Efficient design methods for FIR digital filters

by

Miriam Guadalupe Cruz Jiménez

A thesis submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Electronics

**National Institute for Astrophysics,
Optics and Electronics (INAOE)**

February 2017
Tonantzintla, Puebla

Supervised by:

Prof. Gordana Jovanovic Dolecek, Ph D

Efficient design methods for FIR digital filters

Miriam Guadalupe Cruz Jiménez

Abstract

The design of low-complexity linear-phase Finite Impulse Response (FIR) filters is investigated in this thesis. The proposals developed here are particularly useful for digital communication applications.

An efficient and essential method to achieve low complexity is to split the filters into simple subfilters, and among the most important subfilters for such purpose are the comb and cosine filters. These filters have a low computational complexity and a low utilization of hardware resources but very poor magnitude characteristics. In this sense, novel architectures have been developed in the present thesis using the comb and cosine filters as a basis. The resulting architectures, especially useful for low-pass narrowband filtering in sampling rate conversion, achieve better magnitude characteristics and better trade-offs in power, area and speed compared with previous systems recently developed in literature that rely in simple subfilters as well.

For filters with constant coefficients, an effective method to realize low-complexity filters is to express the coefficients without multipliers, which are the most expensive elements in terms of area, power and speed. For this case, the proposed contribution focuses on the implementation of the constant multiplications as a network of additions and shifts. Novel theoretical lower bounds for the number of pipelined operations that are needed in Single Constant Multiplication (SCM) and Multiple Constant Multiplication (MCM) blocks have been

developed here. These lower bounds have been stablished under the consideration that every operation (addition or subtraction) can have *n* inputs, and the cost of a pipelined operation is the same as the cost of a single pipeline register. The aforementioned consideration is particularly important because it occurs in the newest families of Field Programmable Gate Arrays (FPGAs), which currently are a preferred platform for the implementation of DSP algorithms.

Resumen

En esta tesis se llevo a cabo la investigación del diseño de filtros de fase lineal con respuesta al impulso finita (FIR, Finite Impulse Response) de baja complejidad. Las propuestas desarrolladas son particularmente útiles para aplicaciones en comunicaciones digitales.

Un método para disminuir la complejidad, que ha resultado fundamental y eficiente, consiste en dividir los filtros en subfiltros simples, dentro de los subfiltros más importantes para tal propósito se encuentran los filtros comb y coseno. Estos filtros tienen baja complejidad computacional y utilizan pocos recursos de hardware sin embargo presentan una característica en magnitud pobre. Por tal motivo, nuevas arquitecturas han sido desarrolladas en esta tesis usando los filtros comb y coseno como base. Las arquitecturas resultantes, especialmente útiles para filtrado pasa bajas de banda angosta para conversión de tazas de muestreo, presentan mejor característica de magnitud y mejor trade-offs en potencia, área y velocidad en comparación con sistemas previos recientemente desarrollados en la literatura que también dependen de subfiltros simples.

Para filtros con coeficientes constantes, un método que ha resultado efectivo para diseñar filtros de baja complejidad consiste en expresar los coeficientes sin multiplicadores, los cuales son los elementos más costosos en términos de área, potencia y velocidad. Para este caso, la contribución aquí propuesta está enfocada en la implementación de las multiplicaciones por constantes como una red de adiciones y desplazamientos. Se desarrollaron nuevos límites inferiores teóricos para el número de operaciones con pipelining que son necesarias en bloques de multiplicaciones por una constante (SCM, Single Constant Multiplication) y multiplicaciones por múltiples constantes (MCM, Multiple Constant Multiplication). Estos límites inferiores se establecieron bajo la consideración que cada operación (suma o resta) puede tener *n* entradas, y el costo de una operación con pipelining es igual al costo de un registro simple de pipelining. El argumento anterior es particularmente importante porque así se considera en las familias más nuevas de arreglos de compuertas programables en campo (FPGAs, Field Programmable Gate Arrays), cuya plataforma es preferida actualmente para la implementación de algoritmos DSP .

To my husband David

Acknowledgments

I am thankful to God for all that He has made in my life. *I say of Jehovah, My refuge and My fortress, My God in whom I trust! (Psalm 91:2)*.

I am grateful to CONACYT for granting me the scholarships no. 224191, 290842 and 290935. I also thank all the staff of the institute INAOE: researchers, administrative staff, secretaries, security guards, cleaning staff, library staff and dining room staff. Since my arrival to INAOE I received kindness, support, friendship, teaching and I really appreciate it. *I thank my God always concerning you (1 Cor. 1:4)*.

I did not come to the point of finalizing my PhD without help. There are many people that have contributed in some way, and I thank each one of them. First, I thank my thesis advisor in master's and doctoral theses, Dr. Gordana Jovanovic Dolecek, who guided me with wisdom. I learned so much of her. Also, I am grateful to my doctoral committee: Dr. Uwe Meyer-Baese, Dr. Alfonso Fernández Vázquez, Dr. Francisco Javier De la Hidalga Wade, Dr. Luis Hernández Martínez and Dr. Jorge Roberto Zurita Sánchez for all the support and advices. I also thank my teachers for helping me and teaching me through the master´s and doctoral courses: M. Sc. Jacobo Meza, Dr. Celso Gutiérrez, Dr. Pedro Rosales, Dr. Roberto Murphy, Dr. Juan Manuel Ramírez, Dr. Arturo Sarmiento, Dr. Reydezel Torres, Dr. Esteban Tlelo, Dr. Ignacio Zaldívar, Dr. José de Jesús Rangel. *And I will give you shepherds according to My own heart, who will feed you knowledge and understanding (Jer. 3:15).*

Someone special that has been more than a guide, teacher, partner, friend, I have not words to appreciate his wonderful company; my counterpart David Ernesto Troncoso Romero, thank you for all the advices, the sleepless nights helping me and for encouraging me every moment. *And the two shall be one flesh. So then they are no longer two, but one flesh (Mark 10:8).*

Also, I am grateful to all family, my mom Geno, aunt Chuy, grandpa Benito, my siblings Moy, Luz and Luis, my mother in law Ricarda, my sister in law Anita, my nephews Kevin, Isaac and Elías, my nieces Joana and Ruth, cousins Francis, Marisol, Lupe, uncle Quiri, granny Romana and other relatives for all the patience, personal support and for forgiving me not having been in special moments. *And over all these things put on love, which is the uniting bond of perfectness (Col. 3:14).*

I am more than grateful to Mike Lynch, Raúl and María Oquendo, Art and Adele TerMorshuizen, Laszlo and Jozann Roszol, Aaron and Charis Chen, Daniel and Shifrah Combiths, Philip and Julia Holly, Yin and Lin Zhang, Jim and Pam Waldrup, Camille and Joey Calascibetta, Viridiana and David Colon, Thomas and Dani Flores, Brandon and Marilyn Oquendo, Ryan and Chanti Derrick, Tom and Bonnie, Lucio and Ángeles, Yi-Ling, Claudia Acosta, Sonia, Lily, Jennifer Lee, Tom, Alice, Rebecca, Renee, Kelly, Julia, Rose, Lorena, Su, Justina, Carlota, Anaiz, Elizabeth, Preston, Shanell, Dhaval, John, Joanne, Yesusa, Carrie, Elly, Luisa, for all the shepherding. Thank you for standing firm in The Lord. *So then* *my brothers, beloved and longed for, my joy and crown, in the same way stand firm in the Lord, beloved (Phil. 4:1).*

There are special people that no longer can give me more advices and love but always are alive in my heart and my thoughts; grandma Tula, Don Victor and Barbara Lynch. *This is my comfort in my affliction, for Your word has enlivened me (Psalm 119:50).*

Cinthia, Iris, Eloisa, Zeidi, Diana, Lupe, Emna, Vera thanks for letting me be part of your families, for always being there sharing your stories, your joys, for giving me your beautiful friendship. Elery, thanks for the good wishes and amity. Gerardo, thank you for your support, encouragement, advices and friendship. Orlando, thanks for taking the time to help me in semiconductor devices and principally thank you for being such a good friend. Cinda, Delia, Irak, Marco, Yara, Oscar Romero, Victor González, Oscar Tapia, Gaudencio, Erick Mario, Ignacio Rocha, Zagoya, Julio, Jorge and Irene, Ruben, Oscar Addiel, Miguel Tlaxcalteco, Toño, Carolina Rosas, Daniela, Carolina García, Emmaly, Vanessa, Wilson, Ángel, Gaby, Miguel Hernández, Edel, Ricardo, Gisela, Erika, Luis Alberto, Rafael, Fernando, José Carmona, Lyda, Loth, and Ramón, I treasure every moment with you. *He who walks with wise men will be wise (Prov. 13:20).*

Doña Martha and Don Ernesto Cortes, since I got to your home you have treated me as a family, I love you both and your family. I cannot adequately express how grateful I am to you. Doña Male and Mimi, thank you for always supporting the students. *It is more blessed to give than receive (Act 20:35).*

Agradecimientos

Estoy agradecida con Dios por todas las cosas que Él ha hecho en mi vida. *Diré yo a Jehová: Refugio mío, y fortaleza mía; Mi Dios, en quien confío (Sal. 91:2)*.

Le agradezco a CONACYT por otorgarme las becas núms. 224191, 290842 y 290935. También le doy a gracias a todo el personal del instituto INAOE: investigadores, personal administrativo, secretarias, guardias de seguridad, personal de limpieza, trabajadores de la biblioteca y personal del comedor. *Gracias doy a mi Dios siempre por vosotros (1 Cor. 1:4)*.

Yo no llegué a este punto de finalizar el doctorado sin ayuda. Hay muchas personas que han contribuido de alguna forma, le agradezco a cada uno de ellas. Primero, le agradezco a mi asesora de las tesis de maestría y doctorado, la Dra. Gordana Jovanovic Dolecek, quien me guió con sabiduría. Aprendí mucho de ella. También, estoy agradecida con mis sinodales: Dr. Uwe Meyer-Baese, Dr. Alfonso Fernández Vázquez, Dr. Francisco Javier De la Hidalga Wade, Dr. Luis Hernández Martínez y Dr. Jorge Roberto Zurita Sánchez por todo el apoyo y por los consejos. Así mismo, le agradezco a mis profesores por ayudarme y enseñarme a través de los cursos de maestría y doctorado: M.C. Jacobo Meza, Dr. Celso Gutiérrez, Dr. Pedro Rosales, Dr. Roberto Murphy, Dr. Juan Manuel Ramírez, Dr. Arturo Sarmiento, Dr. Reydezel Torres, Dr.

Esteban Tlelo, Dr. Ignacio Zaldívar, Dr. José de Jesús Rangel. *Y os daré pastores según mi corazón, que os apacienten con ciencia y con inteligencia (Jer. 3:15).*

Alguien especial que ha sido más que un guía, maestro, compañero, amigo, no tengo palabras para apreciar su maravillosa compañía; mi otra mitad David Ernesto Troncoso Romero, gracias por todos los consejos, los desvelos ayudándome y por motivarme en cada momento. *Y los dos serán una sola carne; así que ya no son dos, sino una sola carne (Mc. 10:8).*

También, le agradezco a toda mi familia, mi mamá Geno, tía Chuy, abuelito Benito, mis hermanos Moy, Luz y Luis, mi suegra Ricarda, mi cuñada Anita, mis sobrinos Kevin, Isaac y Elías, Joana y Ruth, mis primas Francis, Marisol, Lupe, tío Quiri, abuelita Romana y demás familiares por toda la paciencia, apoyo personal y por perdonarme no haber estado en momentos especiales. *Y sobre todas estas cosas vestíos de amor, que es el vínculo de la perfección (Col. 3:14).*

Estoy más que agradecida con Mike Lynch, Raúl y María Oquendo, Art y Adele TerMorshuizen, Laszlo y Jozann Roszol, Aaron y Charis Chen, Daniel y Shifrah Combiths, Philip y Julia Holly, Yin y Lin Zhang, Jim y Pam Waldrup, Camille y Joey Calascibetta, Viridiana y David Colon, Thomas y Dani Flores, Brandon y Marilyn Oquendo, Ryan y Chanti Derrick, Tom y Bonnie, Lucio y Ángeles, Yi-Ling, Claudia Acosta, Sonia, Lily, Jennifer Lee, Tom, Alice, Rebecca, Renee, Kelly, Julia, Rose, Lorena, Su, Justina, Carlota, Anaiz, Elizabeth, Preston, Shanell, Dhaval, John, Joanne, Yesusa, Carrie, Elly, Luisa, por todo el pastoreo. Gracias por permanecer firmes en el Señor. *Así que, hermanos míos amados y deseados, gozo y corona mía, estad así firme en el Señor, amados (Fil. 4:1).*

Hay personas especiales que ya no me pueden dar más consejos y cariño, pero siempre están vivas en mi corazón y mis pensamientos; abuelita Tula, Don Victor y Barbara Lynch. *Éste es mi consuelo en la aflicción, que Tu palabra me ha vivificado (Sal. 119:50).*

Cinthia, Iris, Eloisa, Zeidi, Diana, Lupe, Emna, Vera gracias por dejarme ser parte de sus familias, por siempre estar ahí compartiendo sus historias, sus alegrías y por darme su hermosa amistad. Elery, gracias por los buenos deseos y amistad. Gerardo, gracias por tu apoyo, ánimo, consejos y amistad. Orlando, gracias por tomarte el tiempo para ayudarme en dispositivos semiconductores y principalemente gracias por ser un buen amigo. Cinda, Delia, Irak, Marco, Yara, Oscar Romero, Victor González, Oscar Tapia, Gaudencio, Erick Mario, Ignacio Rocha, Zagoya, Julio, Jorge e Irene, Ruben, Oscar Addiel, Miguel Tlaxcalteco, Toño, Carolina Rosas, Daniela, Carolina García, Emmaly, Vanessa, Wilson, Ángel, Gaby, Miguel Hernández, Edel, Ricardo, Gisela, Erika, Luis Alberto, Rafael, Fernando, José Carmona, Lyda, Loth, y Ramón, atesoro cada momento con ustedes. *El que anda con sabios será sabio (Pr. 13:20).*

Doña Martha and Don Ernesto Cortes, desde que llegue a su casa me han tratado como familia, los quiero y a su familia. No tengo como expresar lo agradecida que estoy. Doña Male y Mimi, gracias por ayudar a los estudiantes. *Más bienaventurado es dar que recibir (Hch. 20:35).*

Contents

Chapter 1

Introduction

Digital Signal Processing (DSP) has multiple applications, for example in mobile communications, audio processing, image processing or instrumentation, among others [1]-[4]. Because of that, the popularity of DSP has increased in the last years. Only in 2016, around 7 billion of subscriptions to mobile communications were calculated, which represents the 96% of world's population [5]. Cell phones, hard drives, Digital Subscriber Line (DSL), satellite television, Global Navigation Satellite System (GNSS), are examples of communication systems where data are digitally transmitted [6]-[8]. In these systems, digital filters are widely used and play an important role.

A digital filter is a system whose objectives are improving the quality of the signal, extracting information of the signals or separating previously combined signal components, among others. Due to these reasons, the filter is a vital block in DSP [6], [9]-[10]. Since today´s society increasingly use mobile devices which are battery-powered, it is desirable that the battery charge lasts as long as possible [6], [10]. The high demand of low power consumption in portable devices restricts the permitted number of hardware components. Because of this, the current research is focused on the development of new digital filter techniques that meet characteristics like low power consumption and low utilization of hardware resources [11].

In wireless communication systems, successive generations have increased their bandwidth and data rates. Current systems offer 100 M-bit/sec data rates in 20 MHz bandwidth links, but future generations of wireless systems are expected to offer 1 G-bit/sec data rates in 500 MHz bandwidth links [12]. Moreover, in the future it is expected to perform most of the signal processing in the digital domain, being the digital filters an important part of this processing. Nevertheless, taking into consideration the high rates at which these systems would operate, the filtering tasks can saturate the capacity of the hardware processing. Additionally, digital filters can be computationally expensive (in terms of required arithmetic operations to be implemented) causing the reduction of lifespan of the batteries. For this reason, developing algorithms and architectures of high-performance digital filters is necessary. These filters should be able to operate at higher sampling rates, with less number of arithmetic operations and with as low as possible power consumption, so they can function in such communications systems.

Finite Impulse Response (FIR) filters are preferred in communications although they have higher order than the Infinite Impulse Response (IIR) filters for the same magnitude response specifications. This preference is due, among other characteristics, to the fact that the FIR filters have guaranteed stability, can have lineal phase and can perform less arithmetic operations in multirate blocks due to their simple and direct polyphase decomposition.

The transfer function of a FIR filter is given by

$$H(z) = \sum_{n=0}^{N} h(n) \cdot z^{-n} \, , \tag{1.1}$$

where $h(n)$ are the filter coefficients and N is the order of the filter. Particularly, when the filter has linear phase, the condition $h(n)=\pm h(N-n)$ holds. If the sign is positive, the condition is called symmetry, or anti-symmetry if the sign is negative.

The order of linear-phase FIR filters depends on the magnitude response specifications, i.e., the band edge frequencies (passband edge, ω_p, and stopband edge ω_s) and the allowed deviation from the ideal amplitude in the bands of interest (stopband deviation, δ_p, and passband deviation, δ_s). The formula for estimating the minimum order necessary to satisfy a particular specification, given in [13], is

$$N \approx \frac{-20\log\left(\sqrt{\delta_p\delta_s}\right)-13}{14.6(\Delta\omega/2\pi)}, \tag{1.2}$$

where $\Delta\omega$ is the transition band of the filter, i.e., the difference between the passband edge and the stopband edge. As we see, the order is inversely proportional to the transition band.

The computational complexity of a digital FIR filter is given in terms of the number of multipliers, *Mult*, and the number of adders, *Sum*, which can be estimated as follows:

$$Sum = N, \tag{1.3}$$

$$Mult = \begin{cases} \left\lfloor \frac{N}{2} \right\rfloor + 1; & \text{linear phase,} \\ N; & \text{otherwise.} \end{cases} \tag{1.4}$$

Clearly, the computational complexity is proportional to the order of the filter. Thus, from (1.2), (1.3) and (1.4), we easily can see that the filter becomes more computationally complex when its transition band becomes narrower. The multipliers are the most expensive elements as they increase the area utilization, latency and power consumption [11]. Figure 1.1 shows two low-pass FIR filters with the same deviation specifications but different transition bands, namely, $\Delta\omega_1$ and $\Delta\omega_2$, along with the ideal magnitude response. The filter with wider transition band has a lower order (N_1 = 10), but its magnitude response is less close to the ideal response. The filter with the best magnitude response between these two needs an order N_2 = 36, which implies a higher computational cost.

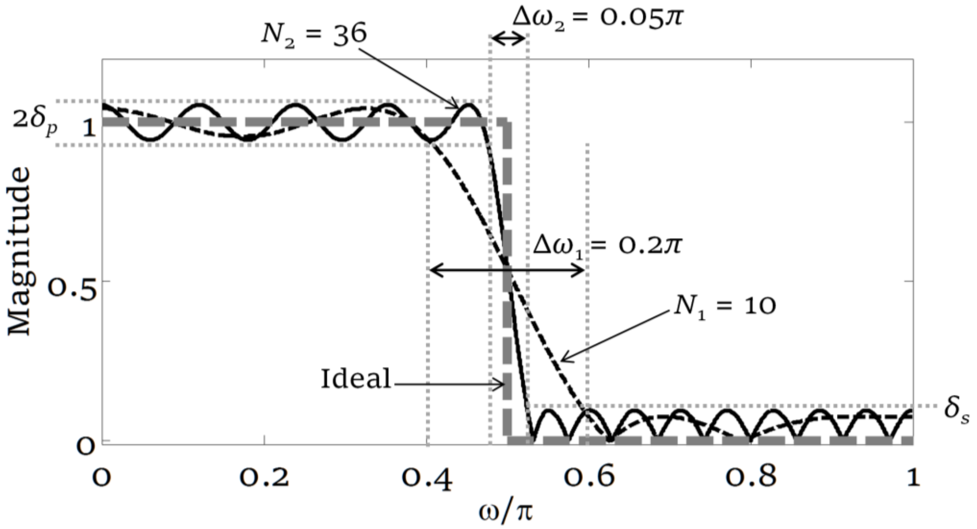


Figure 1.1. Magnitude response of two digital low pass filters.

