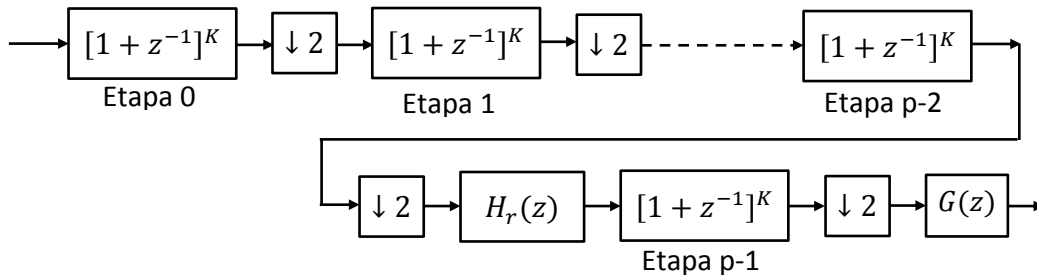


Fig. 3.9. Diagrama a bloques de la estructura general del filtro propuesto para $R = 3$.

Usando identidades de estructuras multi-tasa, y considerando sólo aplicar de uno, dos ó tres filtros coseno rotado para mantener una baja complejidad. Las estructuras eficientes para la implementación del filtro propuesto se muestran en la Fig. 3.10.



(a)

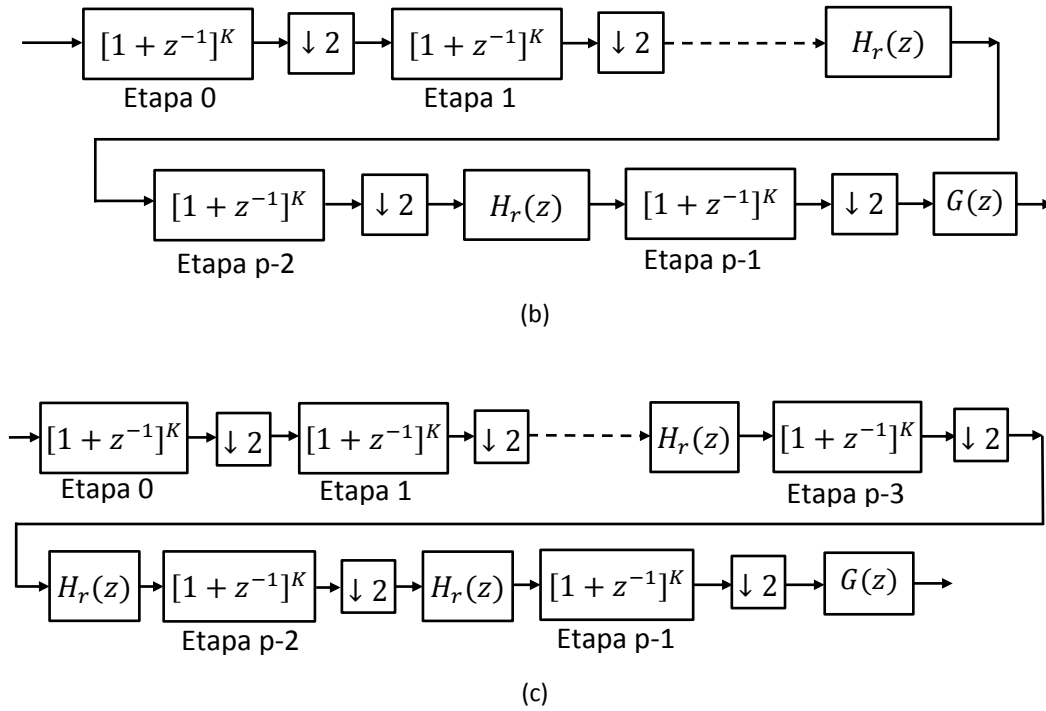


Fig. 3.10. Diagrama a bloques de la estructura del filtro propuesto. (a) $R = 1$, (b) $R = 2$ y (c) $R = 3$.

3.3.1. Respuesta en Magnitud de la Estructura Propuesta

En este apartado a través de una serie de ejemplos, se ilustran las características en magnitud que posee la estructura propuesta. También, se enlista las ventajas y desventajas que presenta.

Ejemplo 3.4. Considere el diseño de un filtro decimador con $M = 16$ y $K = 3$. Se tiene un factor residual de $v = 2$ y se elige que el número de filtros coseno rotados incluidos en la estructura sea $R = 2$. Conociendo esto, el valor de los parámetros restantes es de $B = 1.45$ y $\alpha = 0.0982$. Por lo tanto, la función de transferencia de este filtro es:

$$H_{p_1}(z) = \left[\prod_{i=0}^3 \frac{1}{2} (1 + z^{-2^i}) \right] H_r(z^{M/4}) H_r(z^{M/2}) G(z^M). \tag{3.17}$$

En la Fig. 3.11 se muestra su respuesta en magnitud. Se realiza una comparación con la respuesta de un filtro comb con $M = 16$ y $K = 3$.

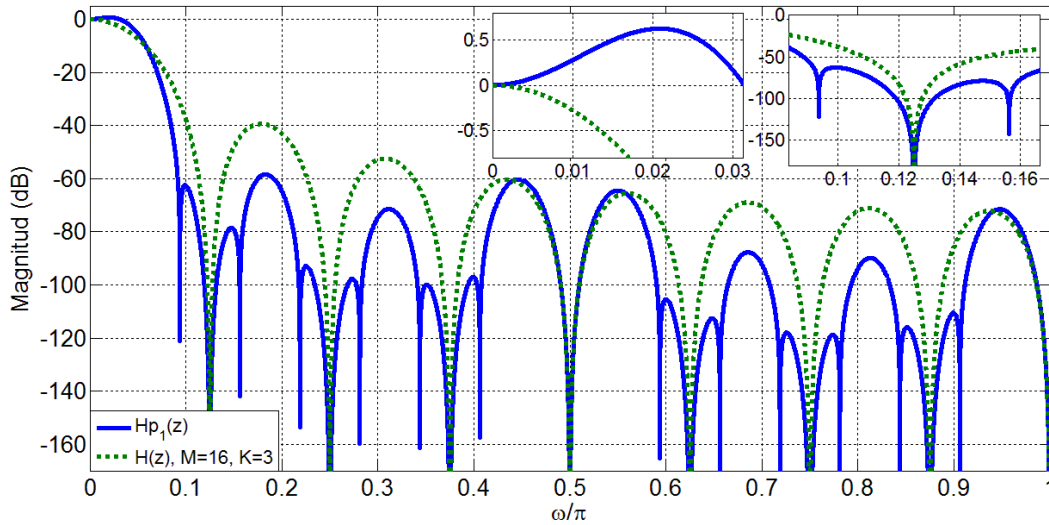


Fig. 3.11. Respuestas en magnitud del Ejemplo 3.4.

Se observa claramente que el filtro propuesto tiene una mejor respuesta en magnitud que un filtro comb convencional. Este presenta una mayor atenuación en las bandas de rechazo impares, y adicionalmente en las bandas 6 y 7. La caída en la banda de paso se ve reducida por la acción del filtro compensador. Sin embargo, su desventaja radica en que requieren dos multiplicadores adicionales en su implementación, además del ya requerido por el filtro compensador. Estos multiplicadores trabajan en frecuencias intermedias y son necesarios para representar los coeficientes $2 \cos(\alpha(M/2))$ y $2 \cos(\alpha(M/4))$ de los filtros $H_r(z^{M/2})$ y $H_r(z^{M/4})$, respectivamente. El diagrama a bloques de su estructura eficiente se muestra en la Fig. 3.12.

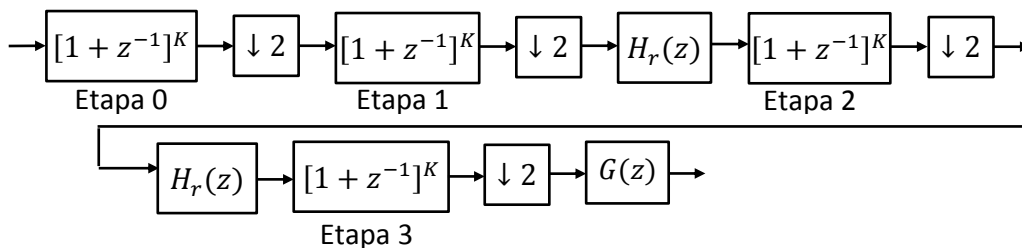


Fig. 3.12. Diagrama a bloques de la estructura del filtro del Ejemplo 3.4.

Ejemplo 3.5. En este ejemplo se utilizan tres filtros coseno rotados ($R = 3$) y se considera un factor residual $v = 4$. El filtro comb al que se le aplican dichos filtros tiene como parámetros de diseño $M = 32$ y $K = 5$. De la Tabla 3.4 se obtiene que el valor de coeficiente del compensador es de $B = 1.35$. Al utilizar (3.16) se determina que $\alpha = 0.0245$. La función de transferencia de este filtro se escribe como:

$$H_{p_2}(z) = \left[\prod_{i=0}^4 \frac{1}{2} (1 + z^{-2^i}) \right] H_r(z^{M/8}) H_r(z^{M/4}) H_r(z^{M/2}) G(z^M). \quad (3.18)$$

En la Fig. 3.13 se muestra la respuesta en magnitud de (3.18) y se hace una comparación con la respuesta de un filtro comb con $M = 32$ y $K = 5$.

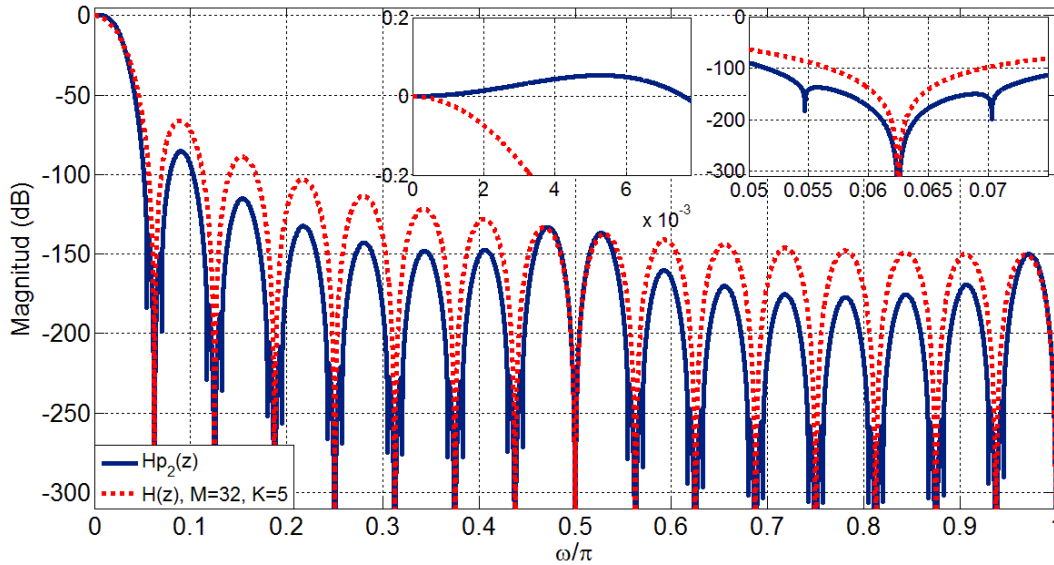


Fig. 3.13. Respuestas en magnitud del Ejemplo 3.5.

Al igual que el ejemplo anterior, el filtro propuesto tiene mejores características en magnitud. El diagrama a bloques de la estructura requerida para su implementación se ilustra en la Fig. 3.14.

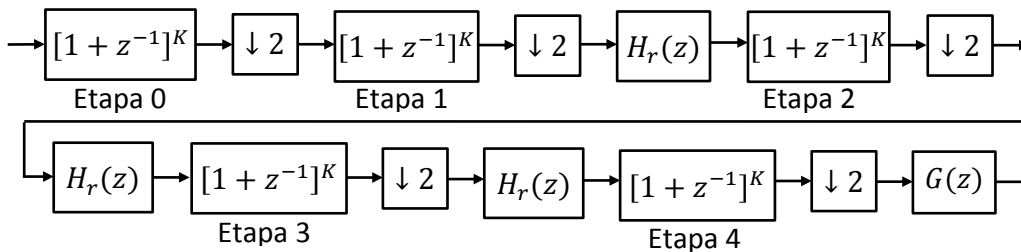


Fig. 3.14. Diagrama a bloques de la estructura del filtro del Ejemplo 3.5.

Ejemplo 3.6. Considérese el uso de tres filtros coseno rotados ($R = 3$). El filtro comb que se usa como base para el diseño del filtro decimador propuesto tiene como parámetros $M = 32$ y $K = 5$. Se tiene como factor residual $v = 8$. Conociendo lo anterior, los valores de los parámetros restantes son $B = 1.20$ y $\alpha = 0.0123$. La función de transferencia de este filtro se escribe como:

$$H_{p_3}(z) = \left[\prod_{i=0}^4 \frac{1}{2} (1 + z^{-2^i}) \right] H_r(z^{M/8}) H_r(z^{M/4}) H_r(z^{M/2}) G(z^M). \quad (3.19)$$

En la Fig. 3.15 se observa la respuesta en magnitud del filtro propuesto (3.19). Se realiza una comparación con la respuesta de un filtro comb con $M = 32$ y $K = 5$. La correspondiente estructura para su implementación es igual a la mostrada en la Fig. 3.14.

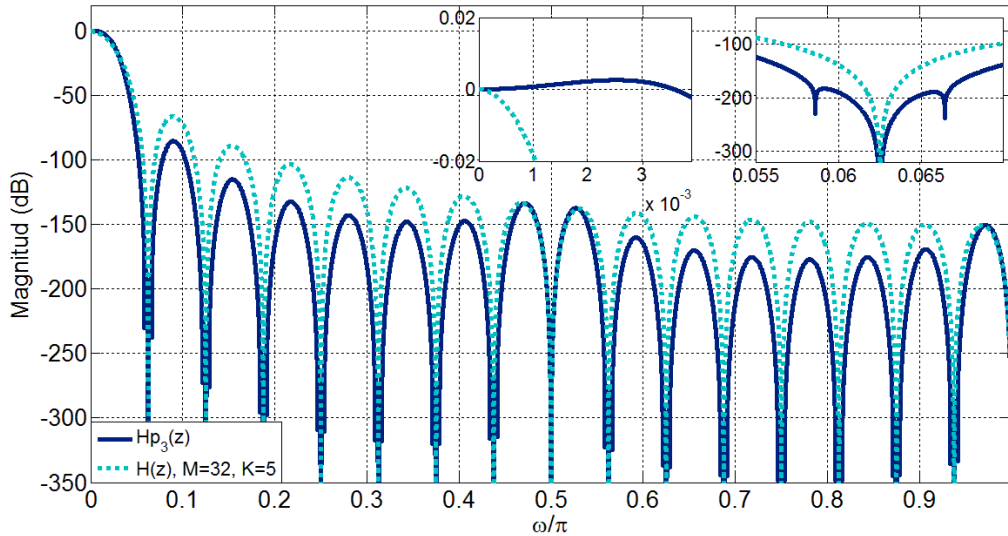


Fig. 3.15. Respuestas en magnitud del Ejemplo 3.6.

De acuerdo con los resultados obtenidos en los ejemplos anteriores, se realizan las siguientes observaciones:

- El filtro propuesto tiene mejores características en magnitud tanto en la banda de paso como en las bandas de rechazo.
- Se obtiene una mejor compensación en la banda de paso para los factores residuales $v = 4$ y $v = 8$. Además, se requiere el uso de un multiplicador para la implementación del único coeficiente del filtro compensador. Este trabaja en baja frecuencia.
- El número de bandas de rechazo mejoradas dependerá del número de filtros coseno rotado utilizados.
- Cada filtro coseno rotado que se introduce en la estructura, implica el uso de un multiplicador. Este trabaja en una frecuencia intermedia, la cual, depende de la posición en la que se localice el filtro $H_r(z)$ dentro en la estructura del filtro decimador (véase Fig. 3.10).
- Por lo tanto, se establece como máximo usar tres filtros coseno rotado. De esta forma, no se incrementa considerablemente la complejidad de la estructura a implementar.

El problema de eliminar los multiplicadores en esta estructura, se discute en la Sección 3.4. A continuación, se realiza una comparación entre el método propuesto y otro ya existente.

3.3.2. Comparación con el Filtro Comb Generalizado (GCF) [22]

Se presenta una comparación con el método descrito en [22].

Ejemplo 3.7. Considere un filtro comb con $M = 16$, $K = 3$ y un factor residual $\nu = 4$. Utilizando el método propuesto para $R = 1$. Los parámetros de diseño restantes de este filtro son $B = 0.85$ y $\alpha = 0.0491$. De esta forma, se obtiene como resultado la siguiente función de transferencia:

$$H_{p_4}(z) = \left[\prod_{i=0}^3 \frac{1}{2} (1 + z^{-2^i}) \right] H_r(z^{M/2}) G(z^M). \quad (3.20)$$

La respuesta en magnitud del filtro $H_{p_4}(z)$, se compara con la respuesta de un filtro comb generalizado de tercer orden (GCF) con $q = [-0.79, 0, 0.79]$. La Fig. 3.16 muestra las respuestas en magnitud de ambos filtros y la del filtro comb correspondiente ($M = 16, K = 3$).

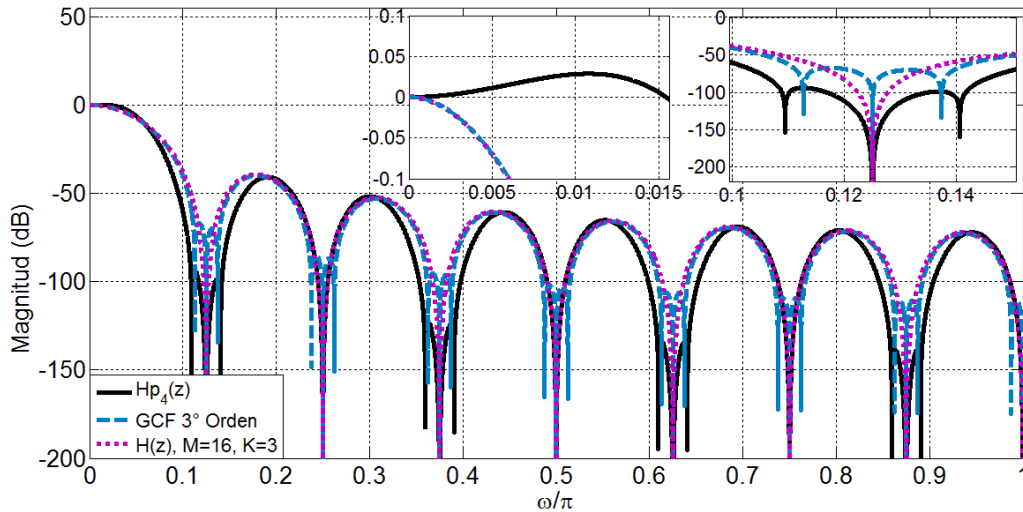


Fig. 3.16. Respuestas en magnitud del Ejemplo 3.7.

Se observa que la respuesta en magnitud de un GCF, tiene la ventaja de presentar un incremento en atenuación en todas las bandas de rechazo. Esto no sucede con la respuesta del filtro propuesto, el cual, sólo presenta una mejora en las bandas impares.

Sin embargo, la rotación de ceros en un GCF se aplica a la estructura recursiva del filtro comb. Derivando, en el problema de que al utilizar precisión finita en la representación de los coeficientes, puede producirse inestabilidad en el filtro. Esto se debe principalmente a la no cancelación perfecta de los polos y ceros. En cambio, el filtro propuesto tiene una estructura no recursiva; teniendo así como ventaja no presentar problemas de inestabilidad. Además, por medio del filtro compensador, se disminuye la caída en la banda de paso. Esto no ocurre en la respuesta del GCF, en donde la caída en la banda de paso sigue siendo igual a la de un filtro comb convencional.