When the classical design methods are employed, digital filters are usually designed by minimizing the maximum error in their passband and stopband deviations (minimax criterion). The resulting filter satisfies the desired magnitude response characteristics with the minimum order [14]. However, the use of these classical methods can result in filters with high order and high computational complexity, which is inconvenient in high performance communication systems.

1.1 Objective

The main purpose of this thesis is the investigation of effective methods to design low-complexity FIR filters. This research is based, on the one hand, in decomposing the overall filter in simple subfilters and, on the other hand, in simplifying the constant coefficients of the filters by eliminating multipliers. These are the most effective solution schemes according to the state of the art.

The following is a review of some special FIR filters with great demand in communications that can benefit from the research developed here.

Multirate filters: In several applications it is necessary to decrease or to increase the sampling rate of a signal. These processes are respectively known as downsampling or upsampling, and they may affect the information contained in the signal if that signal is not properly filtered. Filtering a signal and then applying downsampling is known as decimation, whereas applying upsampling and then filtering a signal is known as interpolation. Figure 1.2a shows the resulting samples of a decimated signal with dowsampling factor equal to 2. The reduction of the sampling rate makes the aliasing effect to appear in the signal spectrum. The aliasing consists in the insertion of undesirable information inside of the band of interest of a signal. Figure 1.2b shows how it affects the spectrum. With the aim of protecting the information prior to downsampling, decimation filters (commonly known as anti-aliasing filters) must be used [15]. Figure 1.3 illustrates a proper decimation process, which consists in a decimation filter cascaded with a downsampling stage.

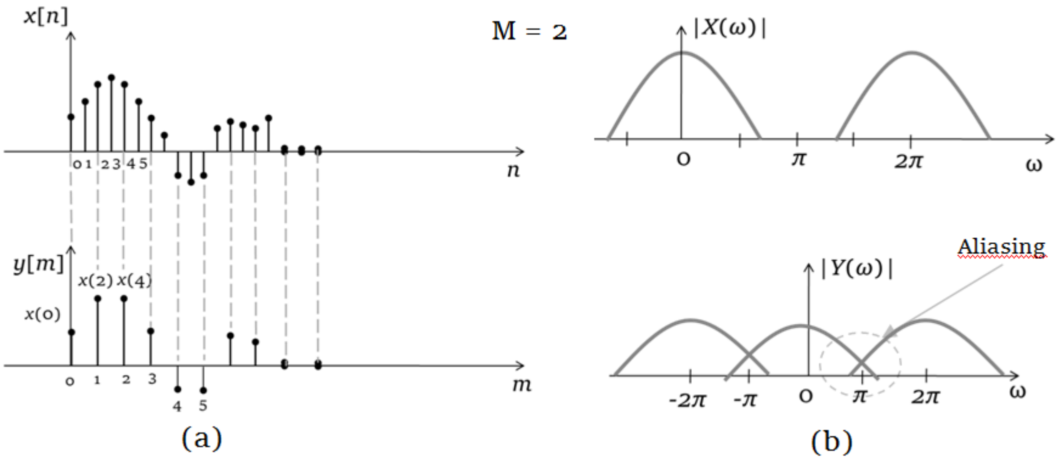


Figure 1.2. (a) Samples of a downsampled signal and (b) spectrum of a downsampled signal.

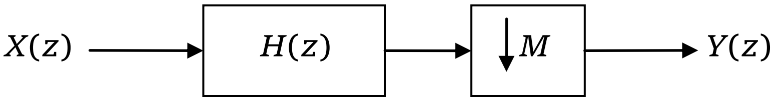


Figure 1.3. Structure of decimation process.

On the other hand, in general terms, the interpolation consists in the calculation of new samples between the existing samples of a signal, see Figure 1.4a. Usually, the interpolation is needed to increase the sample rate of a signal. Due to the increased sampling rate, replicas of the spectrum of the original signal appear. This is known as imaging, as shown in Figure 1.4b. To remove these unwanted copies, a low pass filter is used, which is called interpolator filter [16], Figure 1.5. When the replicas of the spectrum of the original signal are removed, the resulting effect is that new samples appear. These samples are points that interpolate the original samples. The interpolation process is dual to the decimation process, and the methods to design decimators can be straightforwardly extended to design interpolators.

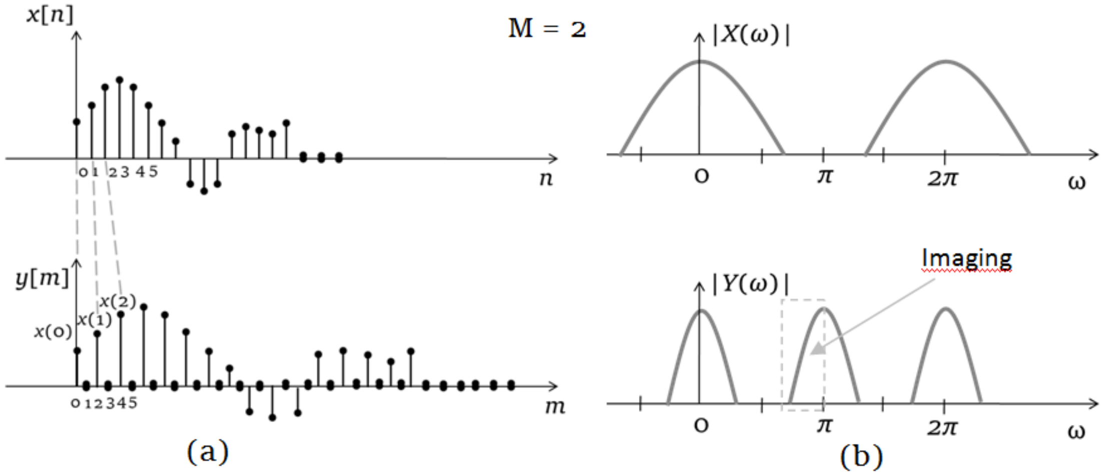


Figure 1.4. (a) Samples of an upsampled signal and (b) spectrum of an upsampled signal.

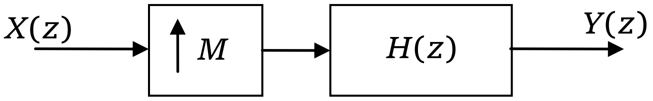


Figure 1.5. Structure of interpolation process.

Filters with constant coefficients: Examples of filters with constant coefficients $h(n)$ are frequency-selective filters, pulse-shaping filters, or minimum-phase filters, among others. Frequency-selective filters pass certain frequency components of the input signal and attenuate other components of that signal according to a given specification. An example of this is the filter whose magnitude response is shown in Figure 1. A particular case of these filters are pulse-shaping filters, which are used to avoid the intersymbol interference (ISI). In this case, the impulse response of the filter shapes the form of every pulse to be transmitted, such that the pulse can be detected at the receiver and simultaneously its frequency response characteristic can fit into a spectral mask previously specified. Thus, pulse-shaping filters are applied to avoid the distortion problems for high speed transmissions

[17]. On the other hand, a Minimum-Phase (MP) FIR filter has its zeros on or inside the unit circle and this characteristic makes it to have the minimum group delay among other filters with the same magnitude response, at expenses of a non-linear phase response [18]-[20]. Thus, MP FIR filters find application in cases where high group delay, usually caused by Linear-Phase (LP) FIR filters, is not allowed. These cases include communication systems or audio processing, among others.

1.2 Contributions

The following contributions have been developed in this thesis.

- A mathematical proof that a filter formed with cascaded cosine subfilters in a sharpening scheme based on Chebyshev polynomials can have Minimum Phase (MP) characteristic. The demonstration that cascaded and expanded Chebyshev-sharpened cosine filters are also MP filters is provided as well, and it is shown that they can have a lower group delay for similar magnitude characteristics in comparison with traditional cascaded expanded cosine filters. Improvements in the group delay at the cost of a slight increase of usage of hardware resources can be achieved. Moreover, for an application of a low-delay decimation filter, the proposed scheme exhibits lower group delay, less computational complexity (in Additions Per Output Sample, APOS) and slightly less usage of hardware elements.

- A method to design low-complexity wide-band compensators to improve the passband characteristic of comb and comb-based filters sharpened with Chebyshev polynomials. The proposed method is based on the amplitude transformation approach, and a simple formula to obtain the coefficients of the compensator is also provided. Design examples and comparisons show that the proposed compensation filters have better frequency characteristics compared to other wide-band compensators recently presented in the literature.

- A method to design comb-based decimation filters with improved magnitude response characteristics, based on compensation filters and Chebyshev polynomials. It is shown that the filters designed with the proposed method exhibit better characteristics than the traditional comb filter and other recent methods from literature.
- A comb-based decimator that consists of an area-efficient structure aided with an embedded simplified Chebyshev-sharpened section. The proposed scheme improves the worst-case aliasing rejection of comb filters and preserves a low-complexity design that requires fewer hardware resources and consumes less power. The proposed system exhibits regularity, a desirable characteristic not present in other comb-based recent methods from literature that have pursued the same goals.
- A method to design comb-based decimation filters with improved magnitude response characteristics, which consists in applying the Hartnett-Boudreaux sharpening technique (so-called improved sharpening) to simultaneously increase the worst case attenuation and correct the droop in the passband region. The coefficients of the sharpening

polynomials are expressed as Sum of Power of Two (SPT), leading to multiplierless implementations.

- Comb-based decimation architectures split in stages, based on the Harnett-Boudreaux sharpening. The non-recursive comb-based decimation architecture is employed when the downsampling factor is a power of two, whereas two and three stages are employed for other composite downsampling factors, with non-recursive structure in the first stage and recursive structure in subsequent stages. To improve the passband characteristic, a simple compensator is applied in the last stage. Then the Hartnett-Boudreaux sharpening technique is applied to decrease the passband droop induced by the comb filter placed in the first stage. As a result, computationally efficient comb-based decimation filters are obtained with better magnitude characteristics than previous proposed sharpening methods.
- New theoretical lower bounds for the number of operators needed in fixed-point constant multiplication blocks. The constant multipliers are constructed with the shift-and-add approach, where every arithmetic operation is pipelined, and with the generalization that n-input pipelined additions/subtractions are allowed, along with pure pipelining registers. These lower bounds, tighter than the state of the art theoretical limits, are particularly useful in early design stages for a quick assessment in the hardware utilization of low-cost constant multiplication blocks implemented in the newest families of Field Programmable Gate Array (FPGA) integrated circuits.

1.3 Organization

This thesis is organized in five chapters. An introduction on the research developed here is given in Chapter 1. Chapter 2 presents a review of the state of the art and introduces the techniques used as a basis to carry out this investigation. The proposed methods and architectures that employ comb and cosine filters as basic building blocks are detailed in Chapter 3. Then, Chapter 4 presents the proposed contribution on the implementation of the constant multiplications as a network of additions and shifts, namely, the novel theoretical lower bounds for the number of pipelined operations that are needed in Single Constant Multiplication (SCM) and Multiple Constant Multiplication (MCM) blocks. Finally, Chapter 5 provides the general conclusions and suggestions for future research.

1.4 References

[1] Huang, S., Tian, L., Ma, X. and Wei, Y. "A reconfigurable sound wave decomposition filterbank for hearing aids based on nonlinear transformation," *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 10, No. 2, pp. 487- 496, 2016.

[2] Edwards, J."Signal Processing drives medical sensor revolution," *IEEE Signal Processing Magazine*, Vol. 32, No. 2, pp. 12- 15, 2015.

[3] Rakhshanfar, M. and Amer, M. A."Low-frecuency image noise removal using white noise filter," *IEEE International Conference on Image Processing (ICIP)*, pp. 1973- 1977, 2016.

[4] Xia, W., Wen, Y., Foh, C. H., Niyato, D. and Xie, H. "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 1, pp. 27- 51, 2015.

[5] Sanou, B. *Information and Communication Technologies: Fact and Figures*, International Telecommunications Union, 2016.

[6] Vinod, A. P. and Smitha, K. G. "A low complexity reconfigurable multistage channel filter architecture for resource-constrained Software Radio handsets," *Journal of Signal Processing Systems*, Vol. 62, No. 2, pp. 217-231, 2011.

[7] Ashrafi, A. "Optimized linear phase square-root Nyquist FIR filters for CDMA IS-95 and UMTS standards," *Signal Processing*, Vol. 93, No. 4, pp. 866- 873, 2013.

[8] He, Z., Hu, Y., Wang, K., Wu, J., Hou, J. and Ma, L. "A novel CIC decimation filter for GNSS receiver based on software defined radio," *7th. Int. Conf. Wireless Communications, Networking and Mobile Computing*, pp. 1-4, 2011.

[9] Sukittanon, S. and Potts, J. "Mobile digital filter design toolbox," *Proceedings of IEEE Southeastcon*, pp. 1-4, 2012.

[10] Wu, J., Zhang, Y., Zukerman, M. and Yung E. K. "Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey," *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 2, pp. 803- 826, 2015.

[11] Aksoy, L., Flores, P. and Monteiro, J. "A tutorial on multiplierless design of FIR filters: algorithms and architectures," *Circ. Syst. Signal Process.* , Vol. 33, No. 6, pp. 1689-1719, 2014.

[12] Chen, X., Harris, F. J., Venosa, E. and Rao, B. D. "Non maximally decimated analysis/synthesis filter banks: applications in wideband digital filtering," *IEEE Transactions on Signal Processing*, Vol. 62, No. 4, pp. 852-867, 2014.

[13] Kaiser, J. F. “Non-recursive digital filter design using *I*o-sinh window function,” *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 20-23, April 1974.

[14] Johansson, H. and Gustafsson, O. “Two rate based structures for computationally efficient wide-band FIR systems,” in *Digital Filters and Signal Processing*, Fausto Pedro García Márquez (Ed.), InTech, 2013.

[15] Hogenauer, E. “An economical class of digital filters for decimation and interpolation, ” *IEEE Trans. Acoust., Speech, Signal Process*, ASSP-29, p. 155-162, 1981.

[16] Awan, M. U. R. and Koch, P. “Combined matched filter and arbitrary interpolator for symbol timing synchronization in SDR receivers,” *IEEE International Symposium on Design and Diagnostics of Electronics Circuits and Systems*, pp. 153- 156, 2010.

[17] Ashrafi, A. and Harris, F. J. “A novel square-root Nyquist filter design with prescribed ISI energy,” *Signal Processing*, Vol. 93, pp. 2626- 2635, 2013.

[18] Pei, S.-C. and Lin, H.-S. “Minimum-phase FIR filter design using real cepstrum,” *IEEE Trans. Circ. and Syst.-II*, vol. 53, no. 10, pp. 1113-1117, Oct. 2006.

[19] Okuda, M., Ikehara, M. and Takahashi, S. “Design of equiripple minimum phase FIR filters with ripple ratio control,” *IEICE Trans. on Fundamentals of Electronics, Communications And Computer Science*, vol. E89-A, no. 3, pp. 751-756, Mar. 2006.

[20] Dolecek, G. J. and Dolecek, V. “Application of Rouche’s theorem for MP filter design,” *Applied Mathematics and Computation*, no. 211, pp. 329-335, 2009.

Chapter 2

Review of techniques for FIR filter design

This section presents a selection of recent methods to design FIR digital filters with great demand in communications. These methods have been efficient because they generate filters with a minimum error in the frequency response and with smaller number of arithmetic operations in comparison with the classic methods. Among these techniques, the ones used as a basis to develop the proposals of this thesis are emphasized. Sections 2.1 and 2.2 provide, respectively, an overview of multirate and subfilter-based techniques. Finally, Section 2.3 details the methods related to the proposals introduced in this thesis.

2.1 Multirate techniques

Multirate systems are those that use multiple sampling frequencies in the processing of digital signals. It has been proved that using multirate techniques in the design of a filter generates a reduction in the number of adders and multipliers required for its implementation [1]-[12]. There are several techniques in digital signal processing available to optimize multirate filters. For example, for M-th band FIR filters design, an algorithm was developed in [1] to optimize a polyphase structure based on two stages for different integer sampling rate conversion. It was demonstrated in that scheme that conversions by odd factors are more efficient than conversions by even factors. A new design method to design differentiators and wide-band filters, that offers a dramatic complexity reduction, was presented in [2]-[3]. In this approach there is a two-frequencies system that takes advantage of the Frequency Response Masking Technique (FRM) to accomplish sharp transition bands with reduced computational load.

A common application of multirate techniques is in filter bank systems [4]-[7]. Method [4] employs Fast Fourier Transform (FFT) and its inverse to achieve computationally-efficient filter banks, whereas a recent design method of cosine modulated filter bank (CMFM) and transmultiplexers uses the Interpolated Finite Impulse Response (IFIR) technique to design the prototype filter [5]. The use of nature-inspired metaheuristics for the optimization of coefficients in filter banks and transmultiplexers was proposed in [6]-[7].

Splitting into q stages the decimation and interpolation processes by an integer D is a proper strategy for computational efficiency, i.e., D is factorized in q factors. For example, for $q = 2$, we have $D = M \times R$. The Cascaded Integrator-Comb (CIC) structure can be used in the first stage with downsampling by M and is efficient in terms of chip area but requires integrators working at high rate, thus having high power consumption. Because of this, multi-stage comb-based decimation schemes have gained great popularity. In methods [8]-[9] the value of q is 3 (i.e., $M = M_1 \times M_2$), while q greater than 3 is set in the works [10]-[12], where D is constrained to be a power of 2 or a power of 3. By using multistage structures, the first-stage filter can be implemented in a non-recursive form and the polyphase decomposition can be applied, thus resulting in power savings at expenses of an increase of chip area.

2.2 Techniques based on simple filters

The use of simple subfilters to design FIR filters has been demonstrated to be efficient. The decomposition of an overall filter into simple subfilters allows to obtain filters with narrow transition band and lower number of arithmetic operations than the direct methods. Thus, these methods are ubiquitous in different applications where the computational complexity must be reduced.

The FRM technique has received considerable attention for digital filters design due to its capabilities. The principal blocks in the FRM technique are the model filters and the masking filters. The model filters are also known as sparse filters (or filters with sparse coefficients) because they have many zero-valued coefficients. These filters provide the shape of the transition band of the overall filter at expenses of introducing unwanted frequency response images in the bands of interest, whereas the masking subfilters cancel these unwanted images. Recent improvements to the FRM method have been introduced in [13]-[14]. A FRM-based design method where the model filter was implemented in hybrid form, allowing the reduction of critical path with low computational complexity and low utilization of hardware resources in the design, was presented in [13]. On the other hand, a unified design framework based on a convex-concave optimization procedure has been recently provided in [14].

The Frequency Transformation (FT) to design linear phase Type I FIR filters with narrow transition band and small error in the passbands and stopbands is another efficient method based in subfilters. The total filter is implemented as a cascaded interconnection of identical subfilters. This interconnection includes structural coefficients that appear in parallel to the subfilters. The method consists in mapping into the bands of interest the amplitude response of a prototype filter, which generates the structural coefficients, using the amplitude response of the subfilter as a mapping function. Recently, a method to design Hilbert transformers based on this technique, which results in few multipliers, was presented in [15], where the FT method is applied in nested levels. On the other hand, a unified view of the frequency transformation method for FIR filters was proposed in [16], where the frequency response of the overall filter is considered as a function composed by simpler identical functions.

2.3 Techniques related to the proposals of this thesis

During the development of this thesis some methods were a main tool to get the resulting proposals:

a) The sharpening methods, an special case of frequency transformation methods, were employed and modified to obtain excellent trade-offs between the computational complexity and the improvement in the magnitude response of FIR decimation filters.

b) The multiplierless methods influenced the elaboration of the new theoretical lower bounds for the number of operations required in Pipelined Single Constant Multiplications (PSCM) and Pipelined Multiple Constant Multiplications (PMCM).

Subsections 2.3.1 and 2.3.2 present the respective fundamentals and state of the art of the aforementioned methods.