3.4. Diseño de la Estructura Propuesta sin Multiplicadores

Como se mencionó en los apartados (3.3) y (3.3.1), la estructura del filtro propuesto tiene el inconveniente de requerir un multiplicador debido al coeficiente del filtro compensador, y uno por cada filtro coseno rotado que es utilizado. Otra desventaja, es que los multiplicadores que se requieren en la implementación de los filtros coseno rotado, no trabajan en la frecuencia más baja posible. Su frecuencia de trabajo depende de la posición que ocupen dentro de la estructura del filtro propuesto. En consecuencia, hay un incremento en el consumo de potencia. Por lo tanto, con la finalidad de obtener una estructura libre de multiplicadores, tanto los valores del coeficiente B como del coeficiente $2 \cos(\alpha N)$ presente en (3.8), son redondeados utilizando las siguientes expresiones:

$$B_r = r \times \text{round}[B / r], \quad (3.21)$$

$$\beta_r = r \times \text{round}[2 \cos(\alpha N) / r], \quad (3.22)$$

donde r corresponde a la constante de redondeo cuyo valor es $r = 2^{-6}$. La expresión $\text{round}(x)$, indica que el valor del argumento x será redondeado al entero más próximo. Esto permite que la implementación de los coeficientes de (3.8) y (3.11) requieran sólo de sumas y corrimientos. Por lo tanto, la función de transferencia de los filtros (3.8) y (3.11) son reescritas como:

$$G'(z^M) = 2^{-2}[(-1 + 2z^{-M} - z^{-2M})B_r + 2^2 z^{-M}]. \quad (3.23)$$

$$H'_r(z^N) = 0.25[1 + \beta_r z^{-N} + z^{-2N}]. \quad (3.24)$$

Finalmente, a consecuencia de lo anterior, la función de transferencia del filtro propuesto que no requiere de multiplicadores es:

$$H_p(z) = H(z) \prod_{k=1}^R H'_r(z^{M/2^k}) G'(z^M), \quad (3.25)$$

En el siguiente ejemplo se muestra el impacto que tiene el redondeo de los coeficientes, en la respuesta en magnitud del filtro propuesto.

Ejemplo 3.8. En este ejemplo, los parámetros de diseño del filtro propuesto son $M = 32$, $K = 5$, $R = 3$, $B = 2.23$, $v = 2$ y $\alpha = 0.0491$. De esta forma, se tiene como función de transferencia:

$$H_{p_5}(z) = H(z) H'_r(z^{16}) H'_r(z^8) H'_r(z^4) G'(z^M). \quad (3.26)$$

Considerando la constante $r = 2^{-6}$, los valores redondeados de los coeficientes de los filtros coseno rotado y del filtro compensador se incluyen en la Tabla 3.8.

Filtro	Coefficiente redondeado
$H_r(z^{16})$	$2^1 - 2^{-1} - 2^{-4} - 2^{-6}$
$H_r(z^8)$	$2^1 - 2^{-3} - 2^{-5}$
$H_r(z^4)$	$2^1 - 2^{-5}$
$G(z^{32})$	$2^1 + 2^{-2} - 2^{-6}$

Tabla 3.8. Valores de los coeficientes redondeados en el Ejemplo 3.8.

En la Fig. 3.17 se muestra la respuesta en magnitud del filtro (3.26). También se incluye la respuesta un filtro $H_p(z)$. Este tiene los mismos parámetros de diseño que (3.26), pero con la diferencia de que no se aplica redondeo en sus coeficientes. Ambas respuestas son comparadas con la de un filtro comb con $M = 32$ y $K = 5$. Se comprueba que el aplicar redondeo en los coeficientes de los filtros coseno rotado y del filtro compensador, no provoca un cambio significativo en la respuesta del filtro propuesto. Por lo tanto, esta es una opción viable para no requerir de multiplicadores e implementar dichos coeficientes a base de sumas y corrimientos.

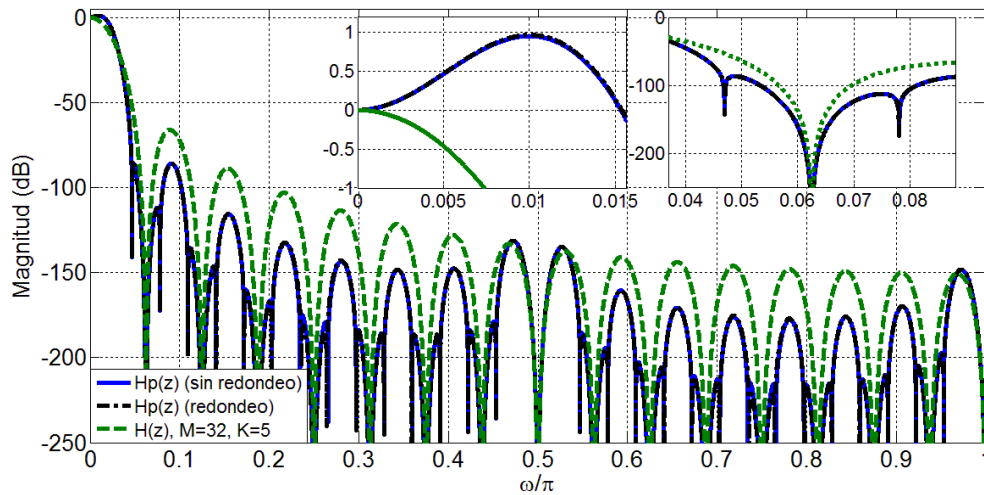


Fig. 3.17. Respuestas en magnitud del Ejemplo 3.8.

El diagrama a bloques de la estructura eficiente para la implementación de (3.24) se presenta en la Fig. 3.18.

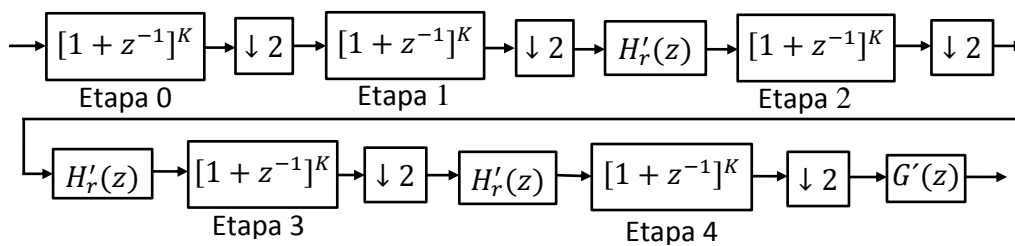


Fig. 3.18. Diagrama a bloques de la estructura del filtro del Ejemplo 3.8.

3.4.1. Comparación con Otros Métodos Libres de Multiplicadores

La estructura libre de multiplicadores del filtro propuesto se compara con las estructuras descritas en [44] y [45] en los siguientes ejemplos.

Comparación con el Método [43]

Ejemplo 3.9. Los parámetros de diseño del filtro propuesto son $M = 16, K = 5, v = 4, \alpha = 0.0491, R = 2$ y $B = 1.33$. Los valores redondeados de sus coeficientes se incluyen en la Tabla 3.9.

Filtro	Coefficiente redondeado
$H_r(z^8)$	$2^1 - 2^{-3} - 2^{-5}$
$H_r(z^4)$	$2^1 - 2^{-5}$
$G(z^{16})$	$2^0 + 2^{-2} + 2^{-4} + 2^{-6}$

Tabla 3.9. Valores de los coeficientes redondeados en el Ejemplo 3.9.

La respuesta en magnitud de este filtro se compara con la de un filtro diseñado a partir del método descrito en [43]. Utilizando la misma nomenclatura que [43], los parámetros de diseño elegidos son $M = 16, M_1 = 8, K = 3, v = 4, b = 0, \alpha = 0.0449823, k = 4$ y $l = 15$.

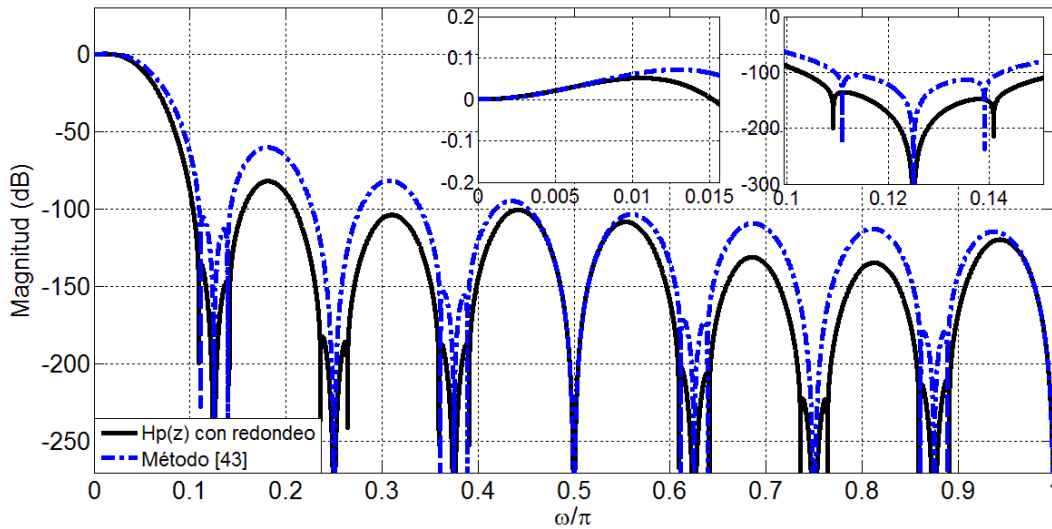


Fig. 3.19. Respuestas en magnitud del Ejemplo 3.9.

Se observa que la respuesta en magnitud del filtro propuesto tiene mejores características. En la banda de paso presenta una menor desviación, en comparación con el filtro diseñado con el método [43]. También, se muestra un aumento en atenuación en una mayor cantidad de bandas de rechazo. Esto se consigue sin incrementar en forma considerable la complejidad de la estructura a implementar.

Comparación con el Método [44]

Ejemplo 3.10. Considérese un filtro comb cuyos parámetros de diseño son $M = 16$ y $K = 5$. Este filtro es utilizado en la construcción de la estructura propuesta, donde se elige que el número de filtros coseno rotados agregados sea de $R = 2$ y se tiene un factor residual $\nu = 2$. Conociendo los valores anteriores y aplicando el método de diseño anteriormente descrito, se obtiene como parámetros restantes $\alpha = 0.0982$ y $B = 2.23$. Los valores redondeados de los coeficientes de los filtros coseno y del filtro compensador se incluyen en la Tabla 3.10.

Filtro	Coefficientes Redondeado
$H_r(z^8)$	$2^1 - 2^{-1} - 2^{-4} - 2^{-6}$
$H_r(z^4)$	$2^1 - 2^{-3} - 2^{-5}$
$G(z^{16})$	$2^1 + 2^{-2} - 2^{-6}$

Tabla 3.10. Valores de los coeficientes redondeados en el Ejemplo 3.10.

El método de diseño descrito en [44] es aplicado considerando utilizar el mismo filtro comb de la estructura anterior. Usando la misma nomenclatura que [44], se tienen como parámetros de diseño $N_1 = 7$, $N_2 = 9$ y $b = -1$.

En la Fig.3.20 se muestra la respuesta en magnitud de los dos filtros decimadores previamente diseñados.

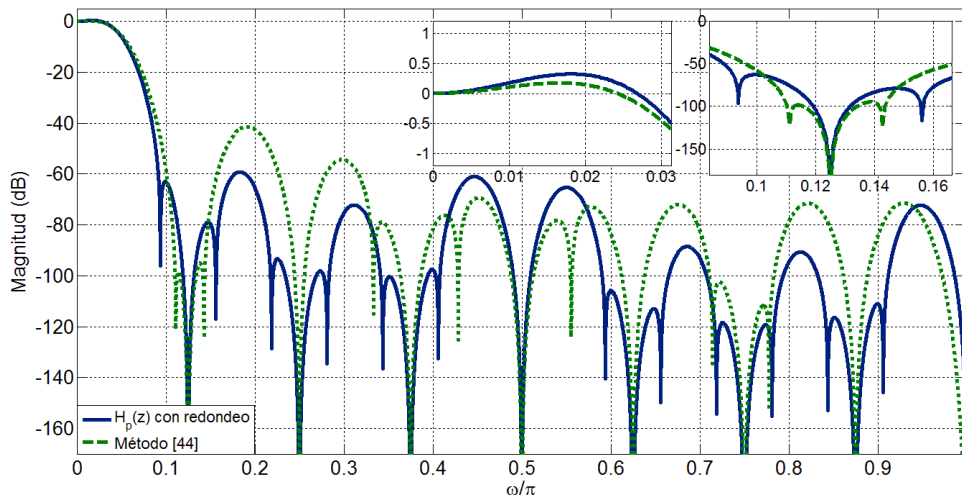


Fig. 3.20. Respuestas en magnitud del Ejemplo 3.10.

La respuesta en magnitud del filtro propuesto tiene la desventaja de exhibir una mayor desviación en la banda de paso, en comparación con el filtro [44]. Por otro lado, se observa que la atenuación en la primera banda de rechazo en ambos filtros es similar. Sin embargo, el filtro propuesto presenta una mejora en las bandas impares y adicionalmente en las bandas 2 y 6. En cambio el filtro [44] sólo tiene mejor atenuación en las bandas 1, 3 y 6.

3.5. Complejidad Computacional de la Estructura Propuesta

Una forma de medir la complejidad computacional de un filtro digital, consiste en determinar el número de adiciones por muestra de salida (APOS). Para un filtro comb no recursivo esta cantidad se obtiene a partir de:

$$APOS = K(2^k + 2^{k-1} + \dots + 2) = 2K(2^k - 1), \quad (3.24)$$

En el caso de la estructura propuesta, tanto el compensador como los filtros coseno rotados participan en el incremento de APOS. La forma en que contribuye cada filtro, sin considerar que sus coeficientes son expresados como sumas y corrimientos, se enlista en la Tabla 3.11.

Filtro	APOS Adicionales
$H_r(z^{M/2})$	4
$H_r(z^{M/4})$	8
$H_r(z^{M/8})$	16
$G(z)$	3

Tabla 3.11. Cantidad de APOS adicionales requeridas por la estructura propuesta.

Finalmente, para obtener el número total de APOS de la estructura propuesta, se debe de agregar las adiciones requeridas para la implementación de los coeficientes redondeados. En el siguiente ejemplo se compara la complejidad computacional de la estructura propuesta, con otras reportadas en la literatura.

Ejemplo 3.11. Se considera un factor de decimación $M = 16$ y un factor residual $v = 4$. A partir de estos valores, se diseña un filtro decimador no recursivo basado en filtros comb, utilizando tres métodos distintos. En primer lugar, se aplica el método propuesto teniendo en cuenta los parámetros $K = 4, R = 2, \alpha = 0.0491$ y $B = 1.13$. Los valores redondeados de los coeficientes de los filtros coseno y del coeficiente del filtro compensador se incluyen en la Tabla 3.12.

Filtro	Coficiente Redondeado
$H_r(z^8)$	$2^1 - 2^{-3} - 2^{-5}$
$H_r(z^4)$	$2^1 - 2^{-5}$
$G(z^{16})$	$2^0 + 2^{-3}$

Tabla 3.12. Valores de los coeficientes redondeados en el Ejemplo 3.11.

La complejidad computacional de este filtro, es comparada con la de los filtros [32] y [35]. Respetando la nomenclatura de ambas referencias. Los parámetros de diseño de [32] son $K_1 = 4$ y en la última etapa se aplica el polinomio de Sharpening $3x^2 - 2x^3$. El filtro obtenido al aplicar con el método [35] tiene como parámetros de diseño el polinomio $5.5x^3 - 4.5x^4$, $l = 1, K_1 = 4$ y $K_2 = 5$. El número de APOS requeridas por cada estructura se muestran en la

Tabla 3.13. También, se incluyen los valores de la caída de la banda de paso y el peor caso de atenuación para cada uno de los filtros.

Estructura	APOS	Caída en la Banda de Paso	Peor Caso de Atenuación
Método [32]	138	-0.3592	-93.66
Método [35]	134	-0.0266	-93.32
Propuesto	139	-0.0075	-165.6

Tabla 3.13. Comparación en APOS del Ejemplo 3.11.

Se observa que la complejidad de la estructura propuesta, es mayor en una APO en comparación con el método [32] y en 5 APOS en comparación con el método [35]. Sin embargo, este pequeño incremento proporciona que la estructura propuesta tenga mejores características en magnitud tanto en la banda de paso como en las bandas de rechazo.

Capítulo 4

Descripción en VHDL de los Filtros Decimadores Propuestos

En este capítulo se ilustra el procedimiento general para la descripción en VHDL (Very High Speed Integrated Circuit Hardware Description Language) de los filtros decimadores propuestos. En primer lugar, se muestra la forma de describir cada bloque que integra la estructura de un filtro comb no recursivo. Posteriormente, aplicando el método de diseño propuesto en el capítulo anterior, se realizan las modificaciones necesarias para mejorar su respuesta en magnitud. Finalmente, se presentan ejemplos que permiten resumir la metodología de diseño aquí mencionada.

4.1. Introducción

El diseño de sistemas digitales puede realizarse en diferentes niveles de abstracción. En la actualidad, la densidad y funcionalidad de estos sistemas se ha incrementado considerablemente. En consecuencia, diseñar sistemas en niveles bajos, como a nivel transistor, se ha vuelto muy complejo. Por lo tanto, la mayoría de los diseñadores se han desplazado hacia niveles altos de abstracción. De esta forma, partiendo del uso de compuertas lógicas hasta bloques completos (multiplexores, comparadores, sumadores etc.), la implementación de un sistema digital se ha facilitado.

El fácil acceso y aprendizaje de los lenguajes de descripción de hardware (HDL por sus siglas en inglés), así como de las herramientas de síntesis y simulación de circuitos, han modificado la forma de diseñar sistemas digitales durante los últimos años. Entre estos lenguajes se encuentra VHDL (*Very High Speed Integrated Circuit Hardware Description Language*). Este permite que el comportamiento de diseños de sistemas digitales pueda ser especificado y simulado con precisión. Entre sus características más importantes se encuentran:

- Se puede establecer jerarquía entre los elementos que integran el diseño de un sistema digital.
- Todo elemento del diseño tiene definido de forma precisa una interfaz y un comportamiento.
- La concurrencia, temporización y señales de reloj pueden ser modeladas. VHDL maneja estructuras de circuitos secuenciales asíncronos y síncronos.
- La operación lógica y de comportamiento de un diseño pueden simularse.
- Se puede utilizar un algoritmo o una estructura de hardware real para definir la operación de un elemento.

Por estas razones, se considera utilizar VHDL en la implementación de los filtros decimadores propuestos. En forma general, estos filtros requieren de bloques aritméticos como sumadores y restadores. También son necesarios elementos de memoria como flip-flops. En los siguientes apartados se ilustra el proceso para su descripción en VHDL.

4.2. El Divisor de Frecuencia

En el diseño e implementación de filtros para aplicaciones multi-tasa, uno de los bloques más importantes es el divisor de frecuencia. Su función principal es generar las señales de control para activar en los tiempos correspondientes, cada uno de los bloques que integran el sistema. Es así como la reducción de frecuencia y filtrado de la señal digital de entrada puede llevarse a cabo.

En el caso particular de los filtros comb no recursivos con $M = 2^p$, la reducción de frecuencia se hace de forma gradual. Cada etapa del filtro disminuirá la frecuencia de entrada por un factor de dos. El bloque para la división de frecuencia en un filtro comb no recursivo se muestra en la Fig. 4.1.

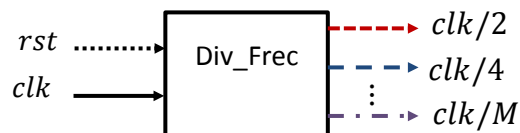


Fig. 4.1. Bloque para la implementación del divisor de frecuencias.

Como se observa, este bloque tiene como entrada la frecuencia inicial del sistema que es clk . A su salida mostrará $p - 1$ señales con frecuencias dadas por:

$$f_i = \frac{clk}{2^{i+1}}, \quad (4.1)$$

donde i es el número de etapas del filtro comb no recursivo con $i = 0, 1, \dots, p - 1$.

Este divisor se implementa por medio de un contador. Cada una de las salidas de entrada será activada cuando el contador alcance el valor que le corresponde. La forma general de describir este bloque en VHDL se muestra en la Fig. 4.2.

```

DIVISOR: process(clk,rst)
  variable count: integer range 0 to M:=0;
  begin

    if rst='1' then
      count := 0; aux2<= '0';           --Reinicio de las señales auxiliares
      aux4<= '0'; ... .. ; auxM <= '0' -- y del contador.

    elsif rising_edge(clk) then

      count := count+1;                 --Incremento del valor del contador

      if (count=2 or count=4 or count=6 or ... count =M ) then
        aux2 <='1';
      else
        aux2 <='0';                     --División clk/2
      end if;

      if (count=4 or count=8 or count=12 or ... count=M) then
        aux4 <='1';
      else
        aux4 <='0';                     --División clk/4
      end if;
      :
      :

      if (count=M) then
        auxM <='1';
      else
        auxM <='0';                     --División clk/M
      end if;

      if (count=M) then
        count := 0;                      --El contador llega a su valor máximo
      end if;                             -- y se reinicia.

    end if;
  end process DIVISOR;

  clk2 <= aux2;                          --Asignación de las señales de salida con
  clk4 <= aux4;                          --la señal auxiliar correspondiente.
  :
  clkM <= auxM;

```

Fig. 4.2. Descripción en VHDL de un divisor de frecuencia para un filtro comb no recursivo.

En la Fig. 4.3 se muestra el diagrama de tiempos del bloque divisor de frecuencia, para un filtro comb no recursivo con $M = 16$. Este fue obtenido a partir de simular el código de la Fig. 4.2 en Active HDL®. El código fue previamente adaptado para el valor de $M = 16$.

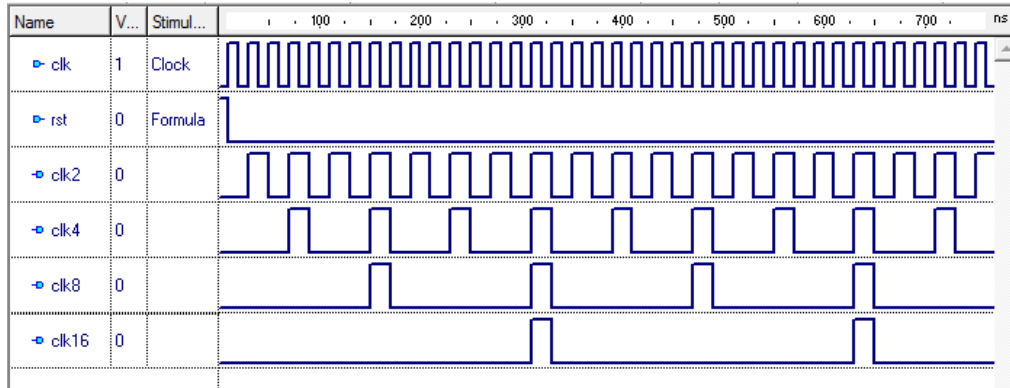


Fig. 4.3. Diagrama de tiempos del divisor de frecuencia con $M = 16$.

4.3. Descripción de un Filtro Comb No Recursivo con $M = 2^p$

La estructura del filtro decimador propuesto en el capítulo anterior, tiene sus bases en el filtro comb no recursivo. Por lo tanto, es necesario conocer en primer lugar, la forma en que este último es descrito en VHDL. En el siguiente apartado se dan a conocer los componentes de hardware que integran la estructura de un filtro comb no recursivo. También, se analizan las consideraciones que se tienen que hacer previo a su descripción en VHDL. Finalmente, se muestra la forma en que cada uno de sus componentes es interconectado.

4.3.1. Celda Básica en un Filtro Comb No Recursivo con $M = 2^p$

En la implementación de un filtro comb no recursivo con $M = 2^p$. La función de transferencia $1 + z^{-1}$ constituye la celda básica de su estructura. Los componentes de hardware requeridos en su construcción son un sumador y un registro. El diagrama a bloques de una celda se muestra en la Fig. 4.4.

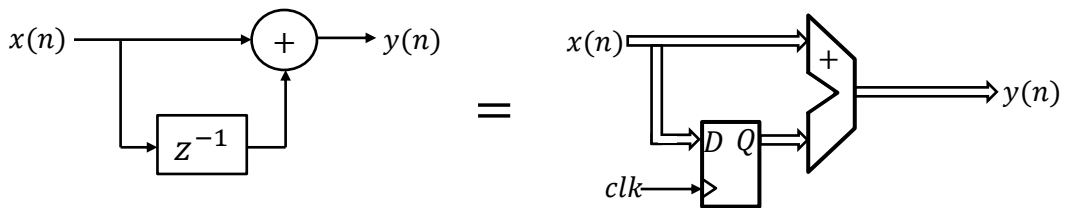


Fig. 4.4. Celda básica para la implementación de $1 + z^{-1}$.

Una etapa de un filtro comb no recursivo se obtiene mediante la conexión en cascada de K celdas básicas. Esto es ilustrado en la Fig.4.5.

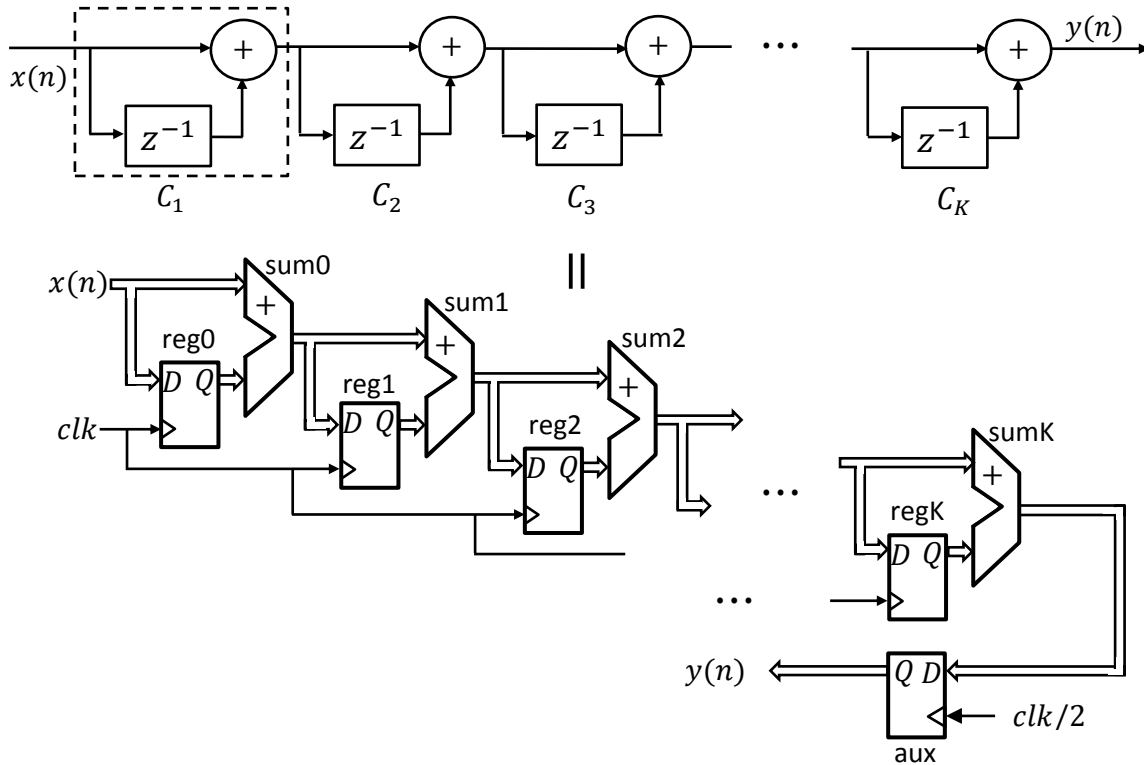


Fig. 4.5. Diagrama a bloques de la implementación de una etapa en un filtro comb no recursivo.

De esta forma, conectando p etapas como se muestra en la Fig. 4.6, se construye la arquitectura general de un filtro comb no recursivo con $M = 2^p$.

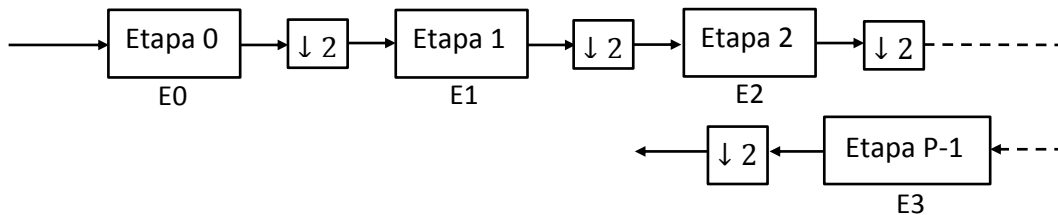


Fig. 4.6. Arquitectura general de un filtro comb no recursivo con $M = 2^p$.

4.3.2. Consideraciones Previas a la Descripción en VHDL del Filtro Comb No Recursivo

4.3.2.1. Incremento en el Tamaño del Dato de Entrada

En el apartado anterior se dio a conocer la estructura general de un filtro comb no recursivo. Sin embargo, antes de describir sus componentes de hardware en VHDL, es necesario hacer ciertas consideraciones. En primer lugar, se debe conocer el tamaño de palabra requerido por cada celda que forma parte de la estructura del filtro. Esto con la finalidad procesar adecuadamente la señal digital de entrada. Básicamente, el tamaño del dato de entrada

tendrá que incrementarse para poder representar el máximo valor que pueda registrarse en los sumadores. Por ejemplo, si se considera un dato de entrada de b bits en la primera celda de la p -ésima etapa. El tamaño del dato de salida de esta celda será de $b + 1$ bits. De esta forma, la segunda celda tendrá un dato de entrada de $b + 1$ bits y a su salida un dato de $b + 2$ bits. Generalizando, cada celda incrementara en un bit el tamaño del dato de entrada. Lo anteriormente dicho se ilustra en la Fig. 4.7.

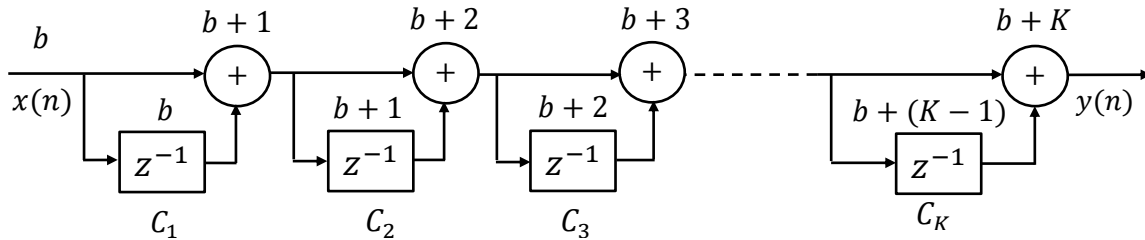


Fig. 4.7. Incremento en el tamaño del dato de entrada en cada celda.

Por lo tanto, si una etapa está conformada por K celdas básicas. El tamaño del dato de salida será:

$$b_{out} = b_{in} + K. \tag{4.1}$$

Finalmente, un filtro comb no recursivo está compuesto por p etapas. Entonces, la longitud máxima del dato de salida es:

$$b_{out} = b_{in} + pK. \tag{4.2}$$

En el siguiente ejemplo se muestra como calcular el tamaño del dato de salida de un filtro comb no recursivo.

Ejemplo 4.1. Se implementa un filtro comb no recursivo con $M = 32$ y $K = 5$. En términos de hardware se requieren cinco etapas en su construcción. Cada una de ellas formada por cinco celdas básicas conectadas en cascada. Considerando un dato de entrada de un bit y usando (4.1). Las longitudes necesarias en cada etapa se ilustran en la Fig. 4.8.

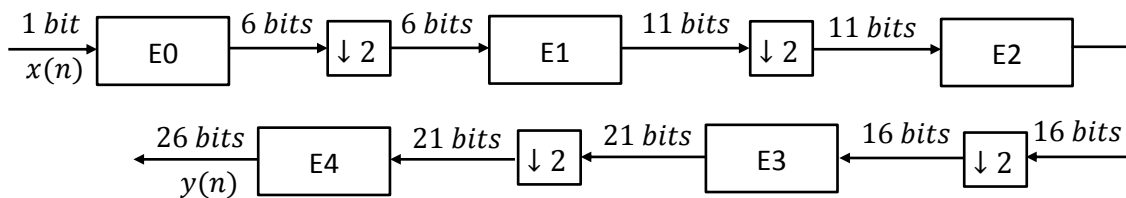


Fig. 4.8. Estructura del filtro comb no recursivo del Ejemplo 4.1.

En la Tabla 4.1 se incluyen los valores de longitud del dato de entrada y salida de cada celda básica.

Etapa ↓ Celda →	Tamaño del dato de entrada (bits)					Tamaño del dato de salida (bits)				
	C ₁	C ₂	C ₃	C ₄	C ₅	C ₁	C ₂	C ₃	C ₄	C ₅
E0	1	2	3	4	5	2	3	4	5	6
E1	6	7	8	9	10	7	8	9	10	11
E2	11	12	13	14	15	12	13	14	15	16
E3	16	17	18	19	20	17	18	19	20	21
E4	21	22	23	24	25	22	23	24	25	26

Tabla 4.1. Crecimiento del dato de entrada en cada celda.

4.3.2.2. Frecuencias de Trabajo

Hasta el momento se ha analizado como incrementar la longitud del dato de entrada. Sin embargo, también es de suma importancia asignar a cada etapa su correspondiente frecuencia de trabajo. Por lo tanto, retomando el bloque divisor de frecuencia descrito en el apartado 4.2. La estructura general para la implementación de un filtro comb no recursivo con $M = 2^p$ es:

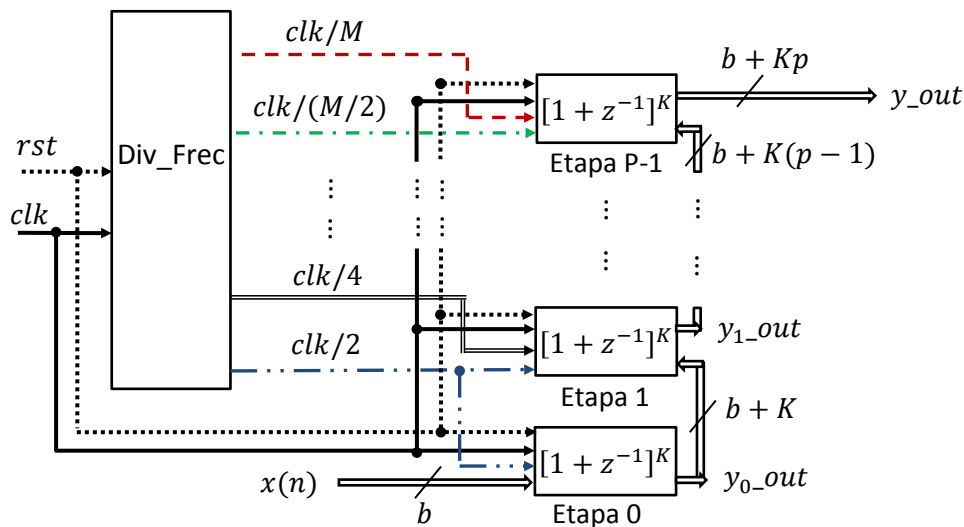


Fig. 4.9. Estructura para la implementación de un filtro comb no recursivo.

Como se observa en la Fig. 4.9, el filtro comb no recursivo tiene una estructura regular. Sólo el bloque correspondiente a la primera etapa, es diferente a los bloques de las demás etapas del filtro. A continuación se describe su funcionamiento.

4.3.2.3. Bloque para la Implementación de la Primera Etapa del Filtro Comb No Recursivo

En la Fig. 4.10 se muestra el diagrama del primer bloque.

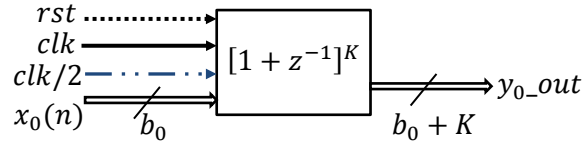


Fig. 4.10. Bloque para la implementación de la primera etapa del filtro comb no recursivo.

Se observa que posee cuatro señales de entrada y una de salida. La función de cada una de ellas es:

- **Señal *rst*:** Si *rst* toma un valor lógico de '1', todos los bits de la salida de los registros se colocarán en '0'. Esto permite regresar al sistema a sus condiciones iniciales.
- **Señal *clk*:** Su tarea consiste en sincronizar todos los registros del filtro comb no recursivo. Estos responderán a un flanco positivo de esta señal. Sólo en la primera etapa, también tiene la función de indicar la frecuencia de trabajo de cada celda básica.
- **Señal *clk/2*:** La frecuencia de esta señal es la mitad de la frecuencia de *clk*. Se trata de una señal de habilitación. Al tomar un valor lógico de '1', se mostrará un valor en la salida del primer bloque. Es decir, disminuye la frecuencia de la señal de entrada por un factor de dos.
- **Señal *x(n)*:** Es el valor de la señal digital de entrada al filtro. Se trata de un dato de *b* bits.
- **Señal *y₀_out*:** Es el valor de la señal digital de salida de la primera etapa. Consiste en un dato de *b + K* bits.

4.3.2.4. Bloque para la Implementación de las Etapas Restantes del Filtro Comb No Recursivo

A diferencia del bloque anterior. Las demás etapas se implementan como se muestra en la Fig. 4.11.

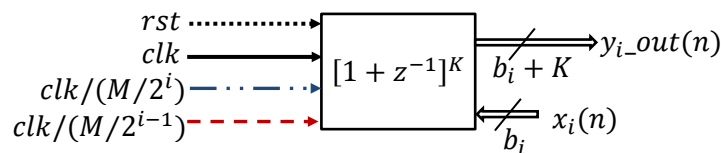


Fig. 4.11. Bloque para la implementación de las etapas restantes del filtro comb no recursivo.

Las señales *clk* y *rst* tienen la misma función que se describió en el apartado anterior. Los datos de entrada y salida de cada una de las etapas son $x_i(n)$ y $y_i_out(n)$, respectivamente. A diferencia del bloque de la primera etapa, este posee dos señales de habilitación:

- **Señal $clk/(M/2^i)$:** Indica la frecuencia de trabajo de la etapa i . Es decir, la velocidad con la cual se realiza el procesamiento de la señal digital de entrada. La frecuencia de esta señal, es el resultado de dividir el valor de frecuencia del reloj clk entre $M/2^i$.
- **Señal $clk/(M/2^{i+1})$:** Es la frecuencia con la cual la etapa i mostrará un dato a su salida. Esto implica una reducción de la frecuencia de la señal de entrada por un factor de dos.