2. 3. 1 Sharpening techniques

The Sharpening technique improves the magnitude characteristics of a filter, i.e., decreases the error in the passband region and improves the attenuation in the stopband region, by cascading identical copies of that filter, and including structural coefficients that are connected in parallel to these cascaded filters. The sharpening technique has been proved to be successful in the design of digital filters. The resulting filters save multipliers significantly compared with the direct form designs.

The first method known as sharpening technique was proposed in [17] by Kaiser and Hamming, where the structural coefficients are obtained from simple polynomials referred as Amplitude Change Functions (ACFs). Many applications of the sharpening technique have been made to FIR filter design, particularly for comb-based decimation filters, corroborating the effectiveness of this method, see for example [18]-[22]. Years later, a method based on the sharpening of Kaiser and Hamming was proposed by Hartnett and Boudreaux [23]. In this approach, called Improved Sharpening, there are more design parameters that allow to generate better magnitude response improvements in comparison with the traditional sharpening.

In the improved sharpening, which is a generalization of the traditional sharpening, the ACF is a polynomial denoted by $P_{m,n,\sigma,\delta}(x)$ which maps the amplitude x into a different amplitude $y = P_{m,n,\sigma,\delta}(x)$. In this notation, x is the amplitude response of the simple filter to be improved and y is the resulting amplitude response after cascading the simple filter several times (the number of cascaded sections is given by the degree of the ACF, and the structural coefficients are the coefficients of the ACF). The improvement in amplitudes near to the passband increases with m, the order of tangency of the ACF at the point $(x, y) = (1, 1)$ to a line with slope equal to σ. Similarly, the improvement in amplitudes near to the stopband increases with n, the order of tangency of the ACF at the point $(x, y) = (0, 0)$ to a line with slope equal to δ.

The desired piecewise linear ACF is illustrated in Figure 2.1 along with the real ACF, i.e., the polynomial $P_{m,n,\sigma,\delta}(x)$. In that figure, x_{pl} and x_{pu} are, respectively, the minimum and maximum amplitude in the passband of the original filter, and x_{sl} and x_{su} are the minimum and maximum amplitude in the stopband of the same filter, respectively. In the same way, y_{pl}, y_{pu}, y_{sl}, and y_{su} are the minimum and maximum amplitudes in the passband and the minimum and maximum amplitudes in the stopband of the sharpened filter, respectively.

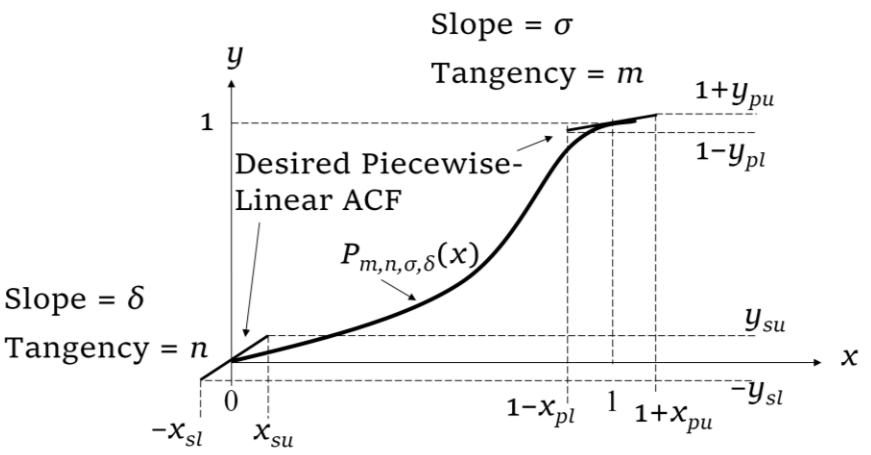


Figure 2.1. The Amplitude Change Function (ACF) given as $P_{m,n,\delta,\sigma}(x)$.

A general formula was deduced in [24] to obtain directly the desired ACF from the design parameters. The polynomial $P_{m,n,\sigma,\delta}(x)$ is given as

$$P_{m,n,\sigma,\delta}(x) = \delta x + \sum_{j=n+1}^{R} (\alpha_{j,0} - \sigma\alpha_{j,1} - \delta\alpha_{j,2})x^{j}\,, \tag{2.1}$$

with $R = n + m + 1$, and

$$\begin{aligned}
\alpha_{j,0} &= \sum_{i=n+1}^{j} (-1)^{j-i} \binom{R}{j}\binom{j}{i}, \\
\alpha_{j,1} &= \sum_{i=n+1}^{j} (-1)^{j-i} \binom{R}{j}\binom{j}{i}\left(1-\frac{i}{R}\right), \\
\alpha_{j,2} &= \sum_{i=n+1}^{j} (-1)^{j-i} \binom{R}{j}\binom{j}{i}\frac{i}{R}.
\end{aligned} \tag{2.2}$$

The traditional sharpening is an special case where δ and σ are both equal to zero. Thus, with the parameters σ and δ, the improved sharpening provides more flexibility in the design process.

The Chebyshev sharpening approach was recently introduced in [25] for comb-based decimation filters with integer downsampling factor M. This approach is based on Chebyshev polynomials and allows to obtain equiripple stopbands. The ACF in Chebyshev sharpening is obtained as

$$Q_K(x) = \sum_{k=0}^{K} C_k \cdot \gamma^k \cdot x^k \,, \tag{2.3}$$

with

$$\gamma = 2^{-r} \times \left\lfloor 2^r \times \left(\frac{\sin\left([\frac{2\pi}{M} - \frac{\pi}{MR}]/2\right)}{\sin\left([\frac{2\pi}{M} - \frac{\pi}{MR}] \times M/2\right)} \right) \right\rfloor, \tag{2.4}$$

where C_k is the coefficient of the k-th power of a K-th degree Chebyshev polynomial of first kind, R is the integer downsampling factor of the decimation stage that is placed after the Chebyshev-sharpened decimator (it is usually $R = 2$), and r is the precision for the fractional part of γ. A new method for two-stage comb-based decimation filters that uses Chebyshev sharpening technique to improve the magnitude response characteristics of the traditional comb filter was presented in

[26]. In [27], the Chebyshev sharpening approach was applied to linear-phase FIR filters design. The resulting filters present equiripple stopbands and the subfilters are constituted by small integer coefficients.

Methods to design filters with improved magnitude characteristics using sharpening approaches are a current research topic specially useful in comb-based decimation filters (i.e., CIC-based structures). In this context, besides of the aforementioned sharpening methods, other sharpening polynomials, i.e., ACFs, have been introduced in [28] and recently in [29]-[34]. These ACFs can not be explicitly expressed with simple formulas, but they have to be found via optimization. An useful implementation structure for sharpened CIC decimators was presented by Saramaki-Ritoniemi in [28], and it has been the basis for all sharpened CIC decimators. Without loss of generality, Figure 2.2(a) illustrates the direct structure for a sharpened comb filter followed by a downsampling factor M. Its transfer function is

$$H(z) = \sum_{k=0}^{K} \beta_k \times \left(z^{-1} \times \frac{1 - z^{-M}}{1 - z^{-1}} \right)^{2k} \times z^{-(K-k)(1+M)}, \qquad (2.5)$$

where β_k represents the coefficient of the k-th power of the sharpening polynomial. The resulting CIC-based decimation structure is shown in Figure 2.2b, which is obtained after applying multirate identities.

2.3.2 Multiplierless techniques

In all the digital signal processing based systems, multiplication of digital signals by a single constant (Single Constant Multiplication, SCM) or by multiple constants (Multiple Constant Multiplication, MCM) is a common operation, found for example in digital filtering, Discrete

Fourier Transform (DFT), Discrete Cosine Transform (DCT), among others [35]-[39]. There is currently abundant research activity focused on developing efficient blocks of multiplications by constants where multipliers, the most power- and area-consuming elements in a DSP arithmetic block, are avoided since their full flexibility is not needed [35]-[57]. In these cases, multiplications are performed using only additions and subtractions, and only scaling by powers of two is allowed. These powers of two are implemented using hardwired shifts and therefore are considered with no cost. This scheme of constant multiplications is so-called shift-and-add multiplication or multiplierless multiplication.

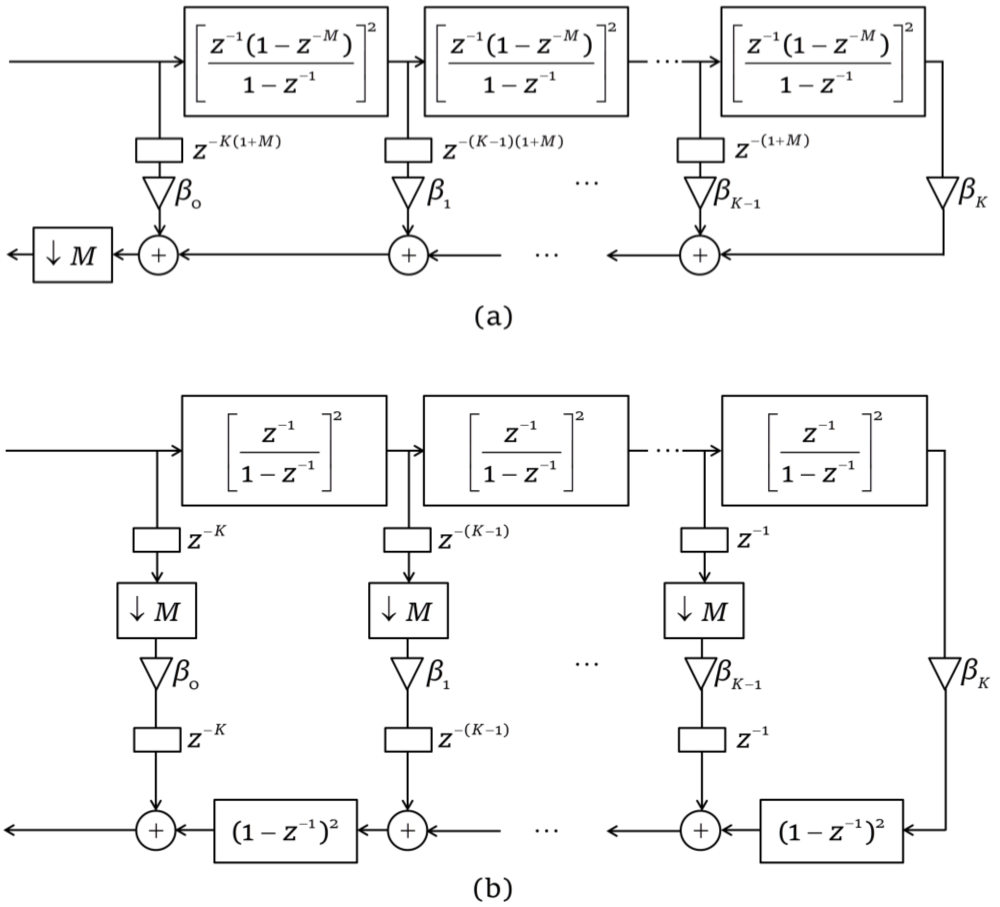


Figure 2.2. (a) Direct structure for a sharpened CIC filter. (b) Efficient implementation structure of a sharpened CIC filter.

The SCM case is when an input is multiplied by a constant coefficient, see Figure 2.3(a), and the MCM operation is when an input is multiplied by a set of constant coefficients, see Figure 2.3(b). Theoretical lower bounds for the number of adders and for the number of depth levels, i.e., the maximum number of serially connected adders (also known as the critical path), in SCM, MCM and other constant multiplication blocks that are constructed with two-input adders under the shift-and-add scheme have been presented in [53], and an extension to these lower bounds in the SCM case was recently given in [54].

The constant multiplications referred here are expressed in fixed-point arithmetic because implementations in this number representation have higher speed and lower cost, thus being usually employed in DSP algorithms [37]-[57].

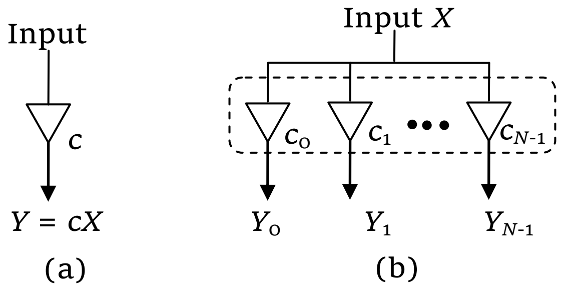


Figure 2.3. Block diagram of constant multiplications: (a) *SCM* and (b) *MCM*.

Only integer, positive, odd constants are considered since this is a useful simplification that does not affect the formulation of constant multiplication problems. In this sense, a constant can be expressed simply in binary form, as follows,

$$c = \sum_{i=0}^{B-1} b_i 2^i \,, \tag{2.6}$$

where $b_i \in \{0, 1\}$ is the i-th bit and B is the word-length [54]. We can express a product of a variable input X by a constant c with the shift-and-add approach using the binary representation of that constant to dictate the multiplier structure. For example, the product $47X$, with $47 = 2^5 + 2^3 + 2^2 + 2^1 + 2^0$ (i.e., a binary string "101111"), needs four additions and has a critical path of three additions, as show in Figure 2.4. The implementation cost of a shift-and-add constant multiplier is the number of arithmetic operations since products by powers of two are implemented as hardwired shifts with no practical cost.

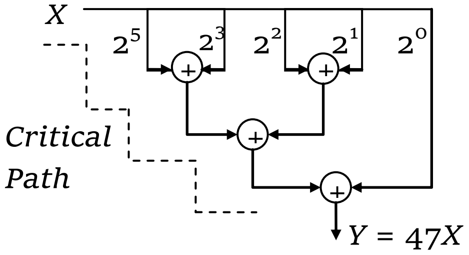


Figure 2.4. Implementation structure of the product $47X$ with constant 47 expressed in binary.

It is worth to highlight that additions and subtractions require practically equal amount of resources in hardware implementation. Hence, Signed Digit (SD) representations of a constant can reduce the aforementioned implementation cost because they employ negative digits, which represent subtractions. An SD representation of a constant is given in the form,

$$c = \sum_{i=0}^{B-1} d_i 2^i , \tag{2.7}$$

where $d_i \in \{-1, 0, 1\}$, with '−1' usually expressed as $\bar{1}$ [55]. Among them, the Canonical Signed Digit (CSD) representation is convenient since its number of non-zero digits is the *Minimum Number of Signed Digits* (MNSD) [54]. Besides, each non-zero digit is followed by at least one zero, which makes the representation unique. The CSD form of a constant can be found from binary by iteratively substituting every string of *k* digits '1' (say, "1111") with a string of *k*−1 digits '0' between a '1' and a '−1' (the string "1111" becomes "$1000\bar{1}$"). In this case, the product 47*X*, with 47 = 2^6 - 2^4 -2^0 (i.e., a CSD string "$10\bar{1}000\bar{1}$"), needs two subtractions and has two operations in its critical path, as shown in Figure 2.5.

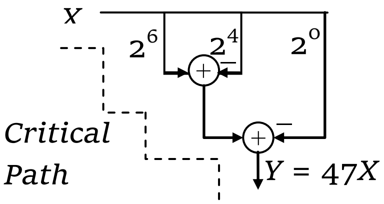


Figure 2.5. Implementation structure of the product 47*X* with constant 47 expressed in CSD.

In a constant multiplication block, the *A*-operation [56] represents two-input addition or subtraction along with shifts, and it is defined as,

$$A_q(u_1,u_2) = \left| 2^{l_1} u_1 + (-1)^{s_2} 2^{l_2} u_2 \right| 2^{-r} , \tag{2.8}$$

where $l_1 \geq 0$, $l_2 \geq 0$ are left shifts, $r \geq 0$ is a right shift, s_2 is a binary value, i.e., $s_2 \in \{0,1\}$, q is the set of parameters (so-called the configuration) of the *A*-operation, i.e., $q = \{l_1, l_2, r, s_2\}$, and u_1, u_2 are odd integers.

An array of interconnected *A*-operations form a SCM or a MCM block. The MCM is built upon SCM because the latter is the simplest case. The SCM array is represented using directed acyclic graphs (DAGs) with the following characteristics [57]:

- The output of each *A*-operation is called *fundamental*.
- For a graph with *m* *A*-operations, there are *m* + 1 vertices and *m* fundamentals.
- Each vertex has an in-degree *n*, except for the input vertex which has in-degree zero.
- A vertex with in-degree *n* corresponds to an *n*-input *A*-operation.
- Each vertex has out-degree larger than or equal to one except for the output vertex which has out-degree zero.
- The constant resulting from the last *A*-operation is *output fundamental* (OF). The constants resulting from previous *A*-operations are *non-output fundamentals* (NOFs).

In the MCM case, there are several OFs.

The Directed Acyclic Graph (DAG) representation is the most useful for saving arithmetic operations because it allows to exploit structures to interconnect *A*-operations that can not be seen in the CSD representation. This expands the opportunity to optimize the constant multiplication blocks. For example, the product 45*X*, with 45 = $2^6 - 2^4 - 2^2 + 2^0$ (i.e., a CSD string "$10\bar{1}0\bar{1}01$"), needs three 2-input additions and has a critical path of two additions, as show in Figure 2.6(a). However, by using the DAG approach, the multiplication 45*X* requires two 2-input additions and has a critical path of two additions. In this case it is possible to factorize the constant in two factors, namely, 5 and 9, as shown in Figure 2.6(b).

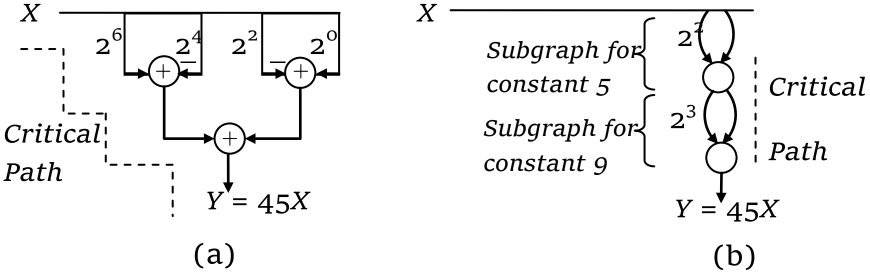


Figure 2.6. Structure of the product 45X (a) constant 45 expressed in CSD and (b) constant 45 in graph representation.

Particularly, in the last two decades many efficient high-level synthesis algorithms have been introduced for the multiplierless design of constant multiplication blocks. The usual cost function to minimize in these algorithms has been the number of arithmetic operations (additions and subtractions) needed to implement the multiplications, which is representative of the computational complexity and the chip area required in that implementation. Nevertheless, the number of operations connected in series, i.e., the number of depth levels forming a critical path, has the main negative impact in the speed and power consumption [41]-[44]. Therefore, substantial research activity has been carried out currently targeting both, Application-Specific Integrated Circuits (ASICs) [45]-[47] and Field-Programmable Gate Arrays (FPGAs) [48]-[52], where the minimization of the number of arithmetic operations subject to a minimum critical path is the ultimate goal.

The design of efficient multiplierless constant multiplication blocks is conjectured to be an NP-complete problem [47]. Thus, the existing algorithms are heuristics that aim to maximize the sharing of partial products. They are generally grouped in two categories based on the search space where they look for a solution.

On the one hand, the Common Sub-expression Elimination (CSE) methods [35], [39]-[41], [46]-[48] define the constants under a number representation, such as binary, Canonical Signed Digit (CSD), or Minimal Signed Digit (MSD). Then, considering possible sub-expressions that can be extracted from the nonzero digits in representations of constants, the “best” sub-expression, generally, the most common, is chosen to be shared among the constant multiplications. The main drawback of these methods is their dependency on a number representation, which can lead to sub-optimal solutions.

On the other hand, the Graph-Based (GB) techniques [36]-[38], [42]-[45], [49]-[52], [56]-[57] are not restricted to any particular number representation and aim to find intermediate sub-expressions that enable to realize the constant multiplications with minimum number of operations. They consider a larger number of realizations of a constant and obtain better solutions than the CSE methods. However, the main drawback of these methods is that they require more computational resources for a proper search due to the larger search space.

2.4 References

[1] Johansson, H. and Gockler, H. “Two-stage-based polyphase structures for arbitrary-integer sampling rate conversion,” *IEEE Transactions on Circuits and Systems II: Express briefs*, vol. 62, no. 5, pp. 486–490, 2015.