En la siguiente sección se indica cómo realizar la descripción en VHDL de estos bloques.

4.3.3. Descripción en VHDL del Filtro Comb No Recursivo

En los apartados anteriores se mostró la estructura general para la implementación de un filtro comb no recursivo con $M = 2^p$. Se realizaron las consideraciones necesarias para garantizar su correcto funcionamiento. Se presentó cada uno de los componentes que lo integran, y se explicó la función que tienen. En este apartado se describirá cada uno de ellos en VHDL.

4.3.3.1. Descripción de la Primera Etapa del Filtro Comb No Recursivo

La descripción de este bloque se lleva a cabo por medio de dos procesos. El primero de ellos tiene la tarea de realizar el procesamiento de la señal digital de entrada. Su descripción en VHDL se muestra en la Fig. 4.12.

```

Etapa_0:process(clk,rst)
  begin
    if rst='1' then
      reg0 <= (others=>'0');      --Se reinicia el valor de todos los
      reg1 <= (others=>'0');      --registros de la primera etapa
      :
      regK <= (others=>'0');
    elsif rising_edge(clk) then
      reg0 <= xn1;               --Implementación de las K celdas
      suma1 <= reg0+xn1;         --básicas
      reg1 <= suma1;
      suma2 <= reg1+suma1;
      :
      sumaK <=regK_sumaK_1;      --K_1=K-1
    end if;
  end process Etapa_0;

```

Fig. 4.12. Descripción en VHDL del primer proceso de la Etapa 0.

El segundo proceso reduce la frecuencia de la señal digital de entrada a la mitad. En la Fig. 4.13 se ilustra la forma de realizar su descripción en VHDL.

```

Salida0:process(clk,clk_2)
begin
    if rising_edge(clk) then
        if clk_2='1' then --Señal de habilitación del div_frec clk_2=clk/2
            aux1 <=sumaK;
            end if;
        end if;
    end process Salida0;
    y_out0<=aux0; --Señal de salida de la primera etapa

```

Fig. 4.13. Descripción en VHDL del segundo proceso de la Etapa 0.

Ambos procesos se incluyen dentro de una misma arquitectura en VHDL.

4.3.3.2. Descripción de las p-1 Etapas Restantes del Filtro Comb No Recursivo

Las etapas restantes del filtro comb no recursivo realizan el procesamiento de la señal digital de entrada, con una frecuencia menor a la frecuencia inicial. Su descripción en VHDL se muestra en la Fig. 4.14.

```

Etapa_i:process(clk,clki,rst)
begin --i=N° Etapa
    if rst='1' then
        reg0i <= (others=>'0'); --Se reinicia el valor de todos los
        reg1i <= (others=>'0'); --registros de la i-ésima etapa
        :
        regKi <= (others=>'0');
    elsif rising_edge(clk) then
        if clki='1' then --Frecuencia de trabajo de la i-ésima etapa
            reg0i <= xni; --Implementación de las K celdas básicas
            suma1i <= reg0i+xni; --de la i-ésima etapa
            reg1i <= suma1i;
            suma2i <= reg1i+suma1i;
            reg2i <= suma2i;
            :
            sumaKi <= regKi_1+sumaKi_1; --Ki_1=Ki-1
        end if;
    end if;
end process Etapa_i;

```

Fig. 4.14. Descripción en VHDL del primer proceso de la Etapa i.

El segundo proceso tiene como tarea principal reducir a la mitad la frecuencia de la señal de entrada. Este es descrito en VHDL como se ilustra en la Fig. 4.15.

```

Salida_i:process(clk,clki_1)
begin
  if rising_edge(clk) then
    if clki_1='1' then
      auxi <= sumaKi;
    end if;
  end if;
end process Salida_i;
y_outi<=auxi;
    
```

Fig. 4.15. Descripción en VHDL del segundo proceso de la Etapa i.

Estos dos procesos deben de incluirse dentro de una mista arquitectura en VHDL.

4.4. Estructura General para la Implementación de un Filtro Decimador Diseñado con el Método Propuesto

La estructura del filtro decimador propuesto, tiene sus bases en el filtro comb no recursivo. En esta se ha agregado un filtro compensador, y de uno a tres filtros coseno rotado para mejorar su respuesta en magnitud. Los coeficientes de ambos filtros pueden ser implementados a base de sumas y corrimientos. En la Fig. 4.16 se muestra la estructura necesaria para la implementación en FPGA del filtro propuesto para $R = 3$.

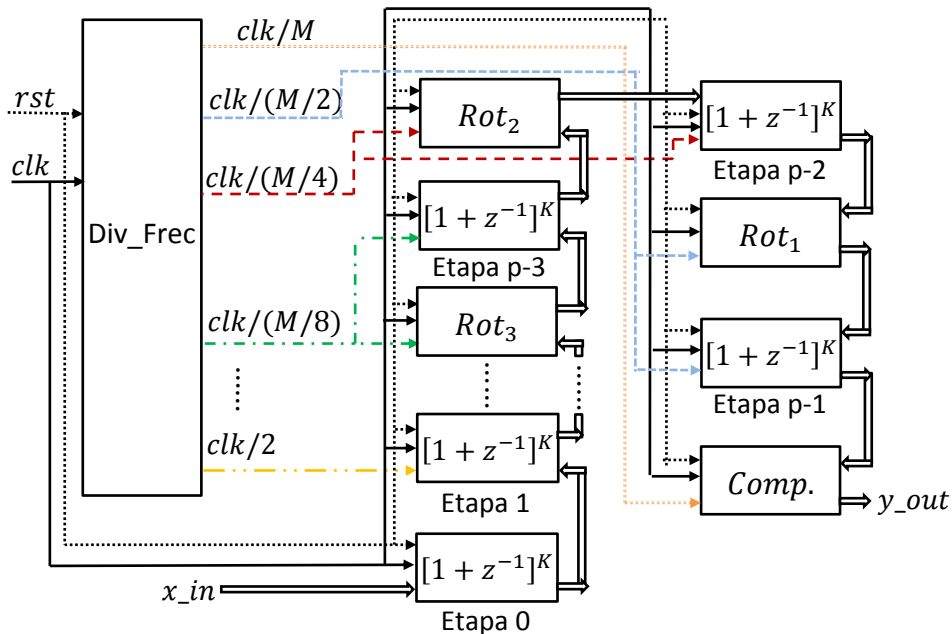


Figura 4.16. Estructura para la implementación en FPGA del filtro propuesto ($R = 3$).

Esta misma estructura puede ser empleada para un filtro con $R = 1$ ó con $R = 2$. En el caso de un decimador al que sólo se la ha aplicado un filtro coseno rotado, los bloques Rot_2 y Rot_3 son descartados. De esta forma, las etapas 0 a $p - 2$ son conectadas en cascada respetando

sus respectivas frecuencias de trabajo. Lo mismo sucede para un filtro decimador con $R = 2$. En este sólo el bloque Rot_3 es descartado. A continuación se describe como hacer el diseño de un filtro coseno rotado y un filtro compensador en VHDL. También se analizan las consideraciones que se tienen que hacer previas a su implementación.

4.4.1. Descripción del Filtro Coseno Rotado

El diagrama de bloque y la estructura para la implementación del filtro coseno rotado se muestra en la Fig. 4.17. Este requiere dos sumadores y un multiplicador. Sin embargo, si el coeficiente β es redondeado, este último es descartado. De esta forma, sólo se necesitarán sumadores extras y registros de corrimiento para la implementación del coeficiente β_r .

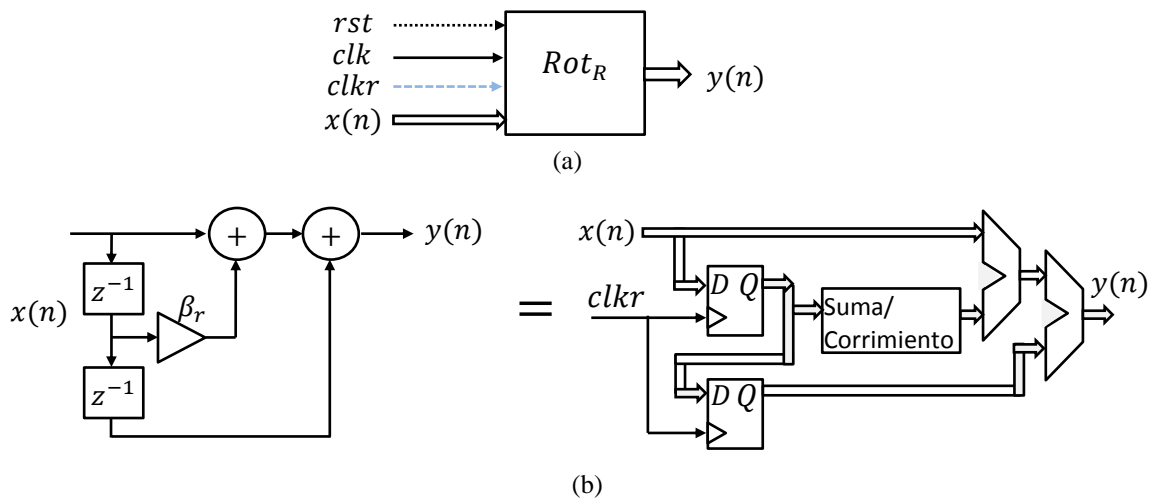


Fig. 4.17. (a) Bloque y (b) estructura para la implementación del filtro coseno rotado.

Las señales de entrada clk y rst tiene la misma función que en los bloques de las etapas del filtro comb no recursivo. Las señales $x(n)$ y $y(n)$ son los datos de entrada y salida del filtro coseno rotado respectivamente.

El dato de entrada tendrá que incrementar su longitud al ser procesado por el filtro coseno. Por lo tanto, se tiene un dato de salida cuya longitud máxima dependerá la cantidad de sumas que tengan que realizarse. Este número de sumas incluye las requeridas para representar el coeficiente β_r , y las dos propias del filtro coseno.

La frecuencia de trabajo de este filtro (señal $clkr$), depende de la posición que ocupe dentro de la estructura del filtro comb (véase Fig. 4.16). En el siguiente ejemplo se muestra como realizar su descripción en VHDL.

Ejemplo 4.2. Se diseña un filtro comb con $M = 16$ y $K = 3$. Un filtro coseno rotado es agregado a su estructura. Este es colocado antes de la última etapa del filtro comb. Por lo tanto, su frecuencia de trabajo es $clk/8$. Considerando que inicialmente se tiene un dato de

entrada de dos bits. El tamaño del dato a procesar por el filtro coseno rotado es de 11 bits. El valor de β_r se calcula con la expresión 3.20, obteniendo como resultado:

$$\beta_r = 2^0 + 2^{-1} - 2^{-4} - 2^{-6}. \quad (4.3)$$

La implementación de β_r requiere de un sumador, de dos restadores y de tres registros de corrimiento. El primero de ellos aplica un corrimiento de un bit a la derecha sobre el dato de entrada. El segundo efectúa un corrimiento de cuatro bits a la derecha. Finalmente, el tercer registro recorre el dato de entrada seis bits a la derecha. El resultado de todos los registros se suma a la señal de entrada.

Por último, conociendo todo lo anterior es posible determinar que el tamaño del dato de salida tendrá dos bits más en comparación con el dato de entrada. La descripción en VHDL de este filtro se muestra en la Fig. 4.18.

```

rot:process(clk,clkr,rst)
begin
  if rst='1' then
    z0<=(others=>'0'); z1<=(others=>'0'); z2<=(others=>'0');
    aux1<=(others=>'0'); aux2<=(others=>'0'); aux3<=(others=>'0');
    aux4<=(others=>'0'); aux5<=(others=>'0'); aux6<=(others=>'0');
    aux7<=(others=>'0'); aux8<=(others=>'0'); aux9<=(others=>'0');
    aux10<=(others=>'0');           --Reinicio de las señales auxiliares
  elsif rising_edge(clk) then
    if clkr='1' then
      z0 <= xn;           --Señal actual
      z1 <= z0;          --Primer retraso
      z2 <= z1;          --Segundo retraso

      aux1<= z1+(z1(11)&'0'&z1( 10 downto 1)); --Implementación de  $B_r$ 
      aux2<= (z1(11)&"0000"&z1(10 downto 4))+(z1(11)&"000000"&z1(10 downto 6));
      aux3<=z0;
      aux4<=z2;

      aux5<=aux3+aux1;
      aux6<=aux2;           --Sumas efectuadas por el filtro
      aux7<=aux4;           --coseno rotado

      aux8<=aux5-aux6;
      aux9<=aux7;
      aux10<=aux8+aux9;
    end if;
  end if;
  y_r<=aux10;
end process rot;

```

Fig. 4.18. Descripción en VHDL del filtro coseno rotado del Ejemplo 4.2.

Las señales auxiliares en el código anterior, son utilizadas para almacenar los resultados previos al resultado final. De esta forma, se evita la pérdida de información durante el procesamiento de la señal.

4.4.2. Descripción del Filtro Compensador

El diagrama de bloque y la estructura necesaria para la implementación del filtro compensador se ilustra en la Fig. 4.19. Inicialmente, este requiere tres sumadores y un multiplicador. Sin embargo, el coeficiente B es redondeado. De esta forma, sólo se necesitaran sumadores extras y registros de corrimiento para su representación.

Las señales clk , rst , $x(n)$ y $y(n)$ tienen la misma función que en los bloques anteriores. El dato de entrada incrementará su longitud al ser procesado por el filtro compensador. La longitud del dato de salida depende de la cantidad de sumas efectuadas por este filtro.

La frecuencia de trabajo del compensador es clk/M (señal clk_c). En el siguiente ejemplo se muestra como realizar su descripción en VHDL.

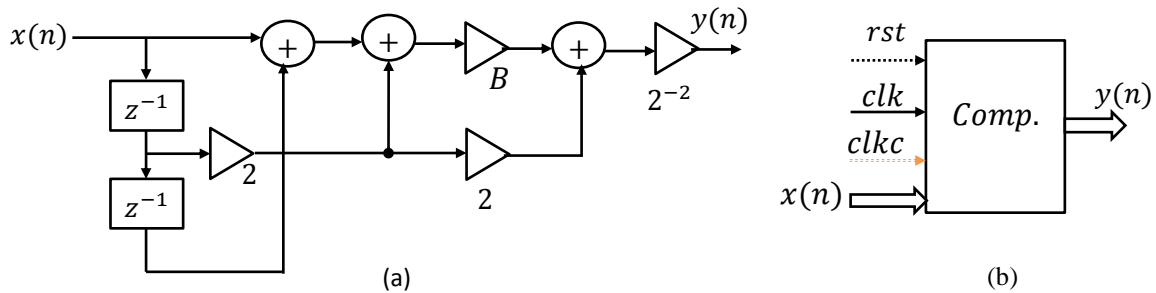


Fig. 4.19. (a) Estructura y (b) bloque para la implementación del filtro compensador.

Ejemplo 4.3. Se diseña un filtro compensador para el filtro del ejemplo 4.2. De acuerdo a la Tabla 3.4 el valor de $B = 1.30$.

$$B_r = 2^0 + 2^{-2} + 2^{-5} + 2^{-6}. \tag{4.4}$$

La implementación de B_r requiere de tres sumadores y de tres registros de corrimiento. Estos aplican un corrimiento de dos, cinco y seis bits a la derecha sobre el dato de entrada, respectivamente. El resultado obtenido se suma a la señal de entrada sin modificar. El dato de salida de la última etapa del filtro del ejemplo 4.2 tiene una longitud de 14 bits

Finalmente, conociendo todo lo anterior es posible determinar que el tamaño del dato de salida tendrá cinco bits más en comparación con el dato de entrada. La descripción en VHDL de este filtro se muestra en la Fig. 4.20.

```

compensador:process(clk,clk,rst)
begin
    if (rst='1') then
        z1<=(others=>'0'); z2<= (others=>'0');
        aux1<= (others=>'0'); aux2<=(others=>'0');
        aux3<= (others=>'0'); aux4<=(others=>'0');
        aux5<= (others=>'0'); aux6<=(others=>'0');
        aux7<= (others=>'0'); aux8<=(others=>'0');
        aux9<= (others=>'0'); aux10<=(others=>'0');

        elsif rising_edge(clk) then
            if clk='1' then

z1<= sxtxg; --Muestras actual y pasadas de la señal de entrada
z2 <=z1;

--Término 1
aux1<=sxtxg+(sxtxg(18)&("00"&sxtxg(17 downto 2)));
aux2<=(sxtxg(27)&("00000"&sxtxg(27 downto 5)))+(sxtxg(28)&("000000"&sxtxg(27 downto 6)));
--Término 2
aux3<=(z1(25 downto 0)&"000")-z1;
aux4<=(z1(28)&("00"&z1(27 downto 2)))+(z1(28)&("00000"&z1(27 downto 5)));
--Término 3
aux6<=(z2(27 downto 0)&'0')-(z2(28)&("00"&z2(27 downto 2)));
aux7<=(z2(28)&("0000"&z2(27 downto 4)))+(z2(28)&("00000"&z2(27 downto 5)));

aux8<=aux1-aux2;
aux9<=aux3+aux4;
aux10<=aux6-aux7;

aux11<=aux9-aux8;
aux12<=aux10;

aux13<=aux11-aux12;

            end if;
        end if;
    end process compensador;
    y_g<=aux13;

```

Fig. 4.20. Descripción en VHDL del filtro compensador del ejemplo 4.3.

Al igual que en la implementación del filtro coseno rotado, las señales auxiliares son utilizadas para almacenar los resultados previos al resultado final.

Capítulo 5

Implementación y Simulación de los Filtros Decimadores Propuestos

En este capítulo se describe como se realizó la implementación en FPGA de los filtros decimadores propuestos. Su correcto funcionamiento se comprueba analizando su respuesta al impulso y al escalón. Los resultados obtenidos en las simulaciones de su comportamiento son presentados. Finalmente, se enlistan los recursos de hardware requeridos en sus estructuras.

5.1. Introducción

En el capítulo anterior se mostró como describir en VHDL un filtro comb no recursivo. También se ilustra que al realizar ciertas modificaciones en su estructura, es posible obtener filtros decimadores como los propuestos en el capítulo tres.

En las siguientes secciones se presenta el diseño e implementación de cuatro filtros decimadores. Estos tienen un mismo factor de decimación y orden. El primero de ellos es un filtro comb no recursivo convencional. Los tres restantes son filtros decimadores obtenidos a partir de aplicar el método de diseño propuesto. El adecuado funcionamiento de cada uno de ellos se comprueba al analizar de su respuesta al impulso y al escalón. Este análisis se hace mediante simulaciones en ACTIVE HDL[®]. Los datos obtenidos son contrastados con los calculados previamente en MATLAB[®].

Posteriormente, cada filtro es sintetizado e implementado en la tarjeta NEXYS 4 que cuenta con el FPGA ARTIX-7 XCA100T CSG324AC. Esto se hace utilizando las herramientas incluidas en ISE Design Suite 14.7, que es proporcionado por XILINX[®]. Finalmente, se enlistan los recursos hardware necesarios en cada estructura, y se hace una comparación entre la cantidad de sumadores y registros requeridos por cada filtro. Esto con el fin de medir el incremento en complejidad de los filtros decimadores propuestos, en relación con un filtro al que no se le ha hecho ninguna modificación.

5.2. Implementación de un Filtro Comb No Recursivo con $M = 32$ y $K = 4$

En la Fig. 5.1 se muestra el diagrama a bloques del filtro comb no recursivo que fue implementado. Este tiene como factor de decimación $M = 32 = 2^5$ y su orden es $K = 4$.