[2] Sheikh, U., and Johansson, H. “A class of wide-band linear-phase FIR differentiators using two-rate approach and the frequency-response

masking technique," *IEEE Transactions on Circuits and Systems I: Regular papers*, vol. 58, no. 8, pp. 1827–1839, 2011.

[3] Johansson, H. y Gustafsson, O. "Two rate Based structures for computationally efficient wide-band FIR systems," en *Digital Filters and Signal Processing*, Fausto Pedro García Márquez (Ed.), InTech, 2013.

[4] Renfors, M., Yli-Kaakinen, J. and Harris, F. J. "Analysis and design of efficient and flexible fast-convolution based multirate filter banks," *IEEE Transactions on Signal Processing*, Vol. 62, No. 15, pp. 3768–3783, 2014.

[5] Soni, R. K., Jain, A. and Saxena, R. "A design of IFIR prototype filter for Cosine Modulated filterbank and transmultiplexer," *International Journal of Electronics and Communications*, vol. 67, pp. 130–135, 2013.

[6] Bindiya T. S. and Elias E., "Modified metaheuristic algorithms for the optimal design of multiplier-less non-uniform channel filters," *Circuits, Systems and Signal Processing*, vol. 33, no. 3, pp. 815–837, 2014.

[7] Shaeen K. and Elias E., "Non-uniform cosine modulated filter banks using meta-heuristic algorithms in CSD space," *Elsevier Journal of Advanced Research*, vol. 6, pp. 839–849, 2015.

[8] Dolecek, G. J. and Laddomada, M. "A novel two-stage nonrecursive architecture for the design of generalized comb filters," *Digital Signal Processing*, vol. 22, no. 5, pp. 859-868, 2012.

[9] Salgado G. M., Dolecek, G. J. and De La Rosa J. M., "Low power two-stage comb decimation structures for high decimation factors,"

Analog Integrated Circuits and Signal Processing, vol. 88, no. 2, pp. 245-254, 2016.

[10] Palla A., Meoni G. and Luca F., "Area and power consumption trade-off for sigma-delta decimation filter in mixed signal wearable IC," *IEEE Nordic Circuits and Systems Conference*, pp. 1-4, 2016.

[11] Dolecek, G. J. and Salgado, G. M. "On efficient nonrecursive comb decimator structure for M=3^n," *IEEE Int. Conf. on Communications and Electronics (ICCE)*, pp. 369–372, 2012.

[12] Nasir N. H. *et al*, "Oversampled sigma-delta ADC decimation filter: design techniques, challenges trade-offs and optimization," *IEEE International Conf. on Recent Advances in Engineering and Computational Sciences*, pp. 1-4, 2015.

[13] Romero, D.E.T. "High-speed multiplierless Frequency Response Masking (FRM) FIR filters with reduced usage of hardware resources," *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1-4, 2015.

[14] Lu, W.-S. and Takao H., "A unified approach to the design of interpolated and frequency response masking FIR filters," *IEEE Transactions on Circuits and Systems I – Reg. Papers*, 2016. (in press)

[15] Tai, Y. L., Liu, J. C. and Chou, H. H. "Design of FIR Hilbert transformers using prescribed subfilters and nested FT technique," *International Journal of Electronics*, pp.1–14, 2014.

[16] Demirtas, S. and Oppenheim A. V., "A functional composition approach to filter sharpening and modular filter design," *IEEE Transactions on Signal Processing*, 2016. (in press)

[17] Kaiser, F., and Hamming R. “Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter,” *IEEE Trans. Acoust., Speech, Signal Process*, *ASSP-25*, pp. 415–422, 1977.

[18] Kwentus, A., Jiang, Z., and Willson, A. N. “Application of filter sharpening to cascaded integrator-comb decimation filters,” *IEEE Trans. Signal Process*, 45, pp. 457–467, 1997.

[19] Dolecek G. J., and Mitra S. K., “A new two-stage sharpened comb decimator,” *IEEE Trans. Circuits and Systems I – Reg. Papers*, vol. 54, no. 4, pp. 994-1005, 2005.

[20] M. Laddomada, “comb-based decimation filters for sigma-delta AD converters: novel schemes and comparisons,” *IEEE Trans. Signal Processing*, vol. 55, no. 5, pp. 1769–1779, 2007.

[21] Dolecek G. J. and Harris F., “Design of wideband CIC compensator filter for a digital IF receiver,” *IEEE Trans. Signal Processing*, vol. 19, no. 5, pp. 827–837, 2009.

[22] Salgado, G. M., Dolecek, G. J., and de la Rosa, J. M. “Novel two-stage comb decimator with improved frequency characteristic,” *Circuits & Systems (LASCAS) 2015 IEEE 6th Latin American Symposium on*, pp. 1–4, 2015.

[23] Hartnett, R., and Boudreaux, G. “Improved filter sharpening,” *IEEE Trans. on Signal Process*, vol. *43*, pp. 2805–2810, 1995.

[24] Samadi, S. “Explicit formula for improved filter sharpening polynomial,” *IEEE Trans. on Signal Process*, vol. *9*, pp. 2957–2959, 2000.

[25] Coleman, J. O. “Chebyshev stopband for CIC decimation filters and CIC-implemented array tapers in 1D and 2D,” *IEEE Trans. on Circuits and Systems I: Regular papers*, vol. *59*, no. *12*, pp. 2956–2968, 2012.

[26] Romero, D. E. T., Dolecek, G. J. and Laddomada, M. “Efficient design of two-stage comb-based decimation filters using Chebyshev sharpening,” *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Columbus, OH, pp. 1011–1014, 2013.

[27] Coleman, J. O. “Integer-coefficient FIR filter sharpening for equiripple stopbands and maximally flat passbands,” *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, Melbourne VIC, pp. 1604–1607, 2014.

[28] Saramaki, T. and Ritoniemi, T. “A modified comb filter structure for decimation,” in Proc. IEEE Int. Symp. on Circuits and Systems, vol. 4, pp. 2353–2356, 1997.

[29] Candan, C. “Optimal Sharpening of CIC filters and an efficient implementation through Saramaki-Ritoniemi decimation filter structure,” 2011. http://www.eee.metu.edu.tr/~ccandan/pub dir/opt sharpened CIC filt extended new.pdf. (last access on February 2017)

[30] Molnar G., Pecotic M. G. and Vucic M. “Weighted least-squares design of sharpened CIC filters,” *IEEE Internat. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2013.

[31] Molnar G. and Vucic M. “Weighted minimax design of sharpened CIC filters,” *IEEE Internat. Conference on Electronics, Circuits and Systems (ICECS)*, Dec. 2013.

[32] Laddomada M., Romero D. E. T. and Dolecek G. J., “Improved sharpening of comb-based decimation filters: analysis and design,” *IEEE Consumer Communications and Networking Conference (CCNC)*, Nov. 2014.

[33] Romero D. E. T., Laddomada M. and Dolecek G. J., “Optimal sharpening of compensated comb decimation filters: analysis and design,” *The Scientific World Journal*, Jan. 2014.

[34] Molnar G., Dudarin A. and Vucic M. “Minimax design of multiplierless sharpened CIC filters based on interval analysis,” *IEEE Internat. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2016.

[35] Kastner, R., Hosangadi, A., and Fallah, F. *Arithmetic optimization techniques for hardware and software design*, Cambridge University Press, 2010.

[36] Aksoy, L., Flores, P. and Monteiro, J. “A tutorial on multiplierless design of FIR filters: Algorithms and architectures,” *Circuits, Systems and Signal Processing*, vol. 33, pp. 1689–1719, 2014.

[37] Qureshi, F. and Gustafsson, O. “Low-complexity reconfigurable complex constant multiplication for FFTs,” *in Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 24–27, 2009.

[38] Thong, J. and Nicolici, N. “An optimal and practical approach to single constant multiplication,” *IEEE Trans. Comput. Aided Des.*, vol. 30, no. 9, pp. 1373–1386, 2011.

[39] Pan, Y. and Meher, P. K. “Bit-level optimization of adder trees for multiple constant multiplications for efficient FIR filter implementation,” *IEEE Trans. Circ. Syst. I*, vol. 61, no. 2, pp. 455-462, 2014.

[40] Guo, R., DeBrunner, L. S., and Johansson, K. “Truncated MCM using pattern modification for FIR filter implementation,” *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, pp. 3881-3884, 2010.

[41] Aksoy, L., Costa, E., Flores, P. and Monteiro, J. “Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits*, vol. 27, no. 6, pp. 1013-1026, 2008.

[42] Aksoy, L., Costa, E., Flores, P. and Monteiro, J. “Finding the optimal tradeoff between area and delay in multiple constant multiplications,” *Elsevier J. Microprocess. Microsyst.*, vol. 35, no. 8, pp. 729-741, 2011.

[43] Faust, M. and Chip-Hong, C. “Minimal logic depth adder tree optimization for multiple constant multiplication,” *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 457-460, 2010.

[44] Johansson, K., Gustafsson, O., DeBrunner, L. S. and Wanhammar, L. “Minimum adder depth multiple constant multiplication algorithm for low power FIR filters,” *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, Rio de Janeiro, pp. 1439-1442, 2011.

[45] Aksoy, L., Costa, E., Flores, P. and Monteiro, J. *Multiplierless design of linear DSP transform*s, in VLSI-SoC: Advanced Research for Systems on Chip, Springer, Chap. 5, pp. 73–93, 2012.

[46] Ho, Y. H., Lei, C. U., Kwan, H. K., and Wong, N. "Global optimization of common subexpressions for multiplierless synthesis of multiple constant multiplications," *in Proceedings of Asia and South Pacific Design Automation Conference*, pp. 119–124, 2008.

[47] Hosangadi, A., Fallah, F., and Kastner, R. "Simultaneous optimization of delay and number of operations in multiplierless implementation of linear systems," *in Proceedings of International Workshop on Logic Synthesis*, 2005.

[48] Mirzaei, S., Kastner, R., and Hosangadi, A. "Layout Aware Optimization of High Speed Fixed Coefficient FIR Filters for FPGAs," *Int. Journal of Reconfigurable Computing*, pp. 1–17 ,2010.

[49] Meyer-Baese, U., Botella, G., Romero, D. E. T. and Kumm, M. "Optimization of high speed pipelining in FPGA-based FIR filter design using Genetic Algorithm," *Proc. SPIE 8401, Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering X,* 2012.

[50] Kumm, M., Zipf, P., Faust, M. and Chang, C. H. "Pipelined adder graph optimization for high speed multiple constant multiplication," *IEEE Int. Symp. on Circuits and Systems*, pp. 49–52, 2012.

[51] Kumm, M., Fanghanel, D., Moller, K., Zipf, P., and Meyer-Baese, U. "FIR filter optimization for video processing on FPGAs," *EURASIP Journal on Advances in Signal Processing*, DOI: 10.1186/1687-6180-2013-111, 2013.

[52] Kumm, M., Hardieck, M., Willkomm, J., Zipf, P., and Meyer-Baese, U., “Multiple constant multiplications with ternary adders,” *International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–8, 2013.

[53] Gustasson, O. “Lower bounds for constant multiplication problems,” *IEEE Trans. Circuits and Syst. II: Express briefs*, vol. 54, no. 11, pp. 974–978, 2007.

[54] Romero D. E. T., Meyer-Baese U. and G. J. Dolecek, "On the inclusion of prime factors to calculate the theoretical lower bounds in multiplierless single constant multiplications," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 122, pp. 1-9, 2014.

[55] Meyer-Baese, U. *Digital Signal Processing with Field Programmable Gate Arrays*, Springer, 2014.

[56] Voronenko, Y., and Püschel, M. “Multiplierless multiple constant multiplication,” *ACM Trans. Algorithms*, vol. 3, no. 2, 2007.

[57] Gustafsson, O., Dempster, A. G., Johansson, K., Macleod, M. D., and Wanhammar, L. “Simplified design of constant coefficient multipliers,” *Circ. Syst. Signal Process*, vol. 25, no.2, pp. 225–251, 2006.

Chapter 3

Methods and architectures that employ comb and cosine filters as basic building blocks

The central idea of the research here developed is a method to design FIR filters with minimum possible number of arithmetic operations for a desired magnitude characteristic. Usually, the main aspects taken into account in filters for communications are a passband close to the ideal and an acceptable attenuation. For that reason, the contributions developed in this thesis are based on these crucial points. Considering that the use of simple filters in the low complexity FIR filter design results effective, it is hypothesized here that a filter with comb and cosine filters as basic building blocks will benefit from their magnitude characteristics by adding low complexity. Although these filters are practical, they have passpband droop and poor attenuation. Using compensator filters in cascade helps to improve the passband characteristic. Complementary to this, the Sharpening techniques can enhance the magnitude characteristics of cosine and comb filters by the tapped cascaded interconnection of these simple filters. With regard to the computational complexity, by using multirate approaches it is possible to reduce the number of arithmetic operations to be implemented, particularly in sampling rate conversion cases.

This chapter is organized as follows. First, the use of Chebyshev sharpening to design cosine-based prefilters is presented in Section 3.1. The proof that the Chebyshev sharpening technique provides filters with Minimum Phase (MP) characteristic when it is applied to cosine filters is given. Additionally, a mathematical demonstration that cascaded expanded Chebyshev-Sharpened Cosine Filters (CSCFs) are also MP filters is established. Then, from Sections 3.2 to 3.6, the subfilter-based approaches are particularly developed for comb-based decimators. Sections 3.2 to 3.4 follow the scheme of increasing the attenuation of comb filters and correcting their passband droop in separate ways, whereas Sections 3.5 and 3.6 follow the scheme of improving these magnitude characteristics in a unified way via sharpening. In Section 3.2, a method to design low-complexity wide-band compensators to improve the passband characteristic of comb and comb-based filters sharpened with Chebyshev polynomials is developed. Subsequently, in Section 3.3, a method to design comb-based decimation filters with improved magnitude response characteristics, based on compensation filters and Chebyshev polynomials is derived. In Section 3.4, a comb-based decimator that consists of an area-efficient structure aided with an embedded simplified Chebyshev-sharpened section is proposed. A method to design comb-based decimation filters with improved magnitude response characteristics, which consists in applying the Hartnett-Boudreaux sharpening technique (so-called improved sharpening) is explained in section 3.5. Finally in 3.6, Comb-based decimation architectures split in stages, based on the Harnett-Boudreaux sharpening, are detailed. The developed proposals are explained and illustrated with examples.

3.1 Minimum phase property of Chebyshev-sharpened Cosine filters

A Minimum Phase (MP) digital filter has all zeros on or inside the unit circle [1]. The basic building block analyzed here, the *cosine filter*, is a simple FIR filter whose transfer function and frequency response are, respectively, given by

$$H_{\cos}(z) = \frac{1}{2} \cdot (1 + z^{-L}), \tag{3.1}$$

$$\left| H_{\cos}(e^{j\omega}) \right| = \left| \cos(\omega L / 2) \right|. \tag{3.2}$$

This filter is of special interest because of the following main reasons:

(a) It has MP property because its zero lies on the unit circle.

(b) It has a low computational complexity because it does not require multipliers, which are the most costly and power-consuming elements in a digital filter [2].

(c) It has a low usage of hardware elements, which can be translated into a low demand of chip area for implementation.

When applied to comb filters, the Chebyshev sharpening approach provides solutions with advantages like a simple and elegant design method, a low-complexity resulting LP FIR filter and improved attenuation characteristics in the resulting filter [3]. However, filters from [3] are not guaranteed to have MP characteristic. In that method the sharpening is performed with a N-th degree Chebyshev polynomial of first kind, defined as

$$P(x) = \sum\nolimits_{n=0}^{N} c_n \cdot x^n \ . \tag{3.3}$$

Demonstrating the MP characteristic of Chebyshev-Sharpened Cosine Filters (CSCFs) is motivated by the following facts: 1) Cosine-based prefilters may result in high delay, which is not tolerated in many applications —particularly, in MP FIR filters the reduction of the group delay is a priority—; 2) The use of cosine filters results in low-complexity multiplierless FIR filters; 3) The recent Chebyshev sharpening method from [3] can improve the attenuation of cosine filters and is a potentially useful approach to preserve a simple multiplierless solution with a lower group delay in comparison with simple cascaded expanded cosine filters. Thus, the demonstration of MP characteristics in CSCF-based prefilters is developed in the following. Subsection 3.1.1 presents the definition of CSCFs and cascaded expanded CSCFs. The proofs of MP characteristic in CSCFs and cascaded expanded CSCFs are given in subsections 3.1.2 and 3.1.3, respectively. In 3.1.4 details on the characteristics and applications of the cascaded expanded CSCFs are provided, and a design example is included.

3.1.1 Definition of Chebyshev-sharpened cosine filter (CSCF) and cascaded expanded CSCF

We define the transfer function and the frequency response of an N-th order Chebyshev-Sharpened Cosine Filter (CSCF) respectively as,

$$H_{C,N}(z,\gamma) = \sum\nolimits_{n=0}^{N} z^{-(N-n)/2} \cdot c_n \cdot [\gamma H(z)]^n \ , \tag{3.4}$$

$$H_{C,N}(e^{j\omega},\gamma) = \left[\sum\nolimits_{n=0}^{N} c_n \cdot [\gamma \cos(\omega/2)]^n\right] e^{-j\omega N/2} , \tag{3.5}$$

with

$$\gamma \leq \frac{1}{\cos(\frac{\pi}{2} - \frac{\pi}{4R})}, \tag{3.6}$$

where c_n are the coefficients of the Chebyshev polynomial of first kind, represented in (3.3), and $H(z)$ is given in (3.1). To obtain a low-complexity multiplierless implementation, the constant γ must be expressible as a Sum of Powers of Two (SOPOT). To this end, we set

$$\gamma = f\left(2^{-B}\left\lfloor \frac{2^{B}}{\cos(\frac{\pi}{2} - \frac{\pi}{4R})} \right\rfloor,\ 1\right), \tag{3.7}$$

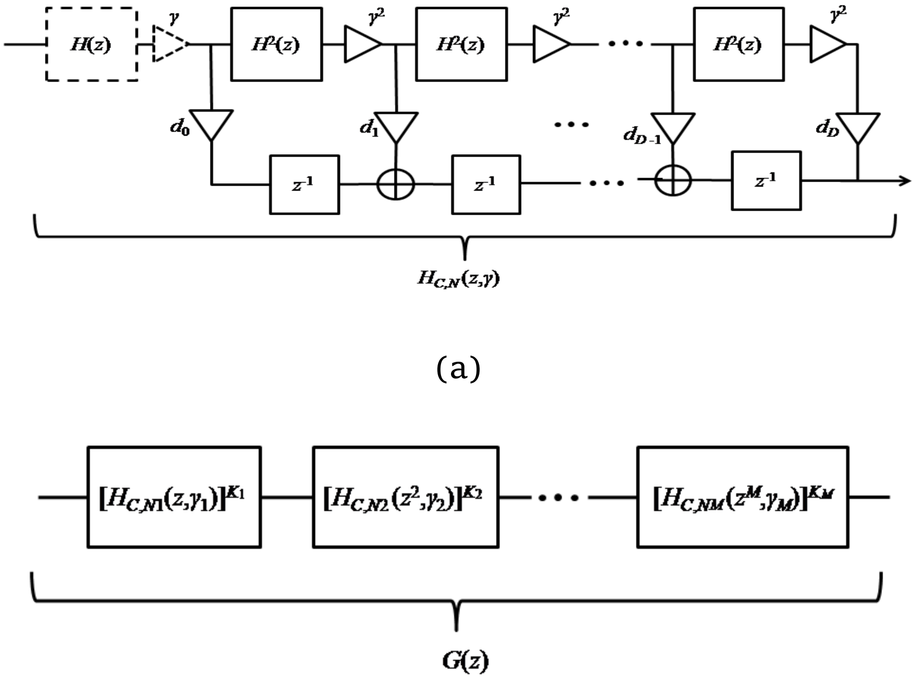
where $f(a, b)$ denotes "the closest value less than or equal to a that can be realized with at most b adders" and $\lfloor x \rfloor$ denotes rounding x to the closest integer less than or equal to x. To provide an improved attenuation around the zero of the cosine filter, γ must be as close as possible to its upper limit [2]. This is achieved by increasing the integer B. The value R in (3.6)-(3.7) is usually set as an integer equal to or greater than 2 for applications in decimation processes [3].

The transfer function and frequency response of a cascaded expanded CSCF are respectively defined as

$$G(z) = \prod_{m=1}^{M} [H_{C,N_m}(z^m, \gamma_m)]^{K_m} \ , \tag{3.8}$$

$$G(e^{j\omega}) = \left\{ \prod\nolimits_{m=1}^{M} \left[\sum\nolimits_{n=0}^{N_m} c_n \cdot [\gamma_m \cos(m \cdot \omega / 2)]^n \right]^{K_m} \right\} e^{-j\omega \sum_{m=1}^{M} m \cdot K_m \cdot N_m / 2}, \tag{3.9}$$

where the integer M indicates the number of cascaded CSCF blocks, each of them repeated K_m times, with m = 1, 2, ..., M. Every value of m is a distinct factor that expands a different CSCF whose corresponding order is N_m. These CSCFs have different factors γ_m, which can be obtained using (3.7), just replacing B by B_m and R by R_m, where B_m and R_m are integer parameters that correspond to the m-th CSCF in the cascade. Figure 3.1(a) shows the structure of the CSCF, where we have that $d_i = c_{2i+v}$, with i = 0, 1, 2, ..., $D = (N - v)/2$ and with $v = 1$ if N is odd or $v = 0$ if N is even. Dashed blocks in Figure 3.1(a) appear only if N is odd. Figure 3.1(b) presents the structure of the cascaded expanded CSCF whose transfer function is given in (3.8).



(a)

(b)

Figure 3.1. General structure of the filters: (a) Chebyshev-Sharpened Cosine Filter (CSCF); (b) Cascaded expanded CSCF.

3.1.2 Proof of minimum phase property in CSCFs

The proof starts with the expression of the Chebyshev polynomial from (3.3) in the form of a product of first-order terms as [4]

$$P(x) = \sum_{n=0}^{N} c_n \cdot x^n = \prod_{n=1}^{N} (x - \sigma_n), \qquad (3.10)$$

$$\sigma_n = \cos\left(\tfrac{\pi}{2} \cdot \tfrac{2n-1}{N}\right). \tag{3.11}$$

On the other hand, we re-write the transfer function of the CSCF from (3.4) as

$$H_{C,N}(z,\gamma) = z^{-N/2}\sum\nolimits_{n=0}^{N} c_n \cdot [z^{1/2}\gamma H(z)]^n\ . \tag{3.12}$$

Using (3.10), and after simple re-arrangement of terms, we express $H_{C,N}(z,\ \gamma)$ as follows,

$$H_{C,N}(z,\gamma) = \prod\nolimits_{n=1}^{N}[\gamma H(z) - z^{-1/2}\sigma_n]\ , \tag{3.13}$$

which can be rewritten

$$\text{as}\, H_{C,N}(z,\gamma) = \begin{cases} \displaystyle\prod_{n=1}^{N/2}\{[\gamma H(z) - z^{-1/2}\sigma_n]\times \\ \qquad [\gamma H(z) - z^{-1/2}\sigma_{N-(n-1)}]\};\ N \text{ even}, \\ [\gamma H(z) - z^{-1/2}\sigma_{\lceil N/2\rceil}] \displaystyle\prod_{n=1}^{\lceil N/2\rceil-1}\{[\gamma H(z) - z^{-1/2}\sigma_n] \\ \qquad \times[\gamma H(z) - z^{-1/2}\sigma_{N-(n-1)}]\};\ N \text{ odd}, \end{cases} \tag{3.14}$$

where $\lceil x \rceil$ denotes rounding x to the closest integer greater than or equal to x.

At this point, it is worth highlighting that the anti-symmetry relations

$$\sigma_n = -\sigma_{N-(n-1)}\,, \qquad n = 1,\, 2,\, \ldots,\ \lceil N/2 \rceil, \tag{3.15}$$

$$\sigma_{\lceil N/2\rceil} = 0 \quad \text{for } N \text{ odd}, \tag{3.16}$$

hold [4]. Thus, replacing (3.15) and (3.16) in (3.14), and after simple manipulation of terms, we have

$$H_{C,N}(z,\gamma)=\begin{cases}\prod_{n=1}^{N/2}Q_n(z);\ N\text{ even},\\ \gamma H(z)\prod_{n=1}^{\lceil N/2\rceil-1}Q_n(z);\ N\text{ odd},\end{cases}\tag{3.17}$$

$$Q_n(z)=\gamma^2H^2(z)-\sigma_n^2z^{-1}\,.\tag{3.18}$$

From (3.17) we have that $H_{C,N}(z,\ \gamma)$ consists of a product of either several terms $Q_n(z)$ if N is even or several terms $Q_n(z)$ and a term $\gamma H(z)$ if N is odd, with n = 1, 2, ..., N. Thus, to prove the MP property of the CSCF it is only necessary to ensure that $Q_n(z)$ and $\gamma H(z)$ have MP characteristic for all values n.

Using (3.1), it is easy to see that the term $\gamma H(z)$ has a root on the unit circle and thus it corresponds to a MP filter. On the other hand, after simple re-arrangement of terms we get

$$Q_n(z)=\tfrac{\gamma^2}{4}[1-(\tfrac{4\sigma_n^2}{\gamma^2}-2)z^{-1}+z^{-2}].\tag{3.19}$$

From (3.19) it is easy to show that the roots of $Q_n(z)$ are placed on the unit circle, i.e.,

$$Q_n(z)=(1-e^{j2\phi_n}z^{-1})(1-e^{-j2\phi_n}z^{-1}),\tag{3.20}$$

$$\phi_n=\arccos(\sigma_n\cdot\gamma^{-1}),\tag{3.21}$$

if the argument $\sigma_n\cdot\gamma^{-1}$ in (3.21) is preserved into the range [−1, 1]. From (3.11) we have that $-1\leq\sigma_n\leq1$ holds. Additionally, by setting

$$R\geq0.5\tag{3.22}$$

in (3.6)-(3.7), we ensure $\gamma\geq1$. Under this condition for R, we have that $-1\leq\gamma^{-1}\leq1$ holds. In this case, $Q_n(z)$ has its roots on the unit circle for all the valid values n and, as a consequence, the filter $H_{C,N}(z,\ \gamma)$ has a MP characteristic. ■

Figure 3.2 shows the pole-zero plots for the filters $H_{C,2}(z, \gamma)$, $H_{C,3}(z, \gamma)$, $H_{C,4}(z, \gamma)$ and $H_{C,5}(z, \gamma)$. For all these filters, we have $\gamma = 2^{-3}\times 15$, which is implemented with just one subtraction.

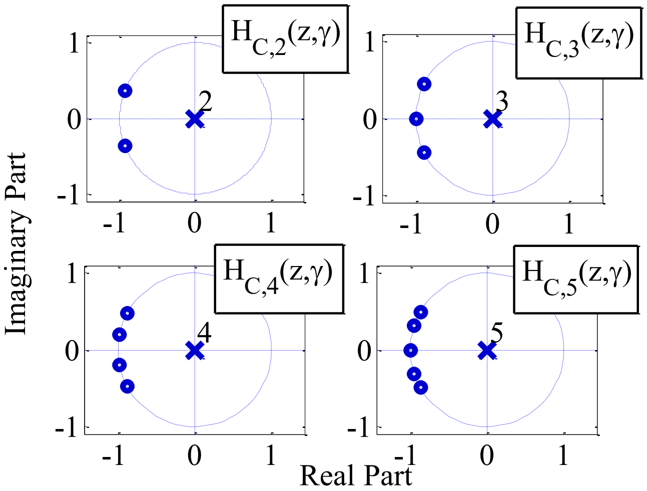


Figure 3.2. Pole-zero plots for CSCFs $H_{C,2}(z, \gamma)$, $H_{C,3}(z, \gamma)$, $H_{C,4}(z, \gamma)$ and $H_{C,5}(z, \gamma)$, where $\gamma=2^{-3}\times 15$.

3.1.3 Proof of minimum phase property in cascaded expanded CSCFs

The proof starts with the expression of every CSCF of the cascaded expanded CSCF from (3.8) in the form of a product of second-order expanded transfer functions using (3.17) and (3.19), i.e.,

$$H_{C,N_m}(z^m,\gamma_m)=\begin{cases}\prod_{n=1}^{N_m/2}Q_n(z^m);\ N_m \text{ even},\\ \gamma_m H(z^m)\prod_{n=1}^{\lceil N_m/2\rceil-1}Q_n(z^m);\ N_m \text{ odd},\end{cases} \qquad (3.23)$$

$$Q_n(z^m)=\tfrac{\gamma_m{}^2}{4}[1-(\tfrac{4\sigma_n^2}{\gamma_m^2}-2)z^{-m}+z^{-2m}]. \qquad (3.24)$$

where m = 1, 2, ..., M and n = 1, 2, ..., N_m. Since the transfer function of the cascaded expanded CSCF from (3.8) consists of a product of several terms $[H_{C,Nm}(z^m, \gamma_m)]^{Km}$ with different values m, it is only necessary to ensure that $H_{C,Nm}(z^m, \gamma_m)$ has a MP characteristic for all values m. Moreover, from (3.23) we see that $H_{C,Nm}(z^m, \gamma_m)$ is expressed as a product of either several terms $Q_n(z^m)$ if N_m is even or several terms $Q_n(z^m)$ and $\gamma_m H(z^m)$ if N_m is odd. Thus, to prove the MP property in cascaded expanded CSCFs we only need to ensure that $Q_n(z^m)$ and $\gamma_m H(z^m)$ have MP characteristic for all values n and m .

By replacing (3.1) in the term $\gamma_m H(z^m)$ and then making the resulting expression equal to zero, we can find the m roots of $\gamma_m H(z^m)$. These roots turn out to be the m complex roots of –1, which have unitary magnitude. Thus, $\gamma_m H(z^m)$ has MP characteristic, since its roots are placed on the unit circle. On the other hand, using (3.20) we can express (3.24) as follows,

$$Q_n(z^m) = (1 - e^{j2\phi_n} z^{-m})(1 - e^{-j2\phi_n} z^{-m}), \tag{3.25}$$

$$\phi_n = \arccos(\sigma_n \cdot \gamma_m^{-1}). \tag{3.26}$$

To preserve the argument $\sigma_n \cdot \gamma_m^{-1}$ in (3.26) into the range [–1, 1], we set

$$R_m \geq 0.5, \quad m = 1, 2, \ldots, M. \tag{3.27}$$

Under this condition for R_m, we have that $-1 \leq \gamma_m^{-1} \leq 1$ holds. In this case, the respective m roots of factors $(1 - e^{j2\phi n} z^{-m})$ and $(1 - e^{-j2\phi n} z^{-m})$ in (3.24) are the m roots of the complex numbers $e^{j2\phi n}$ and $e^{-j2\phi n}$, which have unitary magnitude for all the valid values n. Therefore, $Q_n(z^m)$ has MP characteristic, since its roots are placed on the unit circle. Finally, since $Q_n(z^m)$ and $\gamma_m H(z^m)$ have MP characteristic, the overall cascaded expanded CSCF from (3.7), $G(z)$, also has MP characteristic. ■

Figure 3.3 shows the pole-zero plots for the filters $H_{C,2}(z^5, \gamma)$, $H_{C,3}(z^4, \gamma)$, $H_{C,4}(z^3, \gamma)$ and $H_{C,5}(z^2, \gamma)$. For all these filters, we have $\gamma = 2^{-3}\times 15$, which is implemented with just one subtraction.

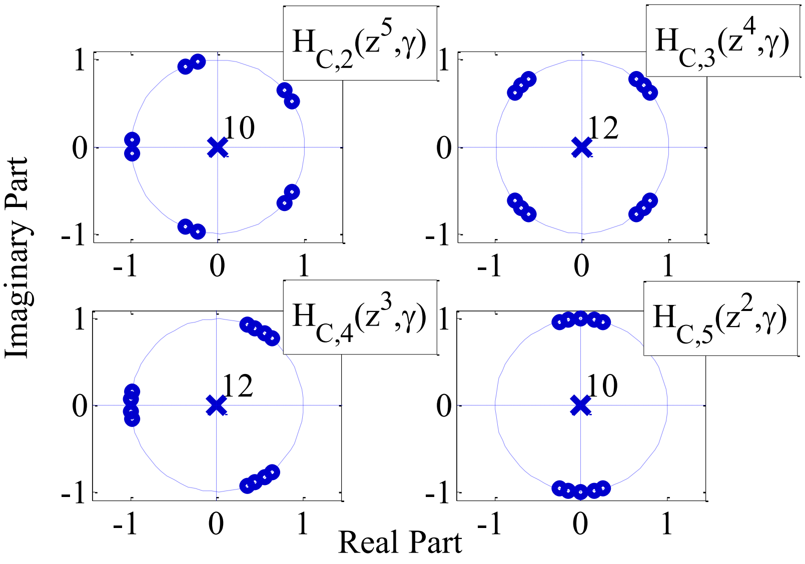


Figure 3.3. Pole-zero plots for cascaded expanded CSCFs $H_{C,2}(z^5, \gamma)$, $H_{C,3}(z^4, \gamma)$, $H_{C,4}(z^3, \gamma)$ and $H_{C,5}(z^2, \gamma)$, where $\gamma=2^{-3}\times 15$.

3.1.4 Characteristics and applications of cascaded expanded CSCFs

A cascaded expanded CSCF has both, MP and LP characteristics. The former was proven in subsection 3.1.3, whereas the latter is easily seen from the frequency response $G(e^{j\omega})$ given in (3.9). A consequence of this is that the cascaded expanded CSCF has a passband droop in its magnitude response. Due to this passband droop, the cascaded expanded CSCF should be employed only to provide a given attenuation requirement of an overall LP or MP FIR filter over a prescribed stopband region (depending on the application). The cascaded expanded CSCF, with transfer function $G(z)$ defined in (3.8), can be used as prefilter. Note that, since a cascaded expanded cosine filter also has both, LP and MP properties, it is used as prefilter in [5].

Since a FIR equalizer with LP characteristic has its zeros placed in quadruplets around the unit circle, it does not accomplish the MP characteristic. Therefore, a MP FIR equalizer (i.e., that filter whose zeros appear inside the unit circle) does not have a linear phase.

In method [5] the delay D has been removed to obtain an MP FIR equalizer. Thus, a first option would be to use the same approach of [5] to design a FIR equalizer. Besides of method [5], other design methods for MP FIR filters have been introduced for example in [6]-[8]. However, in general, these methods have the inconvenience of producing filtering solutions that require multipliers, which are the most costly elements in a digital filter [1]. To solve this problem, the cascaded expanded CSCF can be used as a prefilter to implement an overall MP FIR filter using several multiplierless CSCFs.

Example 1

The comparison is made in terms of:

a) Group delay, measured in samples and defined as follows

$$\tau(\omega) = -\frac{d}{d\omega}\{\arg[F(e^{j\omega})]\}, \qquad (3.28)$$

where $F(e^{j\omega})$ is the frequency response of the corresponding filter.

b) Implementation complexity, measured in the required number of adders and delays for a given attenuation over a prescribed stopband region.

Design a MP FIR filter with minimum attenuation equal to 60 dB *over the range from $\omega = 0.17\pi$ to $\omega = \pi$ (see Fig. 1 of [5]).*

In [5], the filter employed to accomplish such characteristic is obtained using $K = 5$ and $L = 3$. The group delay is obtained by replacing these values in the transfer function of the cascaded expanded CSCF in (3.28). This filter requires 15 adders and 45 delays, but it has a group delay of 22.5 samples.

If we use $M = 4$, $N_1 = N_3 = N_4 = 3$, $N_2 = 4$, $R_1 = 3$, $R_2 = 1.5$, $R_3 = 0.9$, $R_4 = 2$, with $B_m = 4$ and $K_m = 1$ for all m in (3.8), we get a filter whose group delay, obtained by replacing the aforementioned parameters in (3.9) and then using (3.9) in (3.28), is 16 samples, i.e., nearly 30% less delay than that of [5]. Since this filter uses 30 adders and 44 delays, the price to pay is $100\times\{[(30+44)/(15+45)]-1\}\approx 23\%$ of additional implementation complexity. Figure 3.4 shows the magnitude responses and group delays of both filters. Moreover, Table 3.1 and Table 3.2 present, respectively, the first half of the symmetric impulse response of the filter designed with method [5] and the proposed filter. Table 3.3 summarizes the results from the previous examples. From them we observe that the cascaded expanded CSCFs achieve a lower group delay in comparison to the cascaded expanded cosine filters from [5].

Table 3.1. First half of the symmetric impulse response of the filter designed with method [5] in Example 1.

n	$\boldsymbol{h_A(n)}$	***n***	$\boldsymbol{h_A(n)}$	***n***	$\boldsymbol{h_A(n)}$
1	0.000030517578125	**9**	0.004943847656250	**17**	0.037902832031250
2	0.000091552734375	**10**	0.007110595703125	**18**	0.043304443359375
3	0.000183105468750	**11**	0.009887695312500	**19**	0.048431396484375
4	0.000396728515625	**12**	0.013275146484375	**20**	0.052825927734375

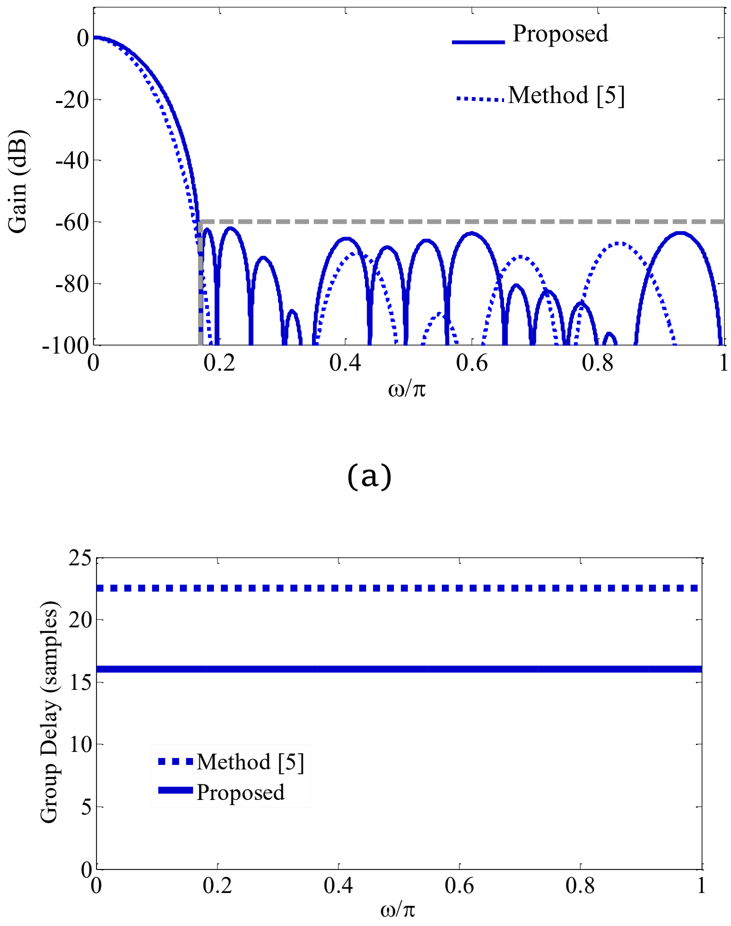
5	0.000732421875000	**13**	0.017272949218750	**21**	0.056488037109375
6	0.001281738281250	**14**	0.021881103515625	**22**	0.058959960937500
7	0.002136230468750	**15**	0.026916503906250	**23**	0.060241699218750
8	0.003295898437500	**16**	0.032409667968750		

Table 3.2. First half of the symmetric impulse response of the proposed filter in Example 1.

n	***g(n)***	***n***	***g(n)***	***n***	***g(n)***
1	0.000365884150812	**7**	0.014876445801089	**13**	0.055883940227800
2	0.001024551768820	**8**	020473561194634	**14**	0.061849547506437
3	0.002122204221257	**9**	0.027011012207314	**15**	0.066392852921837
4	003834694340150	**10**	0.034112182959393	**16**	069389704077153
5	0.006627799290290	**11**	0.041585018019638	**17**	0.070269686319927
6	0.010243501009105	**12**	049072257144308		

Table 3.3. Comparison of results in Example 1.