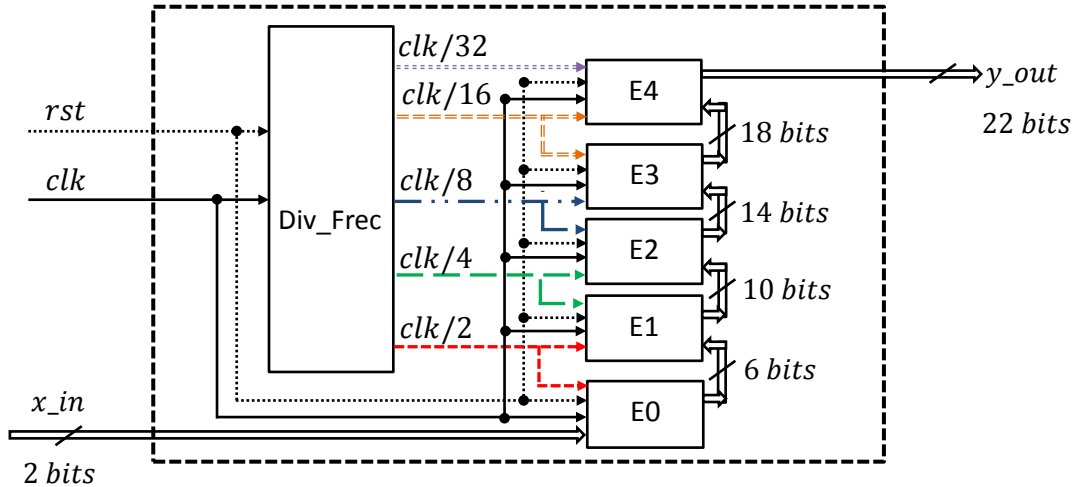


Fig. 5.1. Diagrama a bloques del filtro comb no recursivo con $M = 32$ y $K = 4$.

Se observa que el divisor de frecuencia tiene que proporcionar cinco señales de salida. Cada una de ellas habilitará el bloque que le corresponde, de acuerdo a la frecuencia de trabajo que requiere. Estos bloques están numerados de $E0$ hasta $E4$, y corresponden a las etapas del filtro comb no recursivo. La descripción en VHDL tanto del divisor de frecuencia como de las etapas del filtro comb, se hicieron conforme a lo indicado es las secciones 4.2 y 4.3 del capítulo cuatro.

En la Tabla 5.1 se incluye el valor de longitud de palabra necesaria en cada bloque. También se muestran sus correspondientes frecuencias de trabajo.

Bloque	Frecuencia de entrada	Frecuencia de salida	Longitud de la palabra de entrada (bits)	Longitud de la palabra de salida (bits)
E0	clk	clk/2	2	6
E1	clk/2	clk/4	6	10
E2	clk/4	clk/8	10	14
E3	clk/8	clk/16	14	18
E4	clk/16	clk/32	18	22

Tabla 5.1. Consideraciones en el diseño del filtro comb no recursivo.

Una forma de comprobar que el filtro fue descrito adecuadamente en VHDL, consiste en obtener su respuesta al impulso y al escalón. Esto se consiguió realizando simulaciones en ACTIVE HDL®. En las siguientes secciones se presenta la forma en que estas simulaciones se llevaron a cabo. También se muestran los resultados obtenidos.

5.2.1. Respuesta al Impulso del Filtro Comb No Recursivo con $M = 32$ y $K = 4$

La respuesta al impulso de un sistema lineal se obtiene si su entrada es estimulada con una señal impulso unitario. En el caso de un Filtro FIR, esta respuesta coincide con los coeficientes presentes en su función de transferencia.

Particularmente, la respuesta al impulso de un filtro decimador, se compone de los coeficientes que acompañan a las muestras que son múltiplos enteros de M . Esta se obtiene si todas sus etapas están perfectamente sincronizadas. Por lo tanto, se requiere estimular adecuadamente las señales de entrada del filtro mostrado en la Fig. 5.1.

La tarjeta NEXYS 4 cuenta con un reloj de 100MHz. Esta frecuencia fue asignada a la señal de reloj clk . De esta forma, conociendo su valor fue posible determinar como estimular las señales restantes. A continuación se describe este proceso.

- Al iniciar la simulación se indica que el valor de x_{in} es "00". La señal rst tiene que tener un valor lógico de '1'. Esto permite que el filtro permanezca en condiciones iniciales.
- En el momento en que rst cambie su valor a '0', será necesario esperar cuatro ciclos del reloj clk antes de activar la señal x_{in} .
- En el quinto flanco positivo de clk , la señal x_{in} cambiará su valor de "00" a "01" y permanecerá así durante un ciclo completo. Al finalizar este tiempo, la señal x_{in} volverá a su valor inicial.

En la Tabla 5.2 se incluyen las indicaciones hechas en ACTIVE HDL® para estimular las señales de entrada del filtro.

Señal	Estímulo Aplicado
clk	100 MHz
rst	1 0ns, 0 5ns
x_{in}	00 0 ns, 01 55ns, 00 65ns

Tabla 5.2. Estímulos aplicados a las señales de entrada del filtro comb no recursivo para obtener su respuesta al impulso.

En la Fig. 5.2 se muestra la respuesta al impulso del filtro comb descrito en el apartado anterior. La salida y_{out} tiene una longitud de palabra de 22 bits. Esta representa los coeficientes $z^0, z^{-32}, z^{-64}, \dots etc$, de la función de transferencia del filtro.

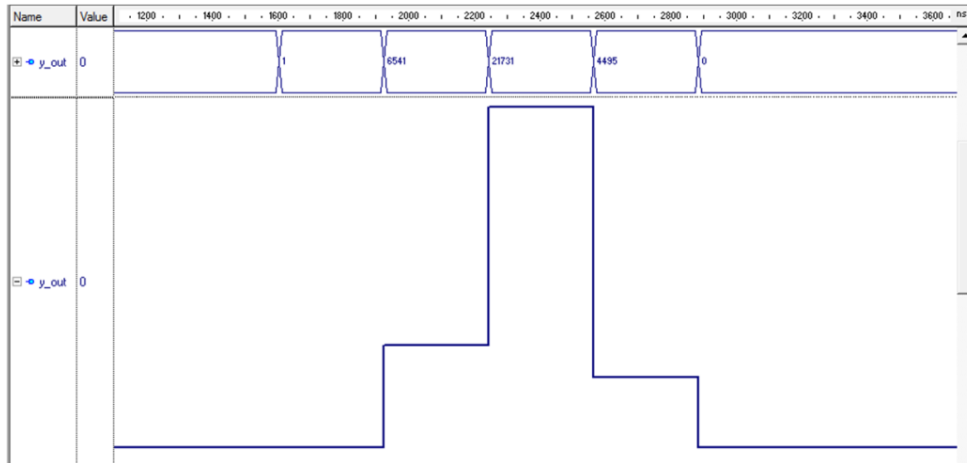


Fig. 5.2. Respuesta al impulso del filtro comb no recursivo ($M = 32, K = 4$) obtenida en ACTIVE VHDL®.

La respuesta al impulso del filtro comb también se calculó utilizando MATLAB®. En la Fig. 5.3 se ilustran los valores obtenidos.

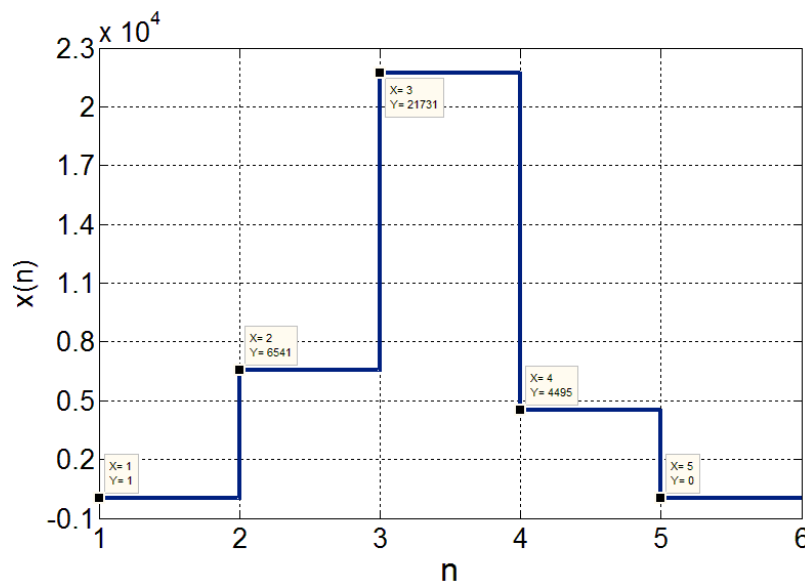


Fig. 5.3. Respuesta al impulso del filtro comb no recursivo ($M = 32, K = 4$) obtenida en MATLAB®.

La Tabla 5.3 incluyen los resultados de las simulaciones hechas en ACTIVE HDL® y MATLAB®. Se observa que los valores calculados en ambos casos son iguales.

MATLAB®	ACTIVE HDL®
1	1
6541	6541
21731	21731
4495	4495

Tabla 5.3. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro comb no recursivo).

5.2.2. Respuesta al Escalón del Filtro Comb No Recursivo con $M = 32$ y $K = 4$

La respuesta al escalón del filtro comb se obtiene de forma similar que la respuesta al impulso. La única diferencia es que la señal x_{in} mantiene el valor "01" una vez que ha sido activada. En la Tabla 5.4 se indica como estimular las señales de entrada del filtro en ACTIVE HDL®.

Señal	Estímulo Aplicado
clk	100 MHz
rst	1 0ns, 0 5ns
x_{in}	00 0 ns, 01 55ns

Tabla 5.4. Estímulos aplicados a las señales de entrada del filtro comb no recursivo para obtener su respuesta al escalón.

La respuesta obtenida en la simulación se muestra en la Fig. 5.4.

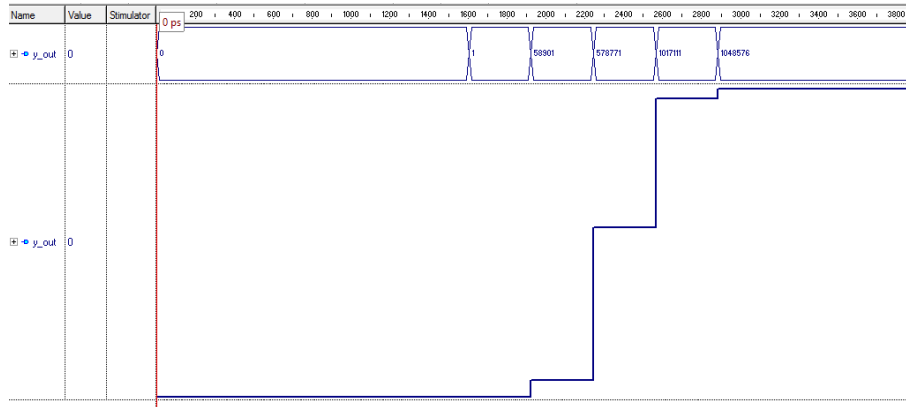


Fig. 5.4. Respuesta al escalón del filtro comb no recursivo ($M = 32, K = 4$) obtenida en ACTIVE HDL®.

La respuesta calculada en MATLAB® es presentada en la Fig. 5.5.

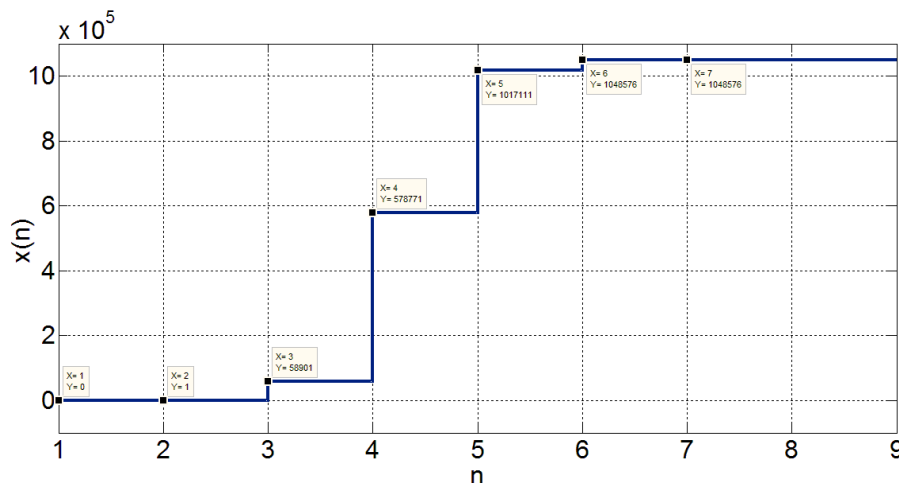


Fig. 5.5. Respuesta al escalón del filtro comb no recursivo ($M = 32, K = 4$) obtenida en MATLAB®.

Finalmente, los valores numéricos obtenidos en ambas simulaciones se incluyen en la Tabla 5.5.

MATLAB®	ACTIVE HDL®
1	1
58901	58901
578771	578771
1017111	1017111
1048576	1048576

Tabla 5.5. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro comb no recursivo).

5.2.3. Recursos Requeridos en la Implementación de un Filtro Comb No Recursivo con $M = 32$ y $K = 4$

Posterior a la verificación de su correcta descripción en VHDL. El filtro comb no recursivo fue sintetizado e implementado en la tarjeta NEXYS 4. La lista de los recursos requeridos en su estructura y la máxima frecuencia de operación se incluyen en las Tablas 5.6 y 5.7.

Resumen de Utilización de Recursos			
Utilización Lógica	Usado	Disponible	Utilización
Número de Registros	560	126800	1%
Número de pares de LUT-FF completos	283	394	71%
Número de LUTs	283	63400	1%
Número de IOBs	26	210	12%
Máxima Frecuencia de Operación	492.005MHz		

Tabla 5.6. Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro comb no recursivo.

Bloque	Sumadores/Restadores	Flip-Flops Tipo D	Multiplexores
Divisor de Frecuencia	1 (6 bits)	11	1
E0	4 (6 bits)	54	0
E1	4 (10 bits)	90	0
E2	4 (14 bits)	126	0
E3	4 (18 bits)	162	0
E4	4 (22 bits)	198	0
Total	21	641	1

Tabla 5.7. Recursos requeridos en la implementación del filtro comb no recursivo.

5.3. Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 1$

En la Fig. 5.6 se muestra el diagrama a bloques del filtro propuesto que fue implementado. Este tiene como parámetros de diseño $M = 32 = 2^5$, $K = 4$ y $R = 1$.

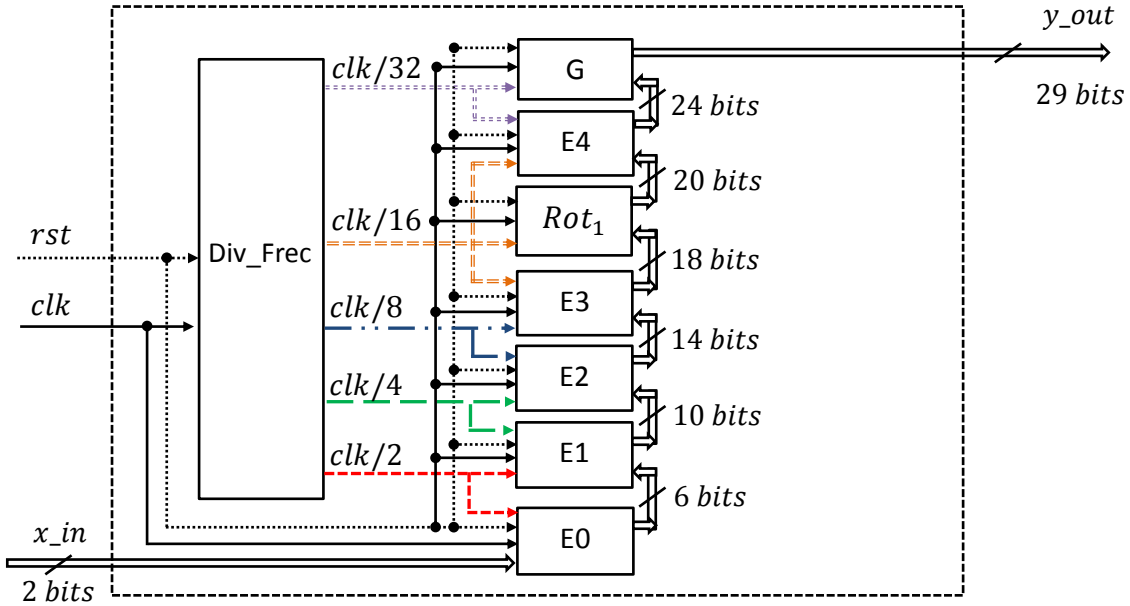


Fig. 5.6. Diagrama a bloques del filtro decimador propuesto con $M = 32, K = 4$ y $R = 1$.

La descripción en VHDL de cada bloque de la estructura se hizo de acuerdo a lo presentado en el capítulo cuatro. La Tabla 5.8 contiene los valores redondeados de los coeficientes del compensador y del filtro coseno rotado.

Filtro	Coefficiente	Implementación como sumas y corrimientos
Compensador G	$B_r = 1.65625$	$B_r = 2^1 - 2^{-2} - 2^{-4} - 2^{-5}$
Coseno Rotado Rot_1	$\beta_r = 1.421875$	$\beta_r = 2^0 + 2^{-1} - 2^{-4} - 2^{-6}$

Tabla 5.8. Coeficientes del compensador y del filtro coseno rotado.

En la Tabla 5.9 se muestran los valores de la longitud de palabra necesaria en cada bloque. También se enlista las correspondientes frecuencias de trabajo.

Bloque	Frecuencia de entrada	Frecuencia de salida	Longitud de la palabra de entrada (bits)	Longitud de la palabra de salida (bits)
E0	clk	clk/2	2	6
E1	clk/2	clk/4	6	10
E2	clk/4	clk/8	10	14
E3	clk/8	clk/16	14	18
Rot_1	clk/16	clk/16	18	20
E4	clk/16	clk/32	20	24
G	clk/32	clk/32	24	29

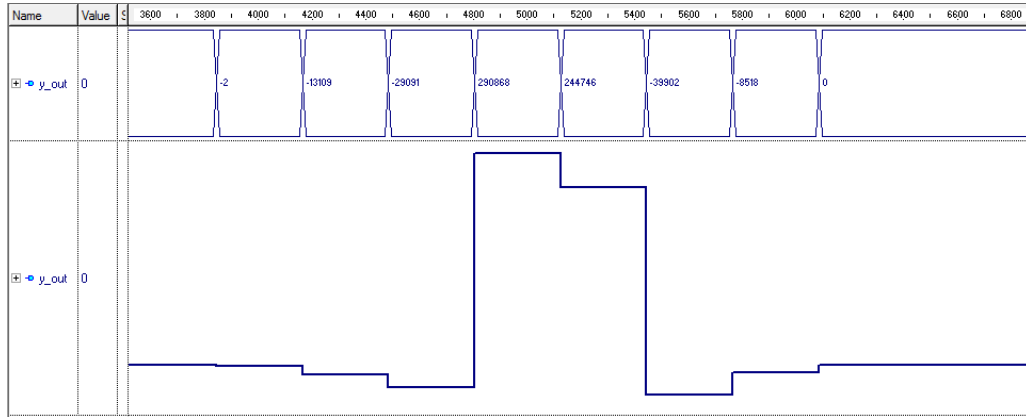
Tabla 5.9. Consideraciones en el diseño del filtro decimador propuesto ($M = 32, K = 4, R = 1$).

Hechas las consideraciones anteriores el comportamiento del filtro fue simulado.