	Example 1	
	Proposed	Method [5]
Group delay (samples)	16	22.5
Complexity of Implementation (No. adders/ No. delays)	30 / 44	15 / 45
% improvement in group delay (compared with method [5])	≈30%	–
% increase in complexity of implementation (compared with method [5])	≈23%	–



(a)

(b)

Figure 3.4. (a) Magnitude responses and (b) group delays of the cascaded expanded CSCF (eq. (3.8)) and the cascaded expanded cosine filter from [5], accomplishing the attenuation required in Example 1.

3.2 Low-complexity compensators based on Chebyshev polynomials

The design of compensator filters is an important branch of research in digital filters design area. To improve the passband region of any digital filter a compensator filter is helpful. Usually, the compensators are simple filters with low order and low arithmetic complexity. By using a compensator filter in cascade of specific filter the magnitude response is enhanced. The aim of this proposals is introducing a formulation to easily design compensation filters specifically for improving the passband characteristic of decimators. In subsection 3.2.1 the use of amplitude transformation technique applied to comb compensators design is detailed. Then in 3.2.2 the design of low-complexity second-order compensators to improve the passband characteristic of Chebyshev Comb Filters is introduced. This formulation is based on the amplitude transformation method recently presented in [9] to design traditional comb compensators. A simple formula to obtain the coefficients of Chebyshev Comb Filters compensators is provided, which makes straightforward the design of these filters. Next in subsection 3.2.3, the design of a wide-band compensation filters for improving the passband behavior of Cascade Integrator Comb decimators is presented. The framework hinges on the amplitude transformation method [9].

3.2.1 Design of Comb compensators using Amplitude Transformation

The approach of designing comb compensators by modifying the amplitude response of a cosine-squared filter with transfer function $F(z)$ and frequency response $F(\mathrm{e}^{j\omega}) = F(\omega)e^{-j\omega}$, where

$$F(z) = 2^{-2}(1 + 2z^{-1} + z^{-2})\,, \tag{3.29}$$

$$F(\omega) = \cos^2(\omega / 2)\,, \tag{3.30}$$

was introduced in [9]. The resulting compensator has the transfer function

$$C(z) = \sum\nolimits_{i=0}^{N} z^{-(N-i)} p_i F^i(z)\,, \tag{3.31}$$

where p_i is the coefficient of the i-th power (with $0 \leq i \leq N$) of a N-th degree polynomial used to transform the amplitude response of the cosine-squared filter into an amplitude characteristic proper for compensation (such polynomial is referred hereafter as transformation polynomial). The frequency response of the compensator is $C(e^{j\omega})$ = $C(\omega, \mathbf{p}) \times e^{-j\omega N}$, where

$$C(\omega, \mathbf{p}) = \sum_{i=0}^{N} p_i F^i(\omega) = \mathbf{p} \cdot [1 \quad F(\omega) \quad \ldots \quad F^N(\omega)]^T, \tag{3.32}$$

with $\mathbf{p} = [p_0 \; p_1 \ldots p_N]$.

For an arbitrarily chosen N, the vector of optimal polynomial coefficients, $\mathbf{p}^*$, is found by minimizing the passband error solving the following optimization problem under the L_p-norm,

$$\mathbf{p}^* = \arg \min_{0 \leq \omega \leq \pi/R} \left\{ \left\| 1 - C(\omega, \mathbf{p}) \cdot \tfrac{1}{M^K} \cdot H^K(\omega M^{-1}) \right\|_{L_p} \right\}, \tag{3.33}$$

where the scaling $1/M^K$ is introduced to achieve a gain of 0 dB in zero frequency.

3.2.2 Design of low-complexity second-order compensators to improve the passband characteristic of Chebyshev Comb Filters

To design a compensation filter for a K-th order Chebyshev Comb Filters (CCFs), the optimization problem is no longer that introduced in (3.33). The passband error must consider in this case the amplitude characteristic of the K-th order CCF, resulting in the following optimization problem,

$$\mathbf{p}^* = \arg \min_{0 \leq \omega \leq \pi/R} \left\{ \left\| 1 - C(\omega, \mathbf{p}) \cdot S \cdot H_{C,K}(\omega M^{-1}) \right\|_{L_p} \right\}. \tag{3.34}$$

In (3.34), S is a scaling constant that allows having unitary gain at zero frequency, given by

$$S = [1/H_{C,K}(\omega)]\Big|_{\omega=0} = 1\Big/\Big[\sum_{k=0}^{K} c_k \cdot (\gamma M)^k\Big]. \qquad (3.35)$$

Since the cosine-squared filter is a second-order filter, it must undergo a linear transformation in order to obtain a second-order CCF compensator, i.e., the order of the transformation polynomial must be N = 1. Using this value of N and replacing (3.30) in (3.32) we obtain

$$C(\omega,\mathbf{p}) = p_0 + p_1\cos^2(\omega/2) = \mathbf{p}\cdot[1 \quad \cos^2(\omega/2)]^T, \qquad (3.36)$$

with $\mathbf{p}$ = [p_0 p_1]. Substituting (3.28), (3.35) and (3.36) in (3.34), the optimization problem becomes

$$\mathbf{p}^* = \arg\min_{0\le\omega\le\pi/R}\left\{\left\|1-\left(p_0+p_1\cos^2(\omega/2)\right)\cdot\left(1\Big/\Big[\sum_{k=0}^{K} c_k\cdot M^k\Big]\right)\times\sum_{k=0}^{K} c_k\cdot H^k(\omega M^{-1})\right\|_{L_p}\right\}. \qquad (3.37)$$

For ω = 0, the passband error is ε = 1 – p_0 – p_1. By arbitrarily setting ε = 0, we can express p_0 in terms of p_1 as follows, p_0 = 1 – p_1. In this way, p_1 becomes the unique unknown coefficient of the transformation polynomial. Replacing p_0 = 1 – p_1 in (3.37), the maximum error in the passband can be minimized by solving the following problem,

$$\begin{aligned} p_1^* = \arg\min_{0\le\omega\le\pi/R}\Bigg\{\Bigg\|&1-\left(1-p_1+p_1\cos^2(\omega/2)\right)\left(1\Big/\Big[\sum_{k=0}^{K} c_k\cdot M^k\Big]\right) \\ &\times\left(1\Big/\Big[\sum_{k=0}^{K} c_k\cdot M^k\Big]\right)\sum_{k=0}^{K} c_k\cdot H^k(\omega M^{-1})\Bigg\|_{L_\infty}\Bigg\}. \end{aligned} \qquad (3.38)$$

Using (3.29), (3.31) and replacing $p_0 = 1 - p_1$, we have that for a given K, M and R, the transfer function of the optimal (in the minimax sense) second order compensation filter is

$$C(z) = 2^{-2}[4z^{-1} + p_1^*(1 - 2z^{-1} + z^{-2})]. \tag{3.39}$$

Instead of solving (3.38) for any set of parameters K, M and R given by the problem at hand, we can consider the following observations:

1. The shape of the amplitude response $H(\omega)$ changes very little with M [10]. Therefore, we can give in advance an arbitrary value to M without affecting the optimization results. Thus, we set $M = 16$.

2. Most of the times, K ranges from 2 to 7. Additionally, R usually ranges from 2 to 4.

From the first point, we have that the problem (3.38) needs only two parameters to be specified in advance (K and R) and from the second point we have the usual values of these two parameters. Therefore, we substituted $M = 16$ in (3.38) and solved (3.3.38) for $K \in [2, 15]$ and $R \in [2, 5]$, finding the proper values of p_1^* in every case. Figure 3.5 shows in grey marks the values of the resulting optimal coefficients, p_1^*. These values can be used as input information to obtain a formula to approximate a given p_1^* in terms of K and R. Using the MATLAB Curve Fitting Tool, this formula is obtained as follows,

$$p_1^* \approx p(K,R) = 0.00185 - 0.544R^{-3.3} - 0.1717K - 0.088R^{-2.578}K^2. \tag{3.40}$$

The four curves $p(K, 2)$, $p(K, 3)$, $p(K, 4)$ and $p(K, 5)$ are also shown in Figure 3.5. Note that the formula has a very accurate approximation to the optimal values. Finally, to obtain a multiplierless compensator, the approximate optimal coefficient can be rounded as $p_1^* \approx 2^{-r} \cdot round\{p(K,R)/2^{-r}\}$, with $2 \leq r \leq 6$, where $round\{x\}$ means rounding x to the nearest integer.

Example 2

In the following example shows that the proposed compensated CCFs provide a better solution for decimation filtering comparing to the traditional compensated comb filters from [11] and [12] in terms of computational complexity measured in Additions Per Output Sample (APOS).

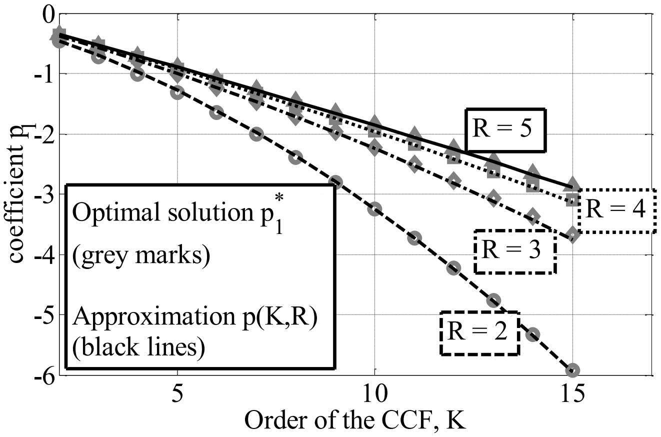


Figure 3.5. Optimal values p_1^* and their approximations using $p(K,R)$ from (3.52).

Consider M=32, R=4 and 60 dB of desired attenuation in the folding bands.

To obtain the desired attenuation, a CCF with order $K = 3$ is used. From (3.40) and with r=4 for rounding, we obtain $p_1^* \approx 2^{-4} \cdot round\{p(3,4)/2^{-4}\} = -2^{-4}(2^3+1)$. From (3.39), the transfer function of the compensator is $C(z) = 2^{-2}[4z^{-1} - (2^{-1}+2^{-4})(1-2z^{-1}+z^{-2})]$, which needs only 4 addition/subtraction operations. Figure 3.6 shows the magnitude response of the compensated 3rd-order CCF. The overall compensated

CCF has three additions working before the downsampling by 32 and 12 additions working after the downsampling, as shown in Figure 3.7. Thus, the overall computational complexity is (3×32)+12=108 APOS.

In order to get a filter with the desired attenuation, methods [11] and [12] use 4 cascaded comb filters employing the traditional Cascaded Integrator-Comb (CIC) structure (see Figure 3.8), with respective compensation filters having the transfer functions $C_1(z)=-2^{-3}[1-(2^3+2)z^{-1}+z^{-2}]$ and $C_2(z)=[(1+2^{-1}-2^{-3}-2^{-9})z^{-1}+(-2^{-3}-2^{-4}+2^{-13})(1+z^{-2})]$. Figure 3.6 also shows the magnitude responses of these filters.

Note that the proposed filter and the filter from [12] have similar passbands, but the compensation filter $C_2(z)$ (used in [12]) requires 7 addition/subtraction operations and almost twice the word-length of the proposed compensator. Moreover the overall filter using method [12] requires (4×32)+10=138 APOS. On the other hand, the compensator $C_1(z)$ used in [11] requires only three additions, and the computational complexity of the overall filter from method [11] is (4×32)+6=134 APOS. However, the passband compensation is poor and the computational complexity is still higher than that of the proposed method. Finally, Table 3.4 summarizes the aforementioned results.

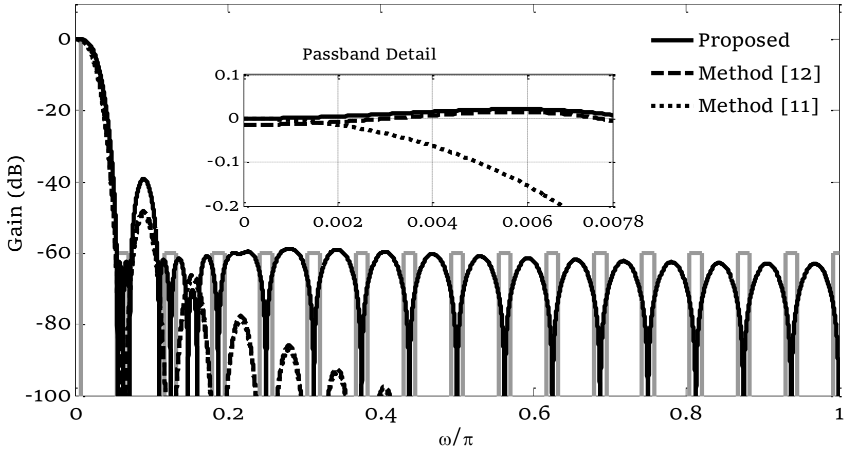


Figure 3.6. Magnitude responses of the proposed filter and filters designed with methods [11] and [12].

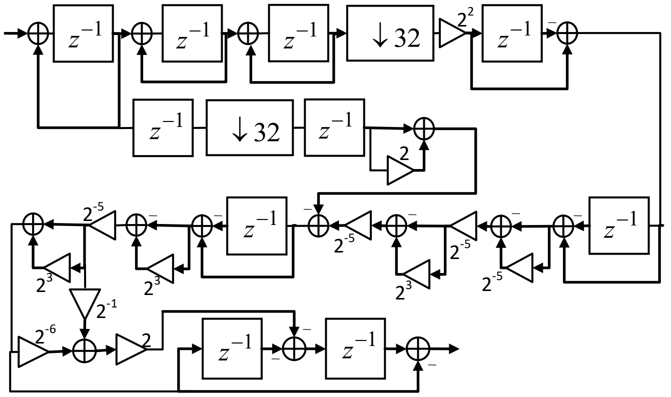


Figure 3.7: Block diagram of 3rd-order compensated CCF. Multipliers by powers of two do not have hardware cost.

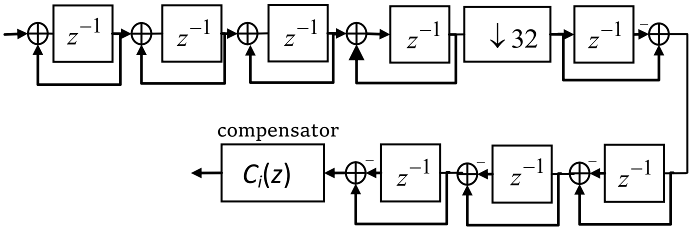


Figure 3.8. Block diagram of 4 cascaded compensated comb filters using the traditional Cascaded Integrator-Comb (CIC) structure (methods [11] and [12]). Note that i = 1 for method [11] and i = 2 for method [12].

Table 3.4. Computational complexity of filters from methods [11], [12] and proposed.

Method	Computational Complexity (APOS)
Method [11]	134
Method [12]	138
Proposed	108

3.2.3 Wide-band compensation filters design for improving the passband behavior of Cascade Integrator Comb decimators

Method [9] offers acceptable wide-band compensation with a simple second-order filter (*N*=1) requiring only four additions. However, the passband deviation may still be high. By using *N*=2, a much noticeable improvement can be obtained at the cost of little additional complexity. This is the starting point of this proposal. The following presents the proposed design method, the compensation filter structures and the details for composite decimation factors.

- *Optimization and near-optimal solution*

Let us start by substituting (3.29) in (3.31) with *N*=2. After some re-arrangement of terms, we get

$$\begin{aligned} C(z) &= z^{-2}p_0 + z^{-1}p_1[2^{-2}(1+2z^{-1}+z^{-2})] + p_2[2^{-2}(1+2z^{-1}+z^{-2})]^2 \\ &= c_0(1+z^{-4}) + c_1(z^{-2}+z^{-3}) + c_2 z^{-2}, \end{aligned} \qquad (3.41)$$

$$c_0 = 2^{-4}p_2, \quad c_1 = 2^{-2}(p_1+p_2), \quad c_2 = 2^{-3}(3p_2+4p_1+8p_0)\,. \qquad (3.42)$$

Using *N*=2, and replacing (3.30) in (3.32), we obtain

$$C(\omega,\mathbf{p}) = \mathbf{p}\cdot[1 \quad \cos^2(\omega/2) \quad \cos^4(\omega/2)]^T\,, \qquad (3.43)$$

with $\mathbf{p} = [p_0 \;\; p_1 \;\;\; p_2]$.

For $\omega = 0$, the passband error to be minimized in (3.33) can be written as $\varepsilon(0) = 1 - p_0 - p_1 - p_2$. By arbitrarily setting $\varepsilon(0) = 0$, we can express p_0 in terms of p_1 and p_2 as

$$p_0 = 1 - (p_1 + p_2)\,. \tag{3.44}$$

Upon replacing (3.44) in (3.43), and then (3.43) in (3.33), the maximum error in the passband can be minimized by finding the optimal values p_1^* and p_2^* that solve (3.33) under the minimax criterion. After performing such optimization, p_0^* is found by substituting p_1 by p_1^* and p_2 by p_2^* in (3.44).

Since the shape of the amplitude response $H(\omega,M)$ changes very little with M [10], we set $M = 16$ in (3.33) beforehand without affecting the optimization results. Moreover, K can be considered in the range 2 to 7 from a practical point of view. Thus, we solve (3.33) for the values of p_1^* and p_2^* by setting $M = 16$ and $K \in \{2,\ldots,7\}$. Figure 3.9 shows in grey marks the values of the resulting optimal coefficients. These values can be used as input data to obtain formulas to approximate p_1^* and p_2^* in terms of K. Using the MATLAB Curve Fitting Tool, these formulas are

$$p_1(K) = -0.08K^2 - 0.22K - 0.17, \tag{3.45}$$

$$p_2(K) = 0.043K^2 + 0.025K + 0.093. \tag{3.46}$$

Curves $p_1(K)$ and $p_2(K)$ are shown in Figure 3.9 as well. Note that formulas (3.45)-(3.46) represent a very accurate approximation to the optimal values. Finally, to obtain a multiplierless compensator, the approximate optimal coefficients can be rounded as

$$p_1^* \approx 2^{-r1} \cdot round\{p_1(K)/2^{-r1}\}, \tag{3.47}$$

$$p_2^* \approx 2^{-r2} \cdot round\{p_2(K)/2^{-r2}\}, \qquad (3.48)$$

with $2 \leq r_1$, $r_2 \leq 6$. In the two previous equations $round\{x\}$ means rounding x to the nearest integer.

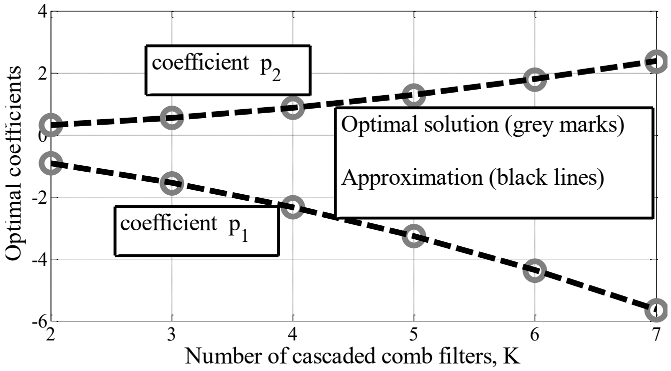


Figure 3.9. Optimal values p_1^* and p_2^* along with their approximations using $p_1(K)$ and $p_2(K)$ from (3.45) and (3.46).

- *Wideband compensator structures*

From (3.41), we can see that filter $F(z)=2^{-2}[1+2z^{-1}+z^{-2}]$ is repeated twice, resembling the well-known sharpening architecture from [13]. The repeated use of the same subfilter can be avoided with the Pipelining-Interleaving (PI) technique in [14]. In this case, the subfilter $F(z^2)$ is implemented only once, and its clock operates at twice the output sampling rate. Figure 3.10 shows the resulting PI-based structure.

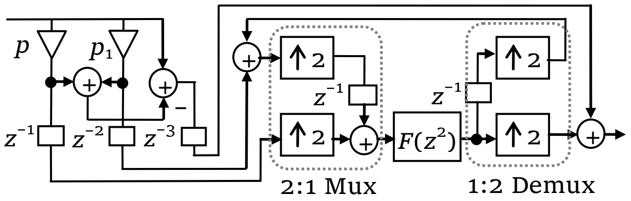


Figure 3.10. PI-based structure with a multiplexed subfilter.

Equation (3.41) presents the symmetric transfer function of the compensator as well. Upon replacing (3.44) in (3.42), it can be shown that $c_2 = 1–2(c_0+c_1)$. This leads to the structure presented in Figure 3.11. Whenever the number of adders required by the coefficients c_0 and c_1 is equal or less than the number of adders required by coefficients p_1 and p_2, it is better to use the structure shown in Figure 3.12. These structures are convenient if the compensator is expected to operate at the output sampling rate. Note that coefficients c_0, c_1 and c_2 are determined by first finding p_1 and p_2 using (3.47) and (3.48), then finding p_0 with (3.44), and finally using p_0, p_1 and p_2 in (3.42).

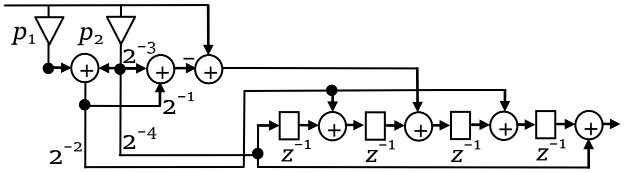


Figure 3.11. Single-rate structure.

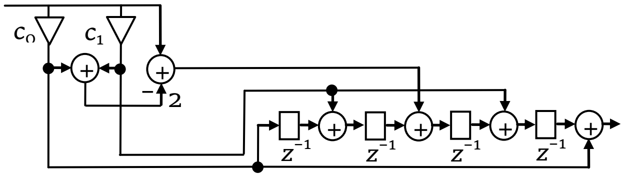


Figure 3.12. Single-rate structure with coefficients c_0 and c_1 that should be used if the number of adders required by c_0 and c_1 is equal or less than the number of adders required by p_1 and p_2.

- *The case of a composite decimation factor*

When M can be factorized into $M = M_1M_2$, we propose to use the two-stage approach presented in [15], where the downsampler M is split into two downsamplers, M_1 and M_2, and a comb-based decimator

$H_{TS}(z)=H^{K1}(z,M_1)\times$ $H^{K2}(z^{M1},M_2)\times G(z^M)$ is adopted ($G(z^M)$ is a compensator).

From multirate identities, $H^{K2}(z^{M1},M_2)$ can be moved after the downsampler by M_1 and $G(z^M)$ after the downamplers by M_1 and M_2. The worst-case attenuation of the overall filter $H_{TS}(z)$ is improved by increasing K_2.

The guidelines that we follow to choose M_1 and M_2 are the same as proposed in method [15], namely, selecting these values to be as close integers as possible. Thus, the improvements to method [15] consist in the following:

1) Choice of K_1 and K_2: Considering that a desired attenuation $|A|$ in dB must be met in all the stopbands, in [15] the authors proposed to increase K_2 at least by 1 for each 10 dB increment of $|A|$ and to choose K_1 such that $K_1 \geq \lfloor K_2/2 \rfloor+1$. However, we propose to use:

$$K_1 \geq \lceil -|A|/20\log_{10}\{|H(\omega_1,M_1)|\} \rceil, \qquad (3.49)$$

$$K_2 \geq \lceil -|A|/20\log_{10}\{|H(\omega_2,M_2)|\} \rceil, \qquad (3.50)$$

$$\omega_1=(2\pi/M_1 - \omega_p), \qquad \omega_2=(2\pi/M_2 - M_1\omega_p). \qquad (3.51)$$

2) Choice of the compensator: In [15], the compensation filter is designed with method [16]. On the other hand, we use the method detailed above. The coefficients p_1 and p_2 are obtained from (3.47) and (3.48) by replacing K by K_2.

Example 3

In the following example is showing how the proposed wide-band compensation filters provide a better solution in comparison to others.

For a fair comparison, we assume that all the compensators are operated at the output sampling rate. Therefore, single-rate structures are used.

Consider $M=17$ and $K=5$ cascaded comb filters to attain an attenuation $A=45$dB in the stopbands.

In this example, we compare the proposed compensator with filters from [9], [12] and [17]. Methods [9] and [17] offer the best near-optimal wide-band second-order compensators, whereas method [12] presents fourth-order multiplierless optimal solutions for values of K up to 5. Figure 3.13 shows the passband magnitude characteristics of the comb filter, the proposed filter and filters from [9], [12] and [17].

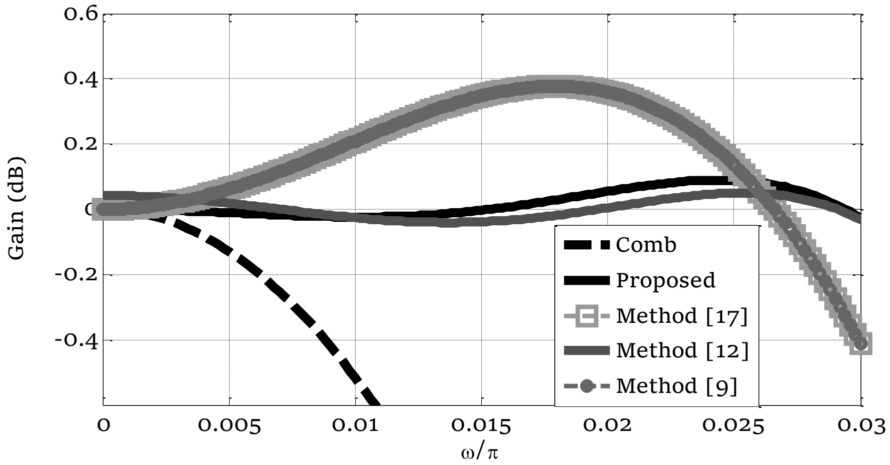


Figure 3.13. Magnitude responses of filters from [9], [12], [17] and proposed.

Using $r_1 = r_2 = 3$ in (3.47)-(3.48), we obtain $p_1 = -2^{-2}\times(2^4-2^2+1)$ and $p_2 = 2^{-2}\times(2^2+1)$. Replacing these values in (3.44) and putting that substitution in (3.42), we get $c_0 = 2^{-6}\times(2^2+1)$ and $c_1 = -2^{-1}$. Note that coefficients p_1 and p_2 need 3 additions while coefficients c_0 and c_1 can be implemented with 1 addition. Thus, we use the structure of Figure 3.12. The resulting compensator requires 7 additions and 4 delays. The solutions from [9] and [17] are actually the same, but method [9] requires only 4 adders, whereas 5 adders are used in [17]. The proposed technique and method [12] present 4-th order filters with much better passband characteristics at the cost of increased complexity. Even though the filter from [12] has a slightly better frequency response, it needs 14 adders and a specialized optimization to obtain the filter coefficients. The proposed method provides a near-optimal solution with 50% of savings in arithmetic complexity when compared to [12].

3.3 Computationally-efficient CIC-based filter with embedded Chebyshev sharpening

In this proposal the scheme Chebyshev-sharpened comb filter was introduced. The proposed filter uses a low-complexity passband droop compensator and the Chebyshev sharpening technique to improve the magnitude response. In this way this method improves the worst-case aliasing rejection and simultaneously decreases the passband deviation of traditional comb decimation filters. The magnitude response improvement of the comb filter was made by the following:

- The efficient use of the Chebyshev sharpening scheme from [2], performed to improve the attenuation in the folding bands.
- The efficient adaptation of the recent simple compensation filter from [9] with the aim to decrease the passband droop.

3.3.1 Embedding a filter into a CIC structure

Let us consider a decimation filter with $M = M_1M_2M_3$. The first stage consists of K_1 cascaded comb filters, the second stage of K_2 cascaded comb filters and the third stage is an auxiliary filter $G(z)$. The overall decimation filter has the transfer function referred to high rate given by

$$H_D(z) = \left[\sum_{i=0}^{M_1-1} z^{-i}\right]^{K_1} \left[\frac{1 - z^{-M_1M_2M_3}}{1 - z^{-M_1}}\right]^{K_2} G(z^{M_1M_2}) \qquad (3.52)$$

The first-stage can be implemented in a non-recursive form and the polyphase decomposition can be applied, thus resulting in power savings [9]. The polyphase decomposition is denoted by $P_1(z)$ to $P_M(z)$ as shown in Figure 3.14. The K_2 cascaded comb filters are implemented in a traditional CIC structure. The filter $G(z^{M1M2})$ can be moved after the downsampler by M_2. This results in the structure of Fig. 3.14.

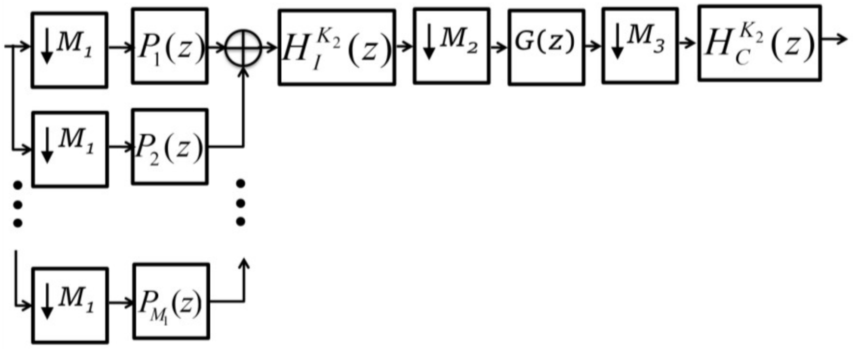


Figure 3.14. Efficient Comb-based structure aided with an auxiliary filter $G(z)$.

The auxiliary filter $G(z)$ has the following tasks:

1) Decrease the passband droop in the band of frequencies spanning the interval from 0 to ω_c, where ω_c is given by

$$\omega_c = \frac{\pi}{M_3 R}\,. \qquad (3.53)$$

where R is the residual factor.

2) Improve the attenuation at least in the band of frequencies spanning the interval from $\omega_{1,a}$ to $\omega_{1,b}$, with these frequencies given by

$$\omega_{k,a} = \frac{2\pi k}{M_3} - \frac{\pi}{M_3 R}, \qquad (3.54)$$

$$\omega_{k,b} = \frac{2\pi k}{M_3} + \frac{\pi}{M_3 R}, \qquad (3.55)$$

$$k = 1, 2, \ldots, \left\lfloor \frac{M_3}{2} \right\rfloor. \qquad (3.56)$$

(The aforementioned bands of frequencies are referred to the downsampled-by-(M_1M_2) sampling rate and $\lfloor x \rfloor$ means rounding to the nearest integer less than or equal to x.)

3) Have a simple and regular structure with few adders.

Consider the filter $G(z)$ given as:

$$G(z) = H_3(z)C(z^{M_3}), \qquad (3.57)$$

where $H_3(z)$ is a comb filter given by

$$H_3(z) = \left[\frac{1 - z^{-M_3}}{1 - z^{-1}}\right]^{K_3}, \qquad (3.58)$$

and $C(z^{M3})$ is the compensation filter with the following desirable properties:

- It works at low sampling rate,
- It is a multiplierless filter.

Additionally, according to (3.58), filter $G(z)$ has the following characteristics:

- Introduces K_3 zeros in the center of all the bands defined by the frequencies (3.54) and (3.55).

It is worth highlighting that, in general, $C(z^{M3})$ can be any compensator from literature, whereas $H_3(z)$ can be any filter that improves the attenuation at least in the band delimited by the frequencies $\omega_{1,a}$ and $\omega_{1,b}$, i.e., the first folding band. This opens the options for the choice of the filter $H_3(z)$, which might be, for example, any comb-based filter with zero-rotation characteristic or the recent Chebyshev-sharpened CIC filter from [2]. Obviously, $G(z)$ must preserve simplicity and it must use modulo arithmetic for overflow-handling characteristics.

3.3.2 Chebyshev sharpening applied into the proposed structure

Chebyshev sharpening is applied to the filter into the proposed structure with $M = M_1M_2M_3$. We use $H_3(z)$ as Chebyshev-sharpened filter in (3.57). Similarly, we choose the compensator $C(z)$ in (3.57) from recent method [9]. In this way, $H_3(z)$ improves the attenuation in the first folding band where the worst-case attenuation occurs, whereas $C(z)$ compensates for the passband droop.

The transfer function $H_3(z)$ is given by [2]

$$H_3(z) = \sum_{k=0}^{N} z^{-(N-k)M_3/2} \cdot c_k \cdot \left[\gamma z^{-1} H_{3b}(z)\right]^k , \tag{3.59}$$

where c_k is the coefficient of the k-th power (with $0 \leq k \leq N$) of a N-th degree Chebyshev polynomial of first kind.

$$H_{3b}(z) = \begin{cases} \left[\dfrac{1 - z^{-M_3}}{1 - z^{-1}}\right]; & M_3 > 2, \\ (1 + z^{-1}); & M_3 = 2, \end{cases} \tag{3.60}$$

$$\gamma = 2^{-L_1} \left\lfloor \left| 2^{L_1} \cdot \frac{\sin(\omega_{1,a}/2)}{\sin(\omega_{1,a} M_3/2)} \right| \right\rfloor . \tag{3.61}$$

where L_1 is the word-length for the fractional part of the Signed Powers of Two (SPT) representations of γ. Moreover, L_1 is usually equal or greater than 2.

The transfer function $C(z)$ is given by [9]

$$C(z) = 2^{-2}[4z^{-1} + B(-1 + 2z^{-1} - z^{-2})], \tag{3.62}$$

where B is the compensation parameter.

Placing (3.57) in (3.52), our proposed decimation filter is given by

$$H_D(z) = \left[\sum\nolimits_{i=0}^{M_1-1} z^{-i}\right]^{K_1} \left[\frac{1 - z^{-M_1M_2M_3}}{1 - z^{-M_1M_2}}\right]^{K_2} H_3(z^{M_1M_2})C(z^{M_1M_2M_3}), \tag{3.63}$$

where $H_3(z)$ is given in (3.59) and $C(z)$ is given in (3.62).

The design method consists in finding the values of K_1 (the number of cascaded comb filters in the first stage), K_2 (the number of cascaded comb filters in the CIC structure), N (the order of the Chebyshev-sharpened filter $H_3(z)$), B (the compensation parameter) and M_1, M_2 and M_3 (the decimation factors) that allow accomplishing the following goals:

- A droop correction in the passband given by

$$\omega_p = \frac{\pi}{MR}. \tag{3.64}$$

  where $M = M_1M_2M_3$.

- A desired attenuation A in the folding bands.

A heuristic solution consists in choosing $M_2 \geq M_3 \geq M_1$, with M_2 and M_3 close in values as much as possible. To find K_1 we use the smallest value that satisfies

$$K_1 \geq \left\lceil -\frac{|A|}{20\log_{10}\{|v_1|\}} \right\rceil, \tag{3.65}$$

$$v_1 = \frac{\sin(M_1\omega_1/2)}{M_1\sin(\omega_1/2)}, \qquad \omega_1 = \frac{2\pi}{M_1} - \frac{\pi}{M_1M_2M_3R}. \tag{3.66}$$

Then, we find K_2 as the smallest value that satisfies

$$K_2 \geq \left\lceil -\frac{|A|}{20\log_{10}\{|v_2|\}} \right\rceil, \tag{3.67}$$

$$v_2 = \frac{\sin(M_2M_3\omega_2/2)}{M_2M_3\sin(\omega_2/2)}, \qquad \omega_2 = \frac{2\pi}{M_3} - \frac{\pi}{M_2M_3R}, \tag{3.68}$$

and to find N we use the smallest value that satisfies

$$N \geq v - \left\lfloor \frac{6v}{(6 - 20\log_{10}\{|w|\})} \right\rfloor, \tag{3.69}$$

$$v = \left\lceil -\frac{|A| + K_2 \cdot 20\log_{10}\{|v_3|\}}{w} \right\rceil, \tag{3.70}$$

$$v_3 = \frac{\sin(M_2M_3\omega_3/2)}{M_2M_3\sin(\omega_3/2)}, \qquad \omega_3 = \frac{2\pi}{M_2M_3} - \frac{\pi}{M_2M_3R}, \tag{3.71}$$

$$w = \frac{\sin(M_3\omega_4/2)}{M_3\sin(\omega_4/2)}, \qquad \omega_4 = \frac{2\pi}{M_3} - \frac{\pi}{M_3R}, \tag{3.72}$$

where $\lceil x \rceil$ means rounding to the nearest integer greater than or equal to x. Finally, the compensation parameter B can be found in terms of K_2 and N, since the contribution on the passband droop of the first-stage filter due to K_1 can be neglected. Table 3.5 shows typical values for B when the residual decimation factor is R = 2.

Table 3.5 Rounded compensation parameter B for a residual decimation factor R = 2.

$K_2 + N$	B
2	2^{-1}
3	$2^{-1} + 2^{-2}$
4	2^{0}
5	$2^{0} + 2^{-2}$
6	$2^{0} + 2^{-1}$
7	$2^{0} + 2^{-1}$
8	2^{1}

Example 4

Let us consider the following examples to show the magnitude response characteristics obtained with the proposed method in comparison with the traditional CIC filter, a three-stage CIC-based structure, method [18] and a three-stage filter based on method [18]. For a fair comparison, we have adapted the compensator from [9] to these filters, in order to obtain passband droop correction in all the cases.

In the first example we compare with the traditional CIC structure and also with a three-stage structure based on the architecture of Figure 3.14, where $G(z)$ is given in (3.57) and $H_3(z)$ is given in (3.58), with $H_3(z)$ implemented in recursive form.

Consider a decimation factor M = 20, a residual decimation factor R = 2 and a desired attenuation A = 80 dB.

We factorize M into M_1=2, M_2=5 and M_3=2. Using L_1=2 in (3.73) we obtain $\gamma=2^{-2}\times5$ ($\gamma^2=2^{-4}\times25$). From (3.65)-(3.72) we obtain K_1=3, K_2=5, and N=3. The compensation parameter is $B = 2^1$. The proposed scheme has 10 adders working at the downsampled-by-M_1 sampling rate, 3 adders working at the downsampled-by-(M_1M_2) sampling rate and 13 adders working at the output sampling rate, resulting in 119 Additions Per Output Sample.

The traditional CIC filter requires K = 9 integrators working at high rate and 9 comb filters working at low rate, plus 4 adders for the compensator, resulting in 193 APOS. On the other hand, the three-stage CIC-based scheme has 10 adders working at the downsampled-by-M_1 sampling rate, 5 adders working at the downsampled-by-(M_1M_2) sampling rate and 14 adders working at the output sampling rate, resulting in 124 Additions Per Output Sample.

Figure 3.15 shows the magnitude responses of the proposed filter, the original CIC filter and the three-stage CIC-based filter. Note that these filters accomplish the desired attenuation, whereas the passband characteristic of the proposed filter is slightly better. Table 3.6 summarizes the results for this example.

Table 3.6. Comparison of characteristics of Example 4.

Method	APOS	Max. passband deviation	Min. stopband attenuation
CIC filter	193	-0.94 dB	-87 dB
Three-stage CIC-based filter	124	-0.9 dB	-84.2 dB
Proposed	119	-0.76 dB	-84.1 dB

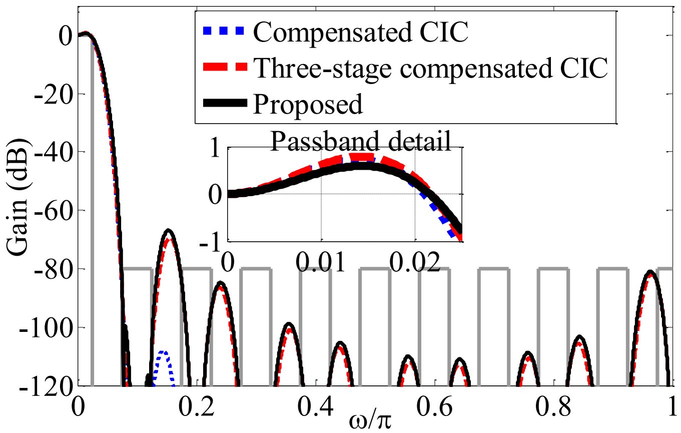


Figure 3.15. Magnitude responses for traditional CIC filter, three-stage CIC-based filter and proposed, with M = 20 and R = 2.