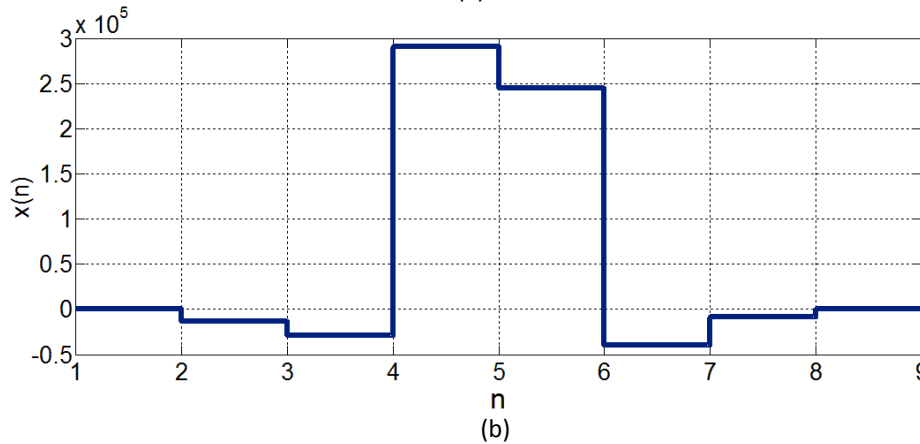
5.3.1. Respuesta al Impulso del Filtro Propuesto con $M = 32, K = 4$ y $R = 1$

La respuesta al impulso de este filtro se obtiene de la misma forma que se obtuvo la respuesta del filtro comb no recursivo. Por lo tanto, sus señales de entrada fueron estimuladas conforme a lo presentado en la Tabla 5.2.

Los resultados obtenidos en las simulaciones de ACTIVE HDL® y MATLAB® se ilustran en las Fig. 5.7a y 5.7b, respectivamente.



(a)



(b)

Fig. 5.7. Respuesta al impulso del filtro decimador propuesto ($M = 32, K = 4, R = 1$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.

Los valores numéricos calculados en ambas simulaciones se enlistan en la Tabla 5.10.

MATLAB®	ACTIVE HDL®
-2	-2
-13109	-13109
-29091	-29091
290868	290868
244746	244746
-39902	-39902
-8518	-8518

Tabla 5.10. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro propuesto con $M = 32, K = 4$ y $R = 1$).

5.3.2. Respuesta al Escalón del Filtro Propuesto con $M = 32, K = 4$ y $R = 1$

La respuesta al escalón de este filtro, se obtiene aplicando a las señales de entrada los estímulos utilizados en el filtro comb no recursivo (véase Tabla 5.4). Los resultados obtenidos en las simulaciones hechas en ACTIVE HDL® y MATLAB® se presentan en las Fig. 5.8a y 5.9b.

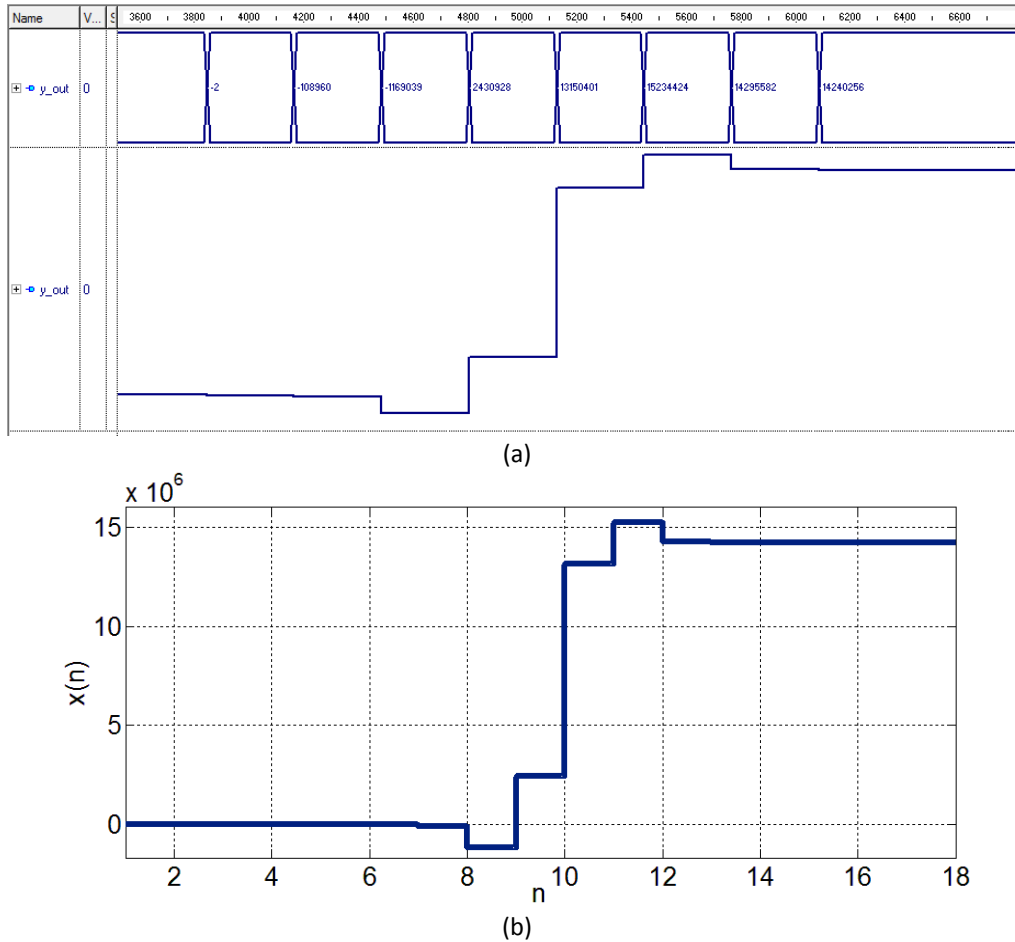


Fig. 5.8. Respuesta al escalón del filtro decimador propuesto ($M = 32, K = 4, R = 1$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.

La Tabla 5.11 incluye los valores numéricos de la respuesta al escalón del filtro decimador.

MATLAB®	ACTIVE HDL®
-2	-2
-108960	-108960
-1169039	-1169039
2430928	2430928
13150401	13150401
15234424	15234424
14295582	14295582
14240256	14240256

Tabla 5.11. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro propuesto con $M = 32, K = 4$ y $R = 1$).

5.3.3. Recursos Requeridos en la Implementación del Filtro Propuesto con $M = 32, K = 4$ y $R = 1$

El filtro propuesto fue sintetizado en ISE 14.7 e implementado en la tarjeta NEXYS 4. La lista de recursos que requiere y su máxima frecuencia de operación se incluyen en las Tablas 5.12 y 5.13.

Resumen de Utilización de Recursos			
Utilización Lógica	Usado	Disponible	Utilización
Número de Registros	1160	126800	1%
Número de pares de LUT-FF completos	692	999	69%
Número de LUTs	696	63400	1%
Número de IOBs	33	210	15%
Frecuencia Máxima	456.767 MHz		

Tabla 5.12. Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro propuesto con $M = 32, K = 4$ y $R = 1$.

Bloque	Sumadores/Restadores	Filp-Flops Tipo D	Multiplexores
Divisor de Frecuencia	1	11	1
E0	4	54	0
E1	4	90	0
E2	4	126	0
E3	4	162	0
E4	4	240	0
Filtro Coseno Rotado	5	260	0
Compensador	11	406	0
Total	37	1349	1

Tabla 5.13. Recursos requeridos en la implementación del filtro propuesto con $M = 32, K = 4$ y $R = 1$.

5.4. Implementación del Filtro Propuesto con $M = 32, K = 4$ y $R = 2$

La Fig. 5.9 muestra el diagrama a bloques del filtro decimador implementado. Este tiene el mismo factor de decimación y orden de los filtro anteriores. La diferencia en su estructura reside en la aplicación de dos filtros coseno rotado. Su descripción en VHDL se hizo teniendo en cuenta las siguientes consideraciones.

En primer lugar se calcularon los coeficientes del filtro compensador y de los filtros coseno rotado. Estos se muestran en la Tabla 5.14. También, se incluye la forma en que son expresados en potencias de dos para facilitar su implementación.

Filtro	Coficiente	Implementación como sumas y corrimientos
Compensador G	$B_r = 1.796875$	$B_r = 2^1 - 2^{-3} - 2^{-4} - 2^{-6}$
Coseno Rotado Rot ₁	$\beta_{r_1} = 1.421875$	$\beta_{r_1} = 2^0 + 2^{-1} - 2^{-4} - 2^{-6}$
Coseno Rotado Rot ₂	$\beta_{r_2} = 1.84375$	$\beta_{r_2} = 2^1 - 2^{-3} - 2^{-5}$

Tabla 5.14. Coeficientes del compensador y de los filtros coseno rotado.

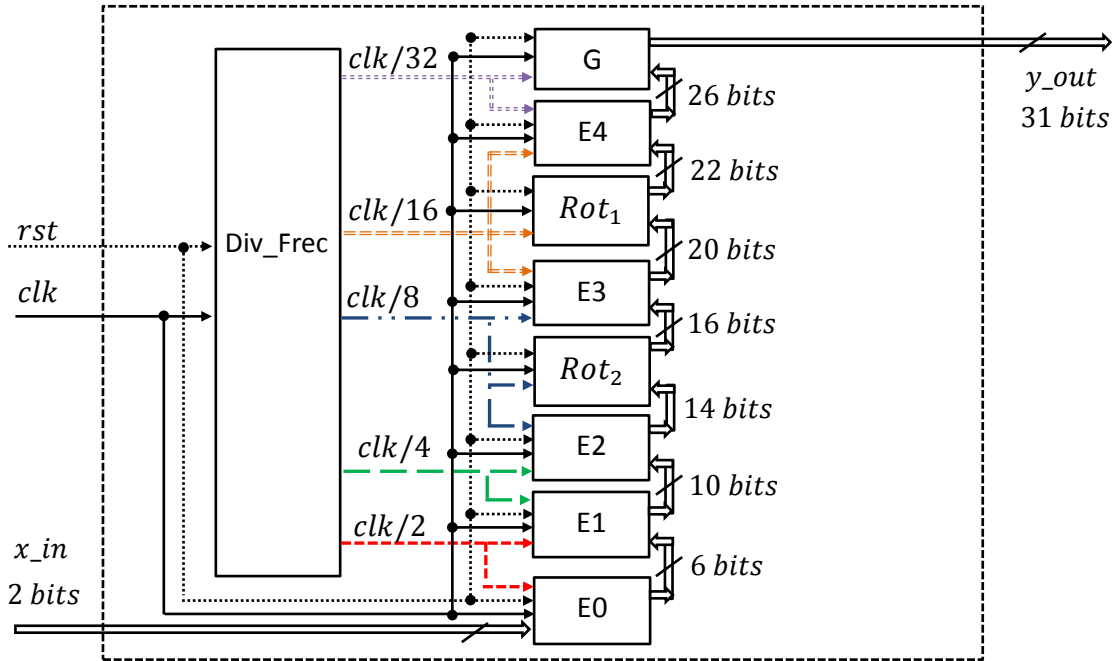


Fig. 5.9. Diagrama a bloques del filtro decimador propuesto con $M = 32, K = 4$ y $R = 2$.

Posteriormente, fue importante conocer la frecuencia de trabajo y la longitud de palabra requerida en cada una de las etapas del filtro. Estos valores son presentados en la Tabla 5.15.

Bloque	Frecuencia de entrada	Frecuencia de salida	Longitud de la palabra de entrada (bits)	Longitud de la palabra de salida (bits)
E0	clk	clk/2	2	6
E1	clk/2	clk/4	6	10
E2	clk/4	clk/8	10	14
Rot ₂	clk/8	clk/8	14	16
E3	clk/8	clk/16	16	20
Rot ₁	clk/16	clk/16	20	22
E4	clk/16	clk/32	22	26
G	clk/32	clk/32	26	31

Tabla 5.15. Consideraciones en el diseño del filtro decimador propuesto ($M = 32, K = 4, R = 2$).

El comportamiento del filtro fue simulado como en los casos anteriores. En las siguientes secciones se resumen los resultados proporcionados por las simulaciones realizadas.

5.4.1. Respuesta al Impulso del Filtro Propuesto con $M = 32, K = 4$ y $R = 2$

La respuesta al impulso obtenida en ACTIVE HDL® y MATLAB® se presenta en las Fig. 5.10a y 5.10b, respectivamente. La señal de salida y_{out} tiene una longitud de 31 bits. A diferencia de las respuestas anteriores esta contiene más muestras de salida. Esto se debe a la inclusión de los dos filtros coseno rotado y del compensador en la estructura del filtro.

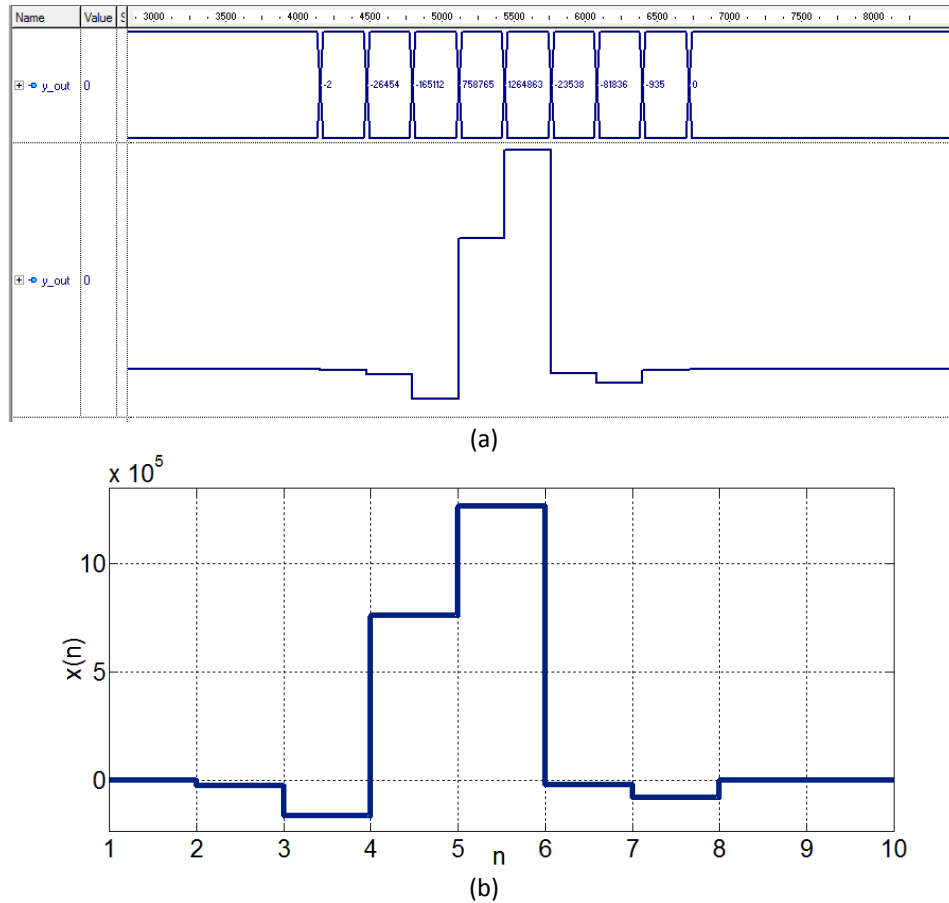


Fig. 5.10. Respuesta al impulso del filtro decimador propuesto ($M = 32, K = 4, R = 2$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.

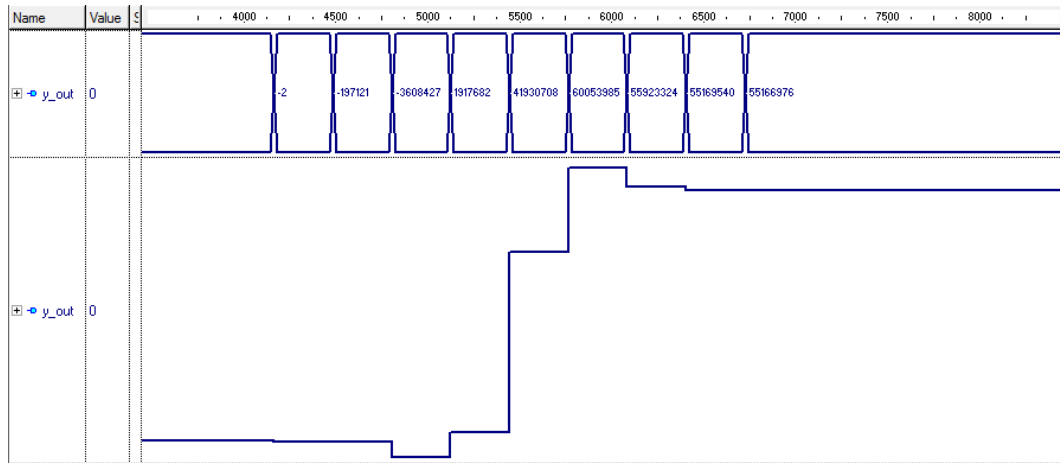
Los valores numéricos de la respuesta al impulso del filtro se presentan en la Tabla 5.16.

MATLAB®	ACTIVE HDL®
-2	-2
-26454	-26454
-165112	-165112
758765	758765
1264863	1264863
-23538	-23538
-81836	-81836
-935	-935

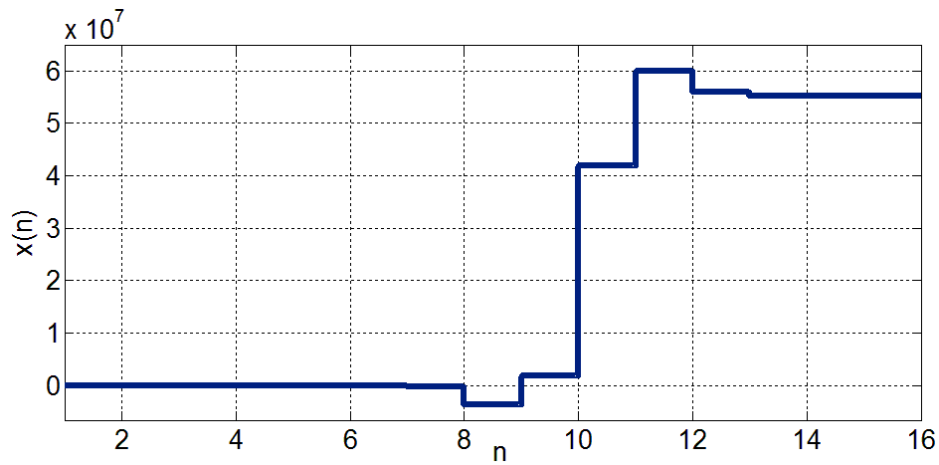
Tabla 5.16. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro propuesto con $M = 32, K = 4$ y $R = 2$).

5.4.2. Respuesta al Escalón del Filtro Propuesto con $M = 32, K = 4$ y $R = 2$

La respuesta al escalón se obtiene aplicando en las señales de entrada los estímulos utilizados en el filtro CIC no recursivo. Las Fig. 5.11a y 5.11b presentan los resultados obtenidos en las simulaciones de ACTIVE HDL® y MATLAB®.



(a)



(b)

Fig. 5.11. Respuesta al escalón del filtro decimador propuesto ($M = 32, K = 4, R = 2$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.

Los valores numéricos obtenidos se incluyen en la Tabla 5.17.

MATLAB®	ACTIVE HDL®
-2	-2
-197121	-197121
-3608427	-3608427
1917682	1917682
41930708	41930708
60053985	60053985
55923324	55923324
55169540	55169540
55166976	55166976

Tabla 5.17. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro propuesto con $M = 32, K = 4$ y $R = 2$).

5.4.3. Recursos Requeridos en la Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 2$

Finalmente, el filtro propuesto fue sintetizado e implementado en la tarjeta NEXYS 4. La lista de recursos consumidos y la máxima frecuencia de operación se incluyen en las Tablas 5.18 y 5.19.

Resumen de Utilización de Recursos			
Utilización Lógica	Usado	Disponible	Utilización
Número de Registros	1381	126800	1%
Número de pares de LUT-FF completos	809	1163	69%
Número de LUTs	815	63400	1%
Número de IOBs	35	210	16%
Frecuencia Máxima	443.616 MHz		

Tabla 5.18. Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro propuesto con $M = 32$, $K = 4$ y $R = 2$.

Bloque	Sumadores/Restadores	Filp-Flops Tipo D	Multiplexores
Divisor de Frecuencia	1	11	1
E0	4	54	0
E1	4	90	0
E2	4	126	0
E3	4	200	0
E4	4	234	0
Filtro Coseno Rotado 1	4	208	0
Filtro Coseno Rotado 2	5	286	0
Compensador	11	234	0
Total	41	1643	1

Tabla 5.19. Recursos requeridos en la implementación del filtro propuesto con $M = 32$, $K = 4$ y $R = 2$.

5.5. Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 3$

Por último, el filtro decimador presentado en de la Fig. 5.12 fue implementado. Al igual que en los filtros anteriores, antes de realizar su descripción en VHDL. Se determinó el valor de los coeficientes del filtro compensador y de los filtros coseno rotado. Estos son presentados en las Tabla 5.20

Filtro	Coficiente	Implementación como sumas y corrimientos
Compensador G	$B_r = 1.84375$	$B_r = 2^1 - 2^{-3} - 2^{-5}$
Coseno Rotado Rot ₁	$\beta_{r_1} = 1.421875$	$\beta_{r_1} = 2^0 + 2^{-1} - 2^{-4} - 2^{-6}$
Coseno Rotado Rot ₂	$\beta_{r_2} = 1.84375$	$\beta_{r_2} = 2^1 - 2^{-3} - 2^{-5}$
Coseno Rotado Rot ₃	$\beta_{r_3} = 1.96875$	$\beta_{r_3} = 2^1 - 2^{-5}$

Tabla 5.20. Coeficientes del compensador y de los filtros coseno rotado.

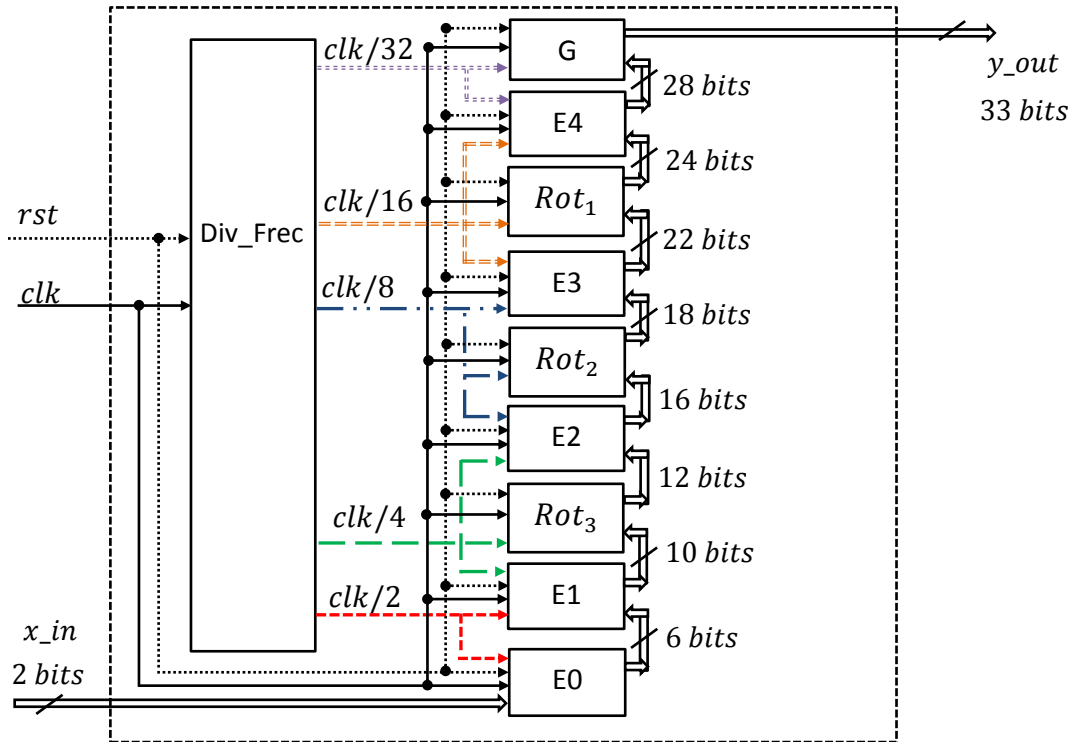


Fig. 5.12. Diagrama a bloques del filtro decimador propuesto con $M = 32, K = 4$ y $R = 3$.

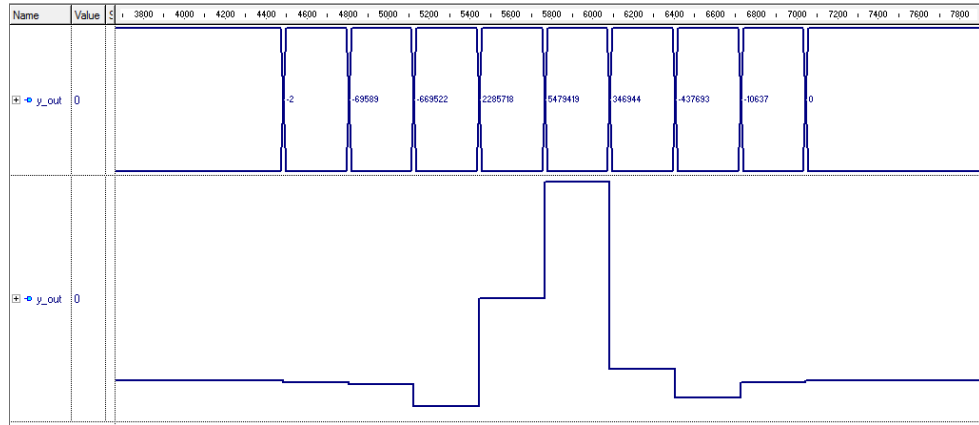
También se calculó la frecuencia de trabajo y longitud de palabra requerida en cada etapa. Estos son incluidos en la Tabla 5.21.

Bloque	Frecuencia de entrada	Frecuencia de salida	Longitud de la palabra de entrada (bits)	Longitud de la palabra de salida (bits)
E0	clk	clk/2	2	6
E1	clk/2	clk/4	6	10
Rot ₃	clk/4	clk/4	10	12
E2	clk/4	clk/8	12	16
Rot ₂	clk/8	clk/8	16	18
E3	clk/8	clk/16	18	22
Rot ₁	clk/16	clk/16	22	24
E4	clk/16	clk/32	24	28
G	clk/32	clk/32	28	33

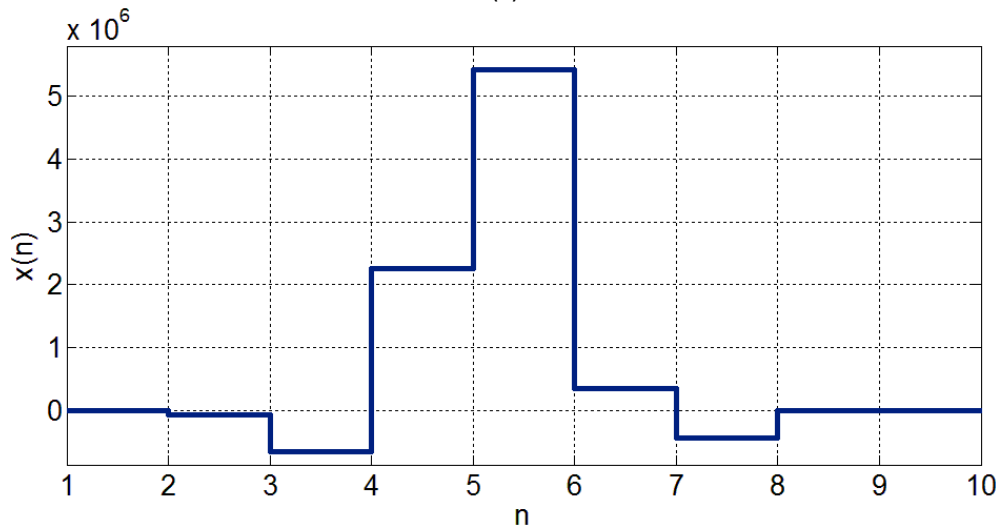
Tabla 5.21. Consideraciones en el diseño del filtro decimador propuesto ($M = 32, K = 4, R = 3$).

5.5.1. Respuesta al Impulso del Filtro Propuesto con $M = 32, K = 4$ y $R = 3$

La respuesta al impulso obtenida en la simulación del comportamiento del filtro se presenta en la Fig. 5.13.



(a)



(b)

Fig. 5.13. Respuesta al impulso del filtro decimador propuesto ($M = 32, K = 4, R = 3$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.

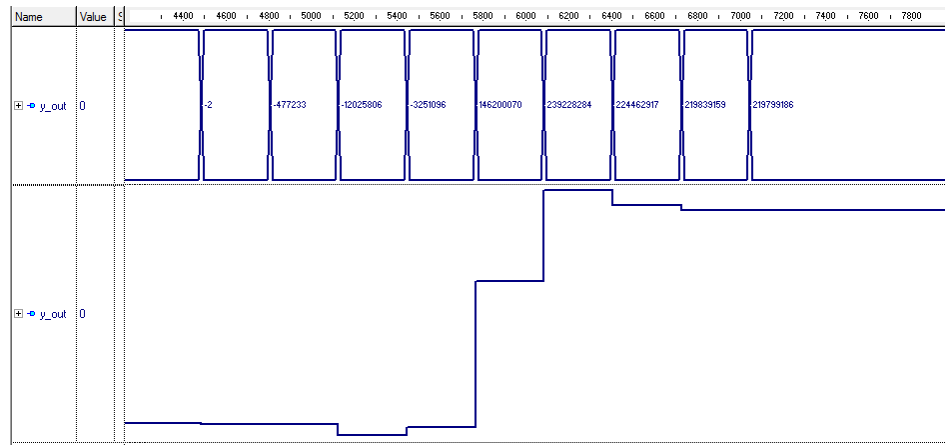
Los valores numéricos calculados son presentados en la Tabla 5.22.

MATLAB®	ACTIVE HDL®
-2	-2
-69589	-69589
-669522	-669522
2285718	2285718
5479419	5479419
346944	346944
-437693	-437693
-10637	-10637

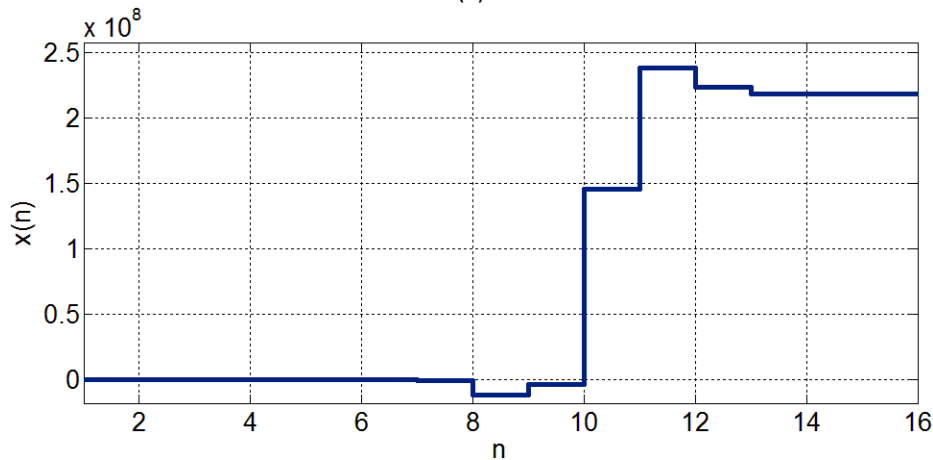
Tabla 5.22. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro propuesto con $M = 32, K = 4$ y $R = 3$).

5.5.2. Respuesta al Escalón del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 3$

La respuesta al escalón de este filtro se ilustra en la Fig. 5.14.



(a)



(b)

Fig. 5.14. Respuesta al escalón del filtro decimador propuesto ($M = 32, K = 4, R = 3$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.

Finalmente en la Tabla 5.23 se incluyen los valores calculados en ambos simuladores.

MATLAB®	ACTIVE HDL®
-2	-2
-477233	-477233
-12025806	-12025806
-3251096	-3251096
146200070	146200070
239228284	239228284
224462917	224462917
219839159	219839159
219799186	219799186

Tabla 5.23. Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro propuesto con $M = 32, K = 4$ y $R = 3$).

5.5.3. Recursos Requeridos en la Implementación del Filtro Propuesto con $M = 32$, $K = 4$ y $R = 3$

Los resultados arrojados por el sintetizador ISE 14.7 se muestran en las Tablas 5.24 y 5.25.

Resumen de Utilización de Recursos			
Utilización Lógica	Usado	Disponible	Utilización
Número de Registros	1647	126800	1%
Número de pares de LUT-FF completos	947	1277	74%
Número de LUTs	948	63400	1%
Número de IOBs	38	210	18%
Frecuencia Máxima	433.971 MHz		

Tabla 5.24. Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro propuesto con $M = 32$, $K = 4$ y $R = 3$.

Bloque	Sumadores/Restadores	Filp-Flops Tipo D	Multiplexores
Divisor de Frecuencia	1	11	1
E0	4	54	0
E1	4	90	0
E2	4	144	0
E3	4	220	0
E4	4	290	0
Filtro Coseno Rotado 1	3	108	0
Filtro Coseno Rotado 2	4	234	0
Filtro Coseno Rotado 3	5	312	0
Compensador	9	476	0
Total	42	1939	1

Tabla 5.25. Recursos requeridos en la implementación del filtro propuesto con $M = 32$, $K = 4$ y $R = 3$.

5.6. Comparaciones

En las secciones anteriores se presentaron las consideraciones que se realizaron para la implementación de cada uno de los filtros. El resumen de los recursos consumidos por los filtros implementados se muestra en la Tabla 5.26.

Filtro	Sumadores/Restadores	Filp-Flops Tipo D
Filtro CIC No Recursivo $M = 32$, $K = 4$.	21	641
Filtro Propuesto $M = 32$, $K = 4$, $R = 1$.	37	1349
Filtro Propuesto $M = 32$, $K = 4$, $R = 2$.	41	1643
Filtro Propuesto $M = 32$, $K = 4$, $R = 3$.	42	1939

Tabla 5.26. Comparación de Recursos Requeridos en la Implementación de los Filtros Decimadores.

Se observa que existe un aumento en el consumo de hardware de los filtros propuestos en relación con un filtro comb no recursivo. Este aumento depende del número de filtros coseno rotado agregados en su estructura. También influye la cantidad de sumas y corrimientos requeridos en la implementación de los coeficientes. Los recursos consumidos por lo filtros, pueden reducirse si se aplican las técnicas de redondeo o truncamiento en los datos de salida de cada una de las etapas. Esto permite reducir el tamaño de palabra interna requerida para el procesamiento de la señal. Sin embargo, se pierde exactitud en el cálculo de las muestras de salida.

Finalmente, se observa que la respuesta en magnitud de los filtros propuestos presenta una notable mejoría. La caída en la banda de paso es reducida, y la atenuación en determinadas bandas de rechazo es incrementada.

Conclusiones

El objetivo principal de este trabajo consistió en obtener una estructura de decimación con características en magnitud mejoradas. Con este fin, se retomó la estructura del filtro comb no recursivo debido a su sencillez, y se realizaron las modificaciones necesarias para mejorar su respuesta en magnitud.

En primer lugar, se buscó incrementar la atenuación en las bandas de rechazo. Por lo tanto, se propuso la aplicación de la técnica de rotación de ceros a un filtro coseno expandido, en lugar de la tradicional rotación de ceros en el filtro comb. De esta forma, se obtiene un término no recursivo, que dependiendo de la posición que ocupe dentro de la estructura del filtro comb, aumentara la atenuación en determinadas bandas de rechazo.

La caída en la banda de paso se ve disminuida con adición de un filtro compensador en baja frecuencia. Sin embargo, la inclusión de estos filtros propicia la necesidad de utilizar multiplicadores para la implementación de sus coeficientes. Por lo tanto, con el objetivo de mantener una estructura sencilla, se aplica redondeo a dichos coeficientes. De esta manera, estos son representados a base de sumas y corrimientos.

Se realizó la implementación de los filtros propuestos, y se observó que requieren un incremento en consumo de hardware en comparación con un filtro comb no recursivo. Este puede ser disminuido buscando otras alternativas para la representación de los coeficientes.

Finalmente, la estructura propuesta puede ser una alternativa útil, donde el requisito más importante a satisfacer sea una respuesta en magnitud con una alta atenuación en las bandas de rechazo y baja caída en la banda de paso.

Trabajo a Futuro

Como trabajo a futuro se plantean los siguientes puntos:

- Mejorar la estructura propuesta en cuestión de consumo de recursos de hardware. Para esto se sugieren dos posibles soluciones. La primera de ellas consiste, en probar otras técnicas en la representación de los coeficientes del filtro compensador y de los filtros coseno rotado. De esta forma, se busca reducir la cantidad de sumadores requeridos por estos filtros, que son los que más contribuyen en el incremento de hardware. En segundo lugar, se propone probar el efecto que tendría utilizar truncamiento o redondeo en los datos de salida de cada etapa del filtro propuesto.
- Caracterizar la estructura propuesta en términos de consumo de potencia y área.

Lista de Figuras

<i>Capítulo 1</i>		<i>Pág.</i>
1.1	Respuesta en magnitud del filtro comb ($M = 12, K = 1$).....	2
1.2	Respuesta en magnitud del filtro comb para diferentes valores de K . (a) Todo el intervalo de frecuencias. (b) Banda de paso.....	3
1.3	Ubicación de polos y ceros de un filtro comb ($M = 12, K = 1$).....	4
1.4	Decimación.....	5
1.5	Proceso de sub-muestreo ($M = 4$). (a) Dominio del tiempo. (b) Dominio de la frecuencia.....	7
1.6	Efecto de aliasing debido al proceso de sub-muestreo ($M = 8$). (a) Dominio del tiempo. (b) Dominio de la frecuencia.....	7
1.7	Respuesta en magnitud ideal en la primera etapa del filtro decimador y de un filtro comb ($M = 12, K = 1, v = 2$).....	8
1.8	Diagrama a bloques de la estructura del filtro CIC.....	10
1.9	Arquitecturas para la implementación de (a) un integrador, (b) un derivador.....	10
1.10	Estructura del filtro comb no recursivo para (a) $M = 2^p$, (b) $M = 3^p$	12
 <i>Capítulo 2</i>		
2.1	Respuestas en magnitud del Ejemplo 2.1. (a) $M = 12, K = 4, B = 1$, (b) $M = 32, K = 4, B = 1$	18
2.2	Respuesta en magnitud del pre-filtro coseno para diferentes valores de N	19
2.3	Respuestas en magnitud del Ejemplo 2.2.....	20
2.4	Respuestas en magnitud del Ejemplo 2.3.....	21
2.5	Respuestas en magnitud del Ejemplo 2.4.....	23
2.6	Respuestas en magnitud del Ejemplo 2.5.....	25
2.7	Respuestas en magnitud del Ejemplo 2.6. (a) Toda la banda de frecuencias. (b) Banda de paso.....	28
2.8	Respuestas en magnitud del Ejemplo 2.7. (a) Toda la banda de frecuencias, (b) Banda de paso.....	29
2.9	Respuestas en magnitud del Ejemplo 2.8. (a) Toda la banda de frecuencias, (b) Banda de paso.....	31

 Capítulo 3

3.1	Respuesta en magnitud del filtro coseno expandido para diferentes valores de N .	35
3.2	Repuestas en magnitud del filtro comb ($M = 32, K = 3$) y de filtros coseno con diferentes valores de N .	36
3.3	Respuestas en magnitud del filtro comb ($M = 32, K = 3$) conectado en cascada con filtros coseno con diferentes valores de N .	37
3.4	Respuestas en magnitud del Ejemplo 3.2.	38
3.5	Caída en la banda de paso de los filtros del Ejemplo 3.2.	39
3.6	Comparación entre la caída de la banda de paso de un filtro comb ($M = 16, K = 1, 2$ y 3) y del filtro $H_r(z^{M/2^k})$ para (a) $k = 1$, (b) $k = 1$ y 2 (c) $k = 1, 2$ y 3 .	40
3.7	Comparación entre la caída de la banda de paso de un filtro comb ($M = 32, K = 1, 2$ y 3) y del filtro $H_r(z^{M/2^k})$ para (a) $k = 1$, (b) $k = 1$ y 2 (c) $k = 1, 2$ y 3 .	40
3.8	Banda de paso de los filtros del Ejemplo 3.3.	41
3.9	Diagrama a bloques de la estructura general del filtro propuesto para $R = 3$.	44
3.10	Diagrama a bloques de la estructura del filtro propuesto. (a) $R = 1$, (b) $R = 2$ y (c) $R = 3$.	45
3.11	Respuestas en magnitud del Ejemplo 3.4.	46
3.12	Diagrama a bloques de la estructura del filtro del Ejemplo 3.4.	46
3.13	Respuestas en magnitud del Ejemplo 3.5.	47
3.14	Diagrama a bloques de la estructura del filtro del Ejemplo 3.5.	47
3.15	Respuestas en magnitud del Ejemplo 3.6.	48
3.16	Respuestas en magnitud del Ejemplo 3.7.	49
3.17	Respuestas en magnitud del Ejemplo 3.8.	51
3.18	Diagrama a bloques de la estructura del filtro del Ejemplo 3.8.	51
3.19	Respuestas en magnitud del Ejemplo 3.9.	52
3.20	Respuestas en magnitud del Ejemplo 3.10.	53

Capítulo 4

4.1	Bloque para la implementación del divisor de frecuencias.	57
4.2	Descripción en VHDL de un divisor de frecuencia para un filtro comb no recursivo.	58
4.3	Diagrama de tiempos del divisor de frecuencia con $M = 16$.	59
4.4	Celda básica para la implementación de $1 + z^{-1}$.	59
4.5	Diagrama a bloques de la implementación de una etapa de un filtro comb no recursivo.	60
4.6	Arquitectura general de un filtro comb no recursivo con $M = 2^p$.	60

4.7	Incremento en el tamaño del dato de entrada en cada celda.....	61
4.8	Estructura del filtro comb no recursivo del Ejemplo 4.1.....	61
4.9	Estructura para la implementación de un filtro comb no recursivo.....	62
4.10	Bloque para la implementación de la primera etapa del filtro comb no recursivo...	63
4.11	Bloque para la implementación de las etapas restantes del filtro comb no recursivo.....	63
4.12	Descripción en VHDL del primer proceso de la Etapa 0.....	64
4.13	Descripción en VHDL del segundo proceso de la Etapa 0.....	65
4.14	Descripción en VHDL del primer proceso de la Etapa i	65
4.15	Descripción en VHDL del segundo proceso de la Etapa i	66
4.16	Estructura para la implementación en FPGA del filtro propuesto ($R = 3$).....	66
4.17	(a) Bloque y (b) estructura para la implementación de filtro coseno rotado.....	67
4.18	Descripción en VHDL del filtro coseno rotado del Ejemplo 4.2.....	68
4.19	(a) Estructura y (b) bloque para la implementación del filtro compensador.....	69
4.20	Descripción en VHDL del filtro compensador del Ejemplo 4.3.....	70

Capítulo 5

5.1	Diagrama a bloques para la implementación de un filtro comb no recursivo con $M = 32$ y $K = 4$	72
5.2	Respuesta al impulso del filtro comb no recursivo ($M = 32, K = 4$) obtenida en ACTIVE HDL®.....	74
5.3	Respuesta al impulso del filtro comb no recursivo ($M = 32, K = 4$) obtenida en MATLAB®.....	74
5.4	Respuesta al escalón del filtro comb no recursivo ($M = 32, K = 4$) obtenida en ACTIVE HDL®.....	75
5.5	Respuesta al escalón del filtro comb no recursivo ($M = 32, K = 4$) obtenida en MATLAB®.....	75
5.6	Diagrama a bloques para la implementación de un filtro decimador propuesto con $M = 32, K = 4$ y $R = 4$	77
5.7	Respuesta al impulso del filtro decimador propuesto ($M = 32, K = 4, R = 1$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.....	78
5.8	Respuesta al escalón del filtro decimador propuesto ($M = 32, K = 4, R = 1$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.....	79
5.9	Diagrama a bloques para la implementación de un filtro decimador propuesto con $M = 32, K = 4$ y $R = 2$	81

5.10	Respuesta al impulso del filtro decimador propuesto ($M = 32, K = 4, R = 2$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.....	82
5.11	Respuesta al escalón del filtro decimador propuesto ($M = 32, K = 4, R = 2$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.....	83
5.12	Diagrama a bloques para la implementación de un filtro decimador propuesto con $M = 32, K = 4$ y $R = 3$	85
5.13	Respuesta al impulso del filtro decimador propuesto ($M = 32, K = 4, R = 3$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.....	86
5.14	Respuesta al escalón del filtro decimador propuesto ($M = 32, K = 4, R = 3$). (a) Resultados obtenidos en ACTIVE HDL®. (b) Resultados obtenidos en MATLAB®.....	87

Lista de Tablas

Capítulo 1

1.1	Caída en la banda de paso.....	3
1.2	Atenuación en la primera banda de rechazo.....	4

Capítulo 2

2.1	Coeficiente B para diferentes valores de K	17
2.2	Coeficiente B redondeado para diferentes valores de K	17
2.3	Valores de B en términos de potencias de dos.....	17
2.4	Valores óptimos de q	23
2.5	Funciones de cambio de amplitud.....	24

Capítulo 3

3.1	Bandas de rechazo mejoradas.....	36
3.2	Valores de B calculados considerando el parámetro $K + 2$	41
3.3	Caída en la banda de paso de los filtros del Ejemplo 3.3.....	42
3.4	Valores ajustados de B	42
3.5	Comparación de la caída en la banda de paso de $H(z)$ y de $H_c(z)$ compensado para $v = 2$	42
3.6	Comparación de la caída en la banda de paso de $H(z)$ y de $H_c(z)$ compensado para $v = 4$	43
3.7	Comparación de la caída en la banda de paso de $H(z)$ y de $H_c(z)$ compensado para $v = 8$	43
3.8	Valores de los coeficientes redondeados en el Ejemplo 3.8.....	51
3.9	Valores de los coeficientes redondeados en el Ejemplo 3.9.....	52
3.10	Valores de los coeficientes redondeados en el Ejemplo 3.10.....	53
3.11	Cantidad de APOS adicionales requeridas por la estructura propuesta.....	54
3.12	Valores de los coeficientes redondeados en el Ejemplo 3.11.....	54
3.13	Comparación de APOS del Ejemplo 3.11.....	55

Capítulo 4

4.1	Crecimiento del dato de entrada en cada celda.....	62
-----	--	----

Capítulo 5

5.1	Consideraciones en el diseño del filtro comb no recursivo.....	72
5.2	Estímulos aplicados a las señales de entrada del filtro comb no recursivo para obtener su respuesta al impulso.....	73
5.3	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro comb no recursivo).....	74
5.4	Estímulos aplicados a las señales de entrada del filtro comb no recursivo para obtener su respuesta al escalón.....	75
5.5	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro comb no recursivo).....	76
5.6	Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro comb no recursivo.....	76
5.7	Recursos requeridos en la implementación del filtro comb no recursivo.....	76
5.8	Coeficientes del compensador y del filtro coseno rotado.....	77
5.9	Consideraciones en el diseño del filtro decimador propuesto ($M = 32, K = 4, R = 1$).....	77
5.10	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro propuesto con $M = 32, K = 4$ y $R = 1$).....	78
5.11	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro propuesto con $M = 32, K = 4$ y $R = 1$).....	79
5.12	Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro propuesto con $M = 32, K = 4$ y $R = 1$	80
5.13	Recursos requeridos en la implementación del filtro propuesto con $M = 32, K = 4$ y $R = 1$	80
5.14	Coeficientes del compensador y de los filtros coseno rotado.....	80
5.15	Consideraciones en el diseño del filtro decimador propuesto ($M = 32, K = 4, R = 2$).....	81
5.16	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro propuesto con $M = 32, K = 4$ y $R = 2$).....	82
5.17	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro propuesto con $M = 32, K = 4$ y $R = 2$).....	83

5.18	Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro propuesto con $M = 32, K = 4$ y $R = 2$	84
5.19	Recursos requeridos en la implementación del filtro propuesto con $M = 32, K = 4$ y $R = 2$	84
5.20	Coefficientes del compensador y de los filtros coseno rotado.....	84
5.21	Consideraciones en el diseño del filtro decimador propuesto ($M = 32, K = 4, R = 3$).....	85
5.22	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al impulso del filtro propuesto con $M = 32, K = 4$ y $R = 3$).....	86
5.23	Comparación de los resultados obtenidos en MATLAB® y ACTIVE HDL® (respuesta al escalón del filtro propuesto con $M = 32, K = 4$ y $R = 3$).....	87
5.24	Recursos ocupados en el FPGA y máxima frecuencia de operación del filtro propuesto con $M = 32, K = 4$ y $R = 3$	88
5.25	Recursos requeridos en la implementación del filtro propuesto con $M = 32, K = 4$ y $R = 3$	88
5.26	Comparación de los recursos requeridos en la implementación de los filtros decimadores.....	88

REFERENCIAS

- [1] F. Daneshgaran and M. Laddomada, "A Novel Class of Decimation Filters for Sigma Delta A/D converters," *Wireless Communications and Mobile Computing*, vol. 2, (2002), pp. 86-882.
- [2] G. Jovanovic Dolecek, *Multirate Systems: Design and Applications*, Idea Group Publishing, (2002), Hershey USA.
- [3] B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Trans. On Acoust. Speech, Signal Processing*, vol.29, (April 1981), pp. 155–162.
- [4] Seongdo Kim, Weon-Cheol Lee, Sungsoon Ahn and Seungwon Choi, "Design of CIC Roll-Off Compensation Filter in a W-CDMA Digital IF Receiver", *Digital Signal Processing*, vol. 16, (November 2006), pp. 846-854.
- [5] G. Jovanovic Dolecek, "Simple Wideband CIC Compensator," *Electronics Letters*, (19 November 2009), pp. 1270-1272.
- [6] G. Jovanovic Dolecek and L. Dolecek, "Novel Multiplierless Wide-band CIC Compensator," *Circuits and Systems (ISCAS), Proceedings IEEE International Symposium*, (30-May-2-June 2010), pp. 2119-2122.
- [7] A. Fernandez Vazquez and G. Jovanovic Dolecek, "Maximally Flat CIC Compensation Filter: Design and Multiplierless Implementation," *Circuits and Systems II: Express Briefs, IEEE Transactions*, (February 2012), pp. 113-117.
- [8] D. E. Troncoso Romero and G. Jovanovic Dolecek, "Application of Amplitude Transformation for Compensation of Comb Decimation Filters," *Electronic Letters*, vol. 49 No. 16, (August 2013), pp. 985-987.
- [9] G. Jovanovic Dolecek and F. J. Trejo Torres, "Compensated CIC-Cosine Decimation Filter," *ECTI Transactions on Computer and Information Technology*, vol. 4, No. 1, (May 2010), pp. 256-259.
- [10] G. Jovanovic Dolecek and A. Fernandez Vazquez, "Trigonometrical Approach to Design a Simple Wideband Comb Compensator," *International Journal of Electronics and Communications*, In press.
- [11] Y. Lian and Y. C. Lim, "New Prefilter Structure for Designing FIR Filters," *Electronic Letters*, vol. 29, (27 May 1993), pp. 1034-1036.
- [12] G. Jovanovic Dolecek and J. D. Carmona, "A New Cascaded Modified CIC-Cosine Decimation Filter," *Circuits and Systems ISCAS, IEEE International Symposium*, vol.4, (23-26 May 2005), pp. 3733-3736.
- [13] G. Jovanovic Dolecek and S. K. Mitra, "Stepped Triangular CIC-Cosine Decimation Filter," *Signal Processing Symposium NORSIG*, (7-9 June 2006), pp. 26-29.

-
- [14] F. J. Trejo Torres and G. Jovanovic Dolecek, "Compensated CIC-Cosine Decimation Filter," Communications and Information Technologies, ISCIT International Symposium, (17-19 October 2007), pp. 256-259.
- [15] G. Jovanovic Dolecek and M. Laddomada, "Comb-Cosine Prefilter Based Decimation Filter," Industrial Technology (ICIT), IEEE International Conference, (14-17 March 2010), pp. 205-210.
- [16] G. Jovanovic Dolecek and J. K. Mitra, "A New Sharpened Cascaded Comb-Cosine Decimation Filter," (30 July 2011).
- [17] G. Jovanovic Dolecek and J. D. Carmona, "Generalized CIC-Cosine Decimation Filter," Industrial Electronics and Applications (ISIEA), IEEE Symposium, (3-5 October 2012), pp. 640-645.
- [18] L. Lo Presti, "Efficient Modified-Sinc Filters for Sigma-Delta A/D Converters," IEEE Trans. Circuits Syst. II, Analog and Digital Signal Processing, vol.47, No.11, (2000), pp 1204-1213.
- [19] G. Jovanovic Dolecek and S. K. Mitra, "Efficient Comb-Rotated-Sinc (RS) Decimator with Sharpened Magnitude Response," Circuits and Systems Midwest Symposium, MWSCAS, vol.2, (25-28 July 2004), pp. 116-120.
- [20] G. Jovanovic Dolecek, "A New Modified Comb-Rotated Sinc (RS) Decimator with Improved Magnitude Response," Electronics, Circuits and Systems ICECS, IEEE International Conference, (11-14 December 2007), pp. 250-253.
- [21] G. Jovanovic Dolecek and A. Fernandez Vazquez, "On Nonrecursive Rotated Comb Filter," Circuits and Systems (MWSCAS), IEEE 55th International Midwest Symposium, (5-8 August 2012), pp. 474-477.
- [22] M. Laddomada, "Generalized Comb Decimation Filters for $\Sigma\Delta$ A/D Converters: Analysis and Design," IEEE Transactions on Circuits and Systems, vol. 57, (May 2007), pp. 94-1005.
- [23] A. Fernandez Vazquez and G. Jovanovic Dolecek, "Multiplierless Implementation of Generalized Comb Filters (GCF) Based on Chebyshev Polynomials," 19th European Signal Processing Conference (EUSIPCO), (29 August-2 September 2011), pp. 1509-1513.
- [24] M. Laddomada, "On the Polyphase Decomposition for Design of Generalized Comb Decimation Filters," Circuits and Systems I: Regular Papers, IEEE Transactions, vol. 55, (September 2008), pp. 2287 – 2299.
- [25] A. Fernandez Vazquez and G. Jovanovic Dolecek, "A General Method to Design GCF Compensation Filter," IEEE Transactions on Circuits and Systems II, vol. 56,(May 2009), pp. 409-413.
- [26] G. Jovanovic Dolecek and M. Laddomada, "An Economical Class of Droop-Compensated Generalized Comb Filters: Analysis and Design," IEEE Transactions on Circuits and Systems, vol.57, no. 7, (April 2010), pp. 275-279.

-
- [27] A. Fernandez Vazquez and G. Jovanovic Dolecek, "An L_1 design of GCF compensation filter," *Signal Processing*, vol. 91, (May 2010), pp. 1143-1149.
- [28] J. F. Kaiser and R. W. Hamming, "Sharpening the Response of a Symmetric Nonrecursive Filter by Multiple Use of the Same Filter," *Acoustics, Speech and Signal Processing*, IEEE International Conference on ICASSP, vol. 2, (May 1977), pp. 82-85.
- [29] A. Y. Kwentus, Z. Jiang, and A. N. Wilson, "Application of Filter Sharpening to Cascaded Integrator-Comb Decimation Filters," *Signal Processing IEEE Transactions*, vol. 45, (February 1997), pp. 457-467.
- [30] G. Jovanovic Dolecek and S. K. Mitra, "Efficient Sharpening of CIC Decimation Filter," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6, (6-10 April 2003), pp.385-388.
- [31] G. Jovanovic Dolecek and S. K. Mitra, "Two-Stage CIC Based Decimator with Improved Characteristics," *IET Signal Processing*, vol.4, Issue 1, (February 2010), pp.22-29.
- [32] G. Stephen and R. W. Stewart, "High-speed Sharpening of Decimating CIC Filter," *Electronic Letters*, vol. 40, (14 October 2004), pp. 1383-1384.
- [33] R. Hartnett and G. Boudreaux, "Improved filter sharpening," *IEEE Trans. on Signal Process*, (1995), p. 2805-2810.
- [34] M. G. Cruz Jimenez and G. Jovanovic Dolecek, "Application of Generalized Sharpening Technique for Two-Stage Comb Decimator Filter Design," *Procedia Technology*, Iberoamerican Conference on Electronics and Computer Science, vol. 7, (2013), pp.142-149.
- [35] M. G. Cruz Jimenez, V. Calderón Reyes and G. Jovanovic Dolecek, "Sharpening of Non-Recursive Comb Decimation Structure," *International Symposium on Communications and Information Technologies*, (4-6 September 2013), pp.458-463.
- [36] D. E. Troncoso Romero, M. Laddomada and G. Jovanovic Dolecek, "Optimal Sharpening of Compensated Comb Decimation Filters: Analysis and Design", *The Scientific World Journal* (2014).
- [37] G. Molnar, M. G. Pecotic and M. Vucic, "Weighted Least-Squares Design of Sharpened CIC Filters," *Information and Communication Technology Electronics and Microelectronics (MIPRO) 36th International Convention*, (20-24 May 2013), pp. 91-95.
- [38] A. Vazquez Fernandez and G. Jovanovic Dolecek, "Passband and Stopband CIC Improvement Based on Efficient IIR Filter Structure," *Circuit and Systems (MWSCAS) 53rd IEEE International Midwest Symposium*, (1-4 August 2010), pp. 765-768.
- [39] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with VHDL design*, Mc. Graw Hill, (2005), USA.
- [40] K. Chang, *Digital Systems Designs with VHDL and Synthesis: An Integrated Approach*, USA, IEEE Society, 1999.

- [41] J. P. Deschamps, G. J. A. Bioul and G. D. Sutter, *Synthesis of Arithmetic Circuits: FPGA, ASIC and embedded Systems*, John Wiley & Sons, USA, 2006.
- [42] Meyer-Baese U. (2007), *Digital Signal Processing with Field Programmable Gate Arrays*, 3rd edition, Springer International Edition.
- [43] G. Jovanovic Dolecek and M. Laddomada, "A Novel Two-Stage Nonrecursive Structure for the Design of Generalized Comb Filters," *Digital Signal Processing*, vol. 22,(September 2011), pp. 859-868.
- [44] G. Jovanovic Dolecek and M. Laddomada, "An Improved Class of Multiplierless Decimation Filters: Analysis and Design," *Digital Signal Processing*, vol. 23, (September 2013), pp. 1773-1782.