3.4 Implementation of a Comb-based decimator that consists of an area-efficient structure aided with an embedded simplified Chebyshev-sharpened section

As a result of this research the implementation of, single-rate version, recursive Chebishev-CIC filter was carried out. A CIC-based structure was achieved with premodified subfilter modified using a Chebyshev of second order. Through an appropriate modification of the simplest case of the Chebyshev sharpening method, partially regulated, a structure for low complexity decimation was obtained, where it is allowed to independently change M1 and M2. In order to obtain adequate attenuation, only a simple configurable coefficient expressed in power of two needs to be adjusted when M2 varies. Due to the above characteristics, the proposed method is considered partially regular. It was found that, for the same attenuation in the folding bands, the bus width is smaller than the bus widths of the traditional CIC filter and the recursive two-stage CIC-based filters where decimation factor can be modified online. Compared to the original CIC structure as well as other partially regular methods the proposed architecture performs fewer operations per output sample.

Reducing the sampling rate by an integer factor is an ubiquitous process in multi-standard reconfigurable receivers [19]. This decimation is performed in stages, usually as shown in Figure 3.16. In order to reduce the hardware utilization of the power-efficient but area-demanding polyphase arrays, F is typically set to a fixed small integer, whereas the last stage is a half-band decimator. Hence, the middle stage is based on a compact Cascaded Integrator-Comb (CIC) filter to allow M to be large and able to change with little hardware utilization even if on-line reconfiguration is needed.

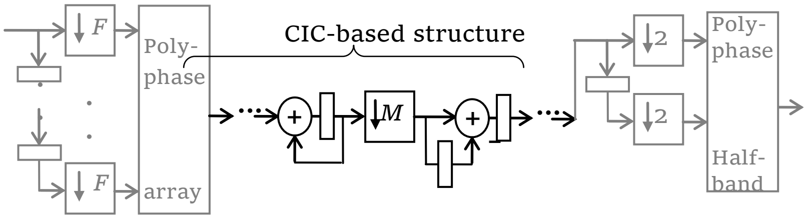


Fig. 3.16. Typical decimation chain.

A new solution for the aforementioned CIC's two problems is presented, with the following characteristics: 1) M is non-prime ($M = M_1 \times M_2$) in order to operate some integrators at a lower rate (decreased by M_1) and thus reducing their power dissipation; 2) M_2 is an small prime between 2 and 7 in order to bound the bus width growth. The resulting system does not compromise the regularity in a great deal because many downsampling factors can be used in the proposed structure.

Proposed solution: Let us split into two terms the transfer function (referred to high rate) of a traditional CIC with K cascaded stages, i.e.,

$$H(z)=\left[\frac{1-z^{-M_1M_2}}{1-z^{-1}}\right]^K=\left[\frac{1-z^{-M_1}}{1-z^{-1}}\right]^K\times\left[\frac{1-z^{-M_1M_2}}{1-z^{-M_1}}\right]^K. \tag{3.73}$$

Since the term $[(1-z^{-M1M2})/(1-z^{-M1})]$ contributes more to the attenuation in the 1st folding band, where the worst-case attenuation occurs, we arbitrarily set K_2+2 cascaded stages for that term and K_1 cascaded stages for the 1st term, with $K_2 \geq K_1$. We denote the resulting filter as $G(z)$,

$$G(z)=\left[\frac{1-z^{-M_1}}{1-z^{-1}}\right]^{K_1}\times\left[\frac{1-z^{-M_1M_2}}{1-z^{-M_1}}\right]^{K_2}\times\left[\frac{1-z^{-M_1M_2}}{1-z^{-M_1}}\right]^{2}. \tag{3.74}$$

In order to improve the worst-case attenuation, we strategically spread two zeros around the first folding band by replacing the term $[(1-z^{-M1M2})/(1-z^{-M1})]^2$ of (3.74) with a CIC filter sharpened with a second-degree Chebyshev polynomial of first kind (that polynomial is denoted by $T_2(x) = -1+2x^2$, see eq. (2.57) in [20]). The transfer function of the sharpened filter is

$$C(z)=\left\{-z^{-M_1(M_2+1)}+2\left[\gamma z^{-M_1}\,\frac{1-z^{-M_1M_2}}{1-z^{-M_1}}\right]^2\right\}. \tag{3.75}$$

The coefficient γ is introduced to keep the zeros into the desired folding band and it must be tuned for every value M_2. Thus, for the sake of regularity, we constrain M_2 to be any small prime between 2 and 7, and we look for a simple power-of-2 representation of γ that can be reconfigured for these values M_2 without needing multipliers or adders, but just an adjustable arithmetic shift called S. With the aforementioned modifications, we arrive to the proposed transfer function (referred to high rate),

$$H_p(z) = \left[\frac{1}{1-z^{-1}}\right]^{K_1} \times \left[\frac{1}{1-z^{-M_1}}\right]^{K_2-K_1} \times \left[1-z^{-M_1M_2}\right]^{K_2} \times \left\{ -z^{-M_1(M_2+1)} + 2^S \left[z^{-M_1} \frac{1-z^{-M_1M_2}}{1-z^{-M_1}} \right]^2 \right\}, \tag{3.76}$$

where S can be chosen according to Table 3.7. The proposed fully pipelined architecture, presented in Figure 3.17 with details in Figure 3.18, is obtained after 1) applying multi-rate identities, 2) cancelling numerators and denominators of the form [1–z^{-M1}] and 3) inserting pipeline registers. That structure uses K_2+2 integrator-comb pairs and it is efficient because, for the common desired attenuations, $K_2+2 < K$ usually holds, making our system to need fewer integrators than a CIC. Moreover, just K_1 integrators work at the high-rate section.

Table 3.7. *Values of the shift S for the first four prime factors M_2*

M_2	S	M_2	S	M_2	S	M_2	S
2	1	3	0	5	-1	7	-2

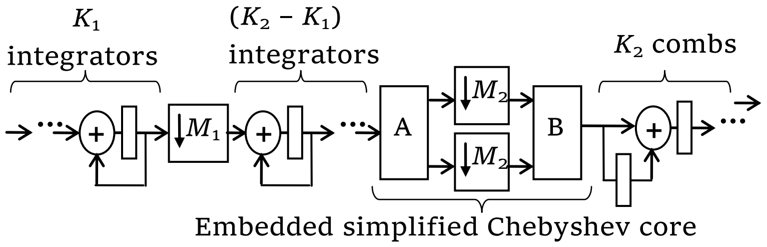


Figure 3.17. Proposed CIC-based fully pipelined structure.

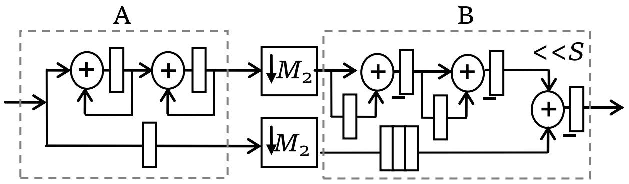


Figure 3.18. Detail of the blocks A and B that compose the Chebyshev core.

The number of integrator-comb pairs in the proposed structure (K_2+2) and the number of integrators working at the high-rate section (K_1), necessary to accomplish 60 dB, 70 dB, 80 dB and 90 dB of worst-case attenuation, are presented in Table 3.8 for values M ranging from 8 to 512. The number of integrator-comb pairs for the classical CIC filter (K, which is the number of integrators working at the high-rate section in the CIC structure) is also shown. M_1 and M_2 were chosen depending on what structure needed the less overall amount of integrators, and this choice turned out to obey a simple rule: M_2 must be as large as possible (for instance, for $M = 2^p$, with $3 \leq p \leq 9$, we use $M_2 = 2$, whereas for $M = 14$ we use $M_2 = 7$). From Table 3.8 we observe that in most cases the number of integrator-comb pairs used in the proposed structure is less than the number of pairs used in the classical CIC, and the number of integrators working at the high-rate section is reduced by a half on average.

The aforementioned advantages can not be exploited neither for values M where the smallest prime factor M_2 is greater than 7 nor for prime factors M (which in total is just about 23% of all the values M between 8 and 512). However, the usefulness of the proposed structure can be extended if we keep decreasing the arithmetic shift S (see Table 1) for primes M_2 greater than 7, taking into account that the bus grows one bit for every decrement in S.

Example 5

Finally, an example for M = 33 (M_1 = 11 and M_2 = 3), with 80 dB of desired attenuation, has been synthesized into the Altera's Cyclone-IV FPGA chip (device EP4CE115F29C7) for a detailed comparison. This chip is currently used on the DE2-115 development kit, popular at most universities. The operation of the proposed filter was simulated with an 8-bit 608 KHz cosine signal as input, sampled at 160 MHz. Power Play Power Analyzer was employed for the estimation of power dissipation, using the Value Change Dump data generated by ModelSim to get an estimation with high level of confidence. TimeQuest Timing Analyzer was employed for the estimation of performance, using the slow 85C timing model (the worst-case scenario). Post place-and-route results are presented in Table 3.9, where we notice the benefits of the proposed system.

Table 3.8. Number of integrator-comb pairs used in the CIC and proposed structures for values M between 8 and 512.

	M_2 = 2 (116 cases)	M_2 = 3 (114 cases)	M_2 = 5 (87 cases)	M_2 = 7 (72 cases)
60 dB K=6	K_1=4 K_2+2=6	K_1=3 K_2+2=5	K_1=3 K_2+2=5	K_1=2 K_2+2=5
70 dB K=7	K_1=4 K_2+2=6	K_1=4 K_2+2=6	K_1=3 K_2+2=6	K_1=3 K_2+2=6
80 dB K=8	K_1=5 K_2+2=7	K_1=4 K_2+2=7	K_1=3 K_2+2=7	K_1=3 K_2+2=7
90 dB K=9	K_1=5 K_2+2=8	K_1=4 K_2+2=8	K_1=4 K_2+2=8	K_1=5 K_2+2=7

Table 3.9. Comparison of the proposed structure with other CIC-based decimators in terms of synthesis results (Note: LE = Logic Element).

	CIC	[2]	[21]	Proposed
Worst-case attenuation	83.68 dB	86.3 dB	84.84 dB	87.95 dB
Hardware utilization	1238 LEs	1432 LEs	5007 LEs	842 LEs
Estimated power dissipation	188.78 mW	195.58 mW	279.97 mW	172.96 mW
Maximum frequency of operation	191.46 MHz	168.83 MHz	166.97 MHz	214.73 MHz

3.5 Comb-based decimation filter design based on Improved sharpening

To improve both passband and stopband characteristics of a comb filter the improved sharpening approach of Hartnett an Boudreoux [22] is adopted. In [23] a general formula was deduced to obtain directly the desired amplitude change function from the design parameters. The formula is given by

$$P_{\sigma,\delta,m,n}(x) = \delta x + \sum_{j=n+1}^{R} (\alpha_{j,0} - \sigma\alpha_{j,1} - \delta\alpha_{j,2})x^{j}\ , \tag{3.77}$$

where $R = n + m + 1$ and

$$\alpha_{j,0} = \sum_{i=n+1}^{j} (-1)^{j-i} \binom{R}{j}\binom{j}{i}, \quad \alpha_{j,1} = \sum_{i=n+1}^{j} (-1)^{j-i} \binom{R}{j}\binom{j}{i}\left(1-\frac{i}{R}\right),$$
$$\text{and} \quad \alpha_{j,2} = \sum_{i=n+1}^{j} (-1)^{j-i} \binom{R}{j}\binom{j}{i}\frac{i}{R}. \tag{3.78}$$

By taking advantage of the two-stage decomposition of the comb filter to apply the sharpening technique only in the second stage. The resulting transfer function is given by:

$$H(z)=\left[H_1(z)\right]^L\cdot Sh\left\{\left[H_2(z^{M_1})\right]^K\right\}, \tag{3.79}$$

$$H_1(z)=\frac{1}{M_1}\cdot\frac{1-z^{-M_1}}{1-z^{-1}},\quad H_2(z)=\frac{1}{M_2}\cdot\frac{1-z^{-M_2}}{1-z^{-1}}, \tag{3.80}$$

where $M = M_1M_2$ is the decimation factor, L and K are the number of cascaded filters $H_1(z)$ and $H_2(z^{M1})$, respectively, and $Sh\{H(z)\}$ means that sharpening has been applied to $H(z)$. The value K must be even [15].

The advantages of this approach are the following:

- The down-sampling block M can be divided into two separated down-sampling blocks, M_1 and M_2. Since the first folding band, where the worst case attenuation occurs, is essentially determined by $H_2(z^{M1})$, it is only required to apply sharpening to this filter. As a result we get better passband and stopband characteristics with lower complexity than applying sharpening to the original single stage comb filter.

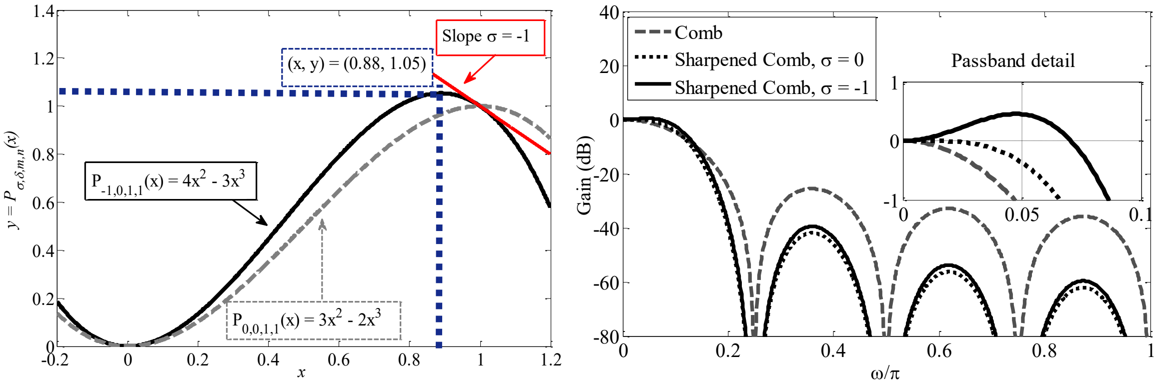
- The filter $H_2(z^{M1})$ can be moved after the down-sampling by M_1, resulting in lower power consumption because $H_2(z)$ works at a lower rate.

- The filter $H_1(z)$ can work at a lower rate after the down-sampling by M_1 using polyphase decomposition [23].

However, regardless of the passband improvement by the sharpened filter of the second stage, the resulting filter has always a passband droop that is a consequence of the first-stage comb filter. This can not be solved using the traditional sharpening proposed by Kaiser and Hamming [13]. In this proposal we will apply the improved sharpening technique to the compensated comb filter of the second stage. As a result, we can take advantage of taking into account the slope parameter σ, and thus correcting the aforementioned effect.

- *Sharpening of the second-stage comb filter*

Observe in the Figure 3.19(a) that, by setting a negative slope σ, the amplitude values over the axis x, that are slightly less than one, can be mapped into values greater than one. Since the comb filters have amplitude values slightly less than one in their passband region, they will have values greater than one after being sharpened. Thus, after cascading the sharpened second-stage comb filter with the first-stage comb filter a compensated droop in the passband region can be obtained. On the other hand, knowing that the desired stopband amplitude values are zero, the slope δ has to be equal to zero.



(a) (b)

Figure 3.19. (a) The traditional sharpening polinomial $P_{0,0,\ 1,\ 1}(x)$ = $3x^2$ – $2x^3$ and the generalized sharpening polinomial $P_{-1,0,\ 1,\ 1}(x)$ = $4x^2$ – $3x^3$. (b) Magnitude responses of a comb filter, a sharpened-comb filter with the traditional polynomial $3x^2$ – $2x^3$ and a sharpened-comb filter with the polynomial $4x^2$ – $3x^3$, obtained from the generalized approach.

Figure 3.19(a) shows a comparison of the traditional 3rd-order polynomial of Kaiser and Hamming with parameters σ = 0, δ = 0, m = 1 and n = 1, $P_{0,0,\,1,\,1}(x) = 3x^2 - 2x^3$, and a polynomial with parameters σ = –1, δ = 0, m = 1 and n = 1, $P_{-1,0,\,1,\,1}(x) = 4x^2 - 3x^3$, obtained from the generalized sharpening approach. Note that the value 0.88 is mapped to a new value greater than one, 1.05. Figure 3.19(b) shows a comparison between the magnitude responses of a comb filter, a comb filter sharpened with the polynomial $3x^2 - 2x^3$ and a comb filter sharpened with the polynomial $4x^2 - 3x^3$. Observe that the attenuations around the zeros are very similar for both sharpened comb filters. However, the sharpened comb which uses the generalized approach, has a resulting passband with increased amplitudes over the frequencies $\omega = 0$ to $\omega \approx 0.05\pi$. This characteristic can be used to compensate the droop introduced by the first-stage comb filter.

- *Sharpening of the compensated second-stage comb filter*

In Figure 3.20 we have, on the right side, the amplitudes of three filters: a comb filter and two different compensated comb filters. One of them has been compensated with a wideband compensator and the other with a narrowband compensator. On the left side we have the mapping from the original values to new values through the polynomial $4x^2 - 3x^3$. Observe that, at the frequency point ω_p, which represents the upper edge of the passband of interest, the amplitude of the comb filter is mapped to a value that is away from the desired line with slope σ. Moreover, since this line only approximates the necessary values to compensate the droop of the first-stage comb filter, it is not convenient to map values of the original amplitude that are too far from 1. Additionally, it can be seen that the original amplitude values of the comb filter compensated with a wideband compensator (which are greater than one), are mapped to new amplitude values less than one. For this reason it is not convenient to use a wideband compensator. On the other hand, the original amplitude values of the comb compensated with a narrowband compensator are mapped to values greater than one that closely follow the values of the desired line.

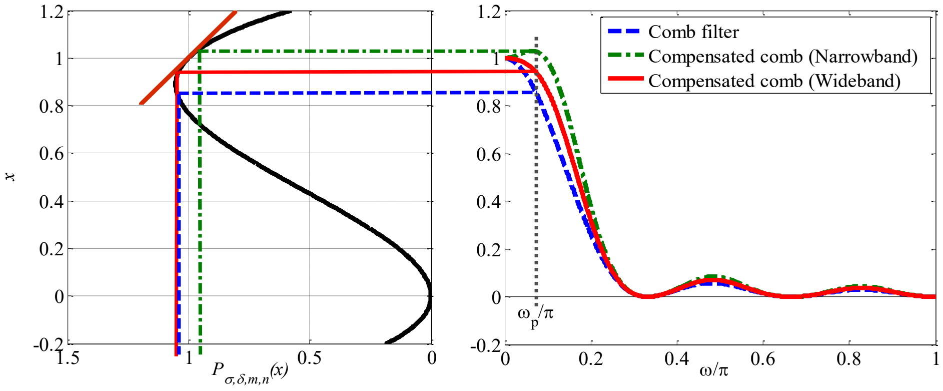


Figure 3.20. Amplitude changes of a comb filter and two compensated comb filters through the sharpening polynomial $4x^2 - 3x^3$.

A simple multiplierless compensator with only one parameter b, which depends on the number of K stages, was proposed in [24]. This filter has a low complexity and provides a good compensation in a narrow passband. Therefore, we adopt this compensation filter in this proposal. The transfer function of this compensator is

$$G(z^M) = -2^{-(b+2)}\left[1-(2^{b+2}+2)z^{-M}+z^{-2M}\right]. \qquad (3.81)$$

The compensated second-stage filter becomes,

$$H_{2C}(z) = G(z^M)\left[H_2(z^{M_1})\right]^K. \qquad (3.82)$$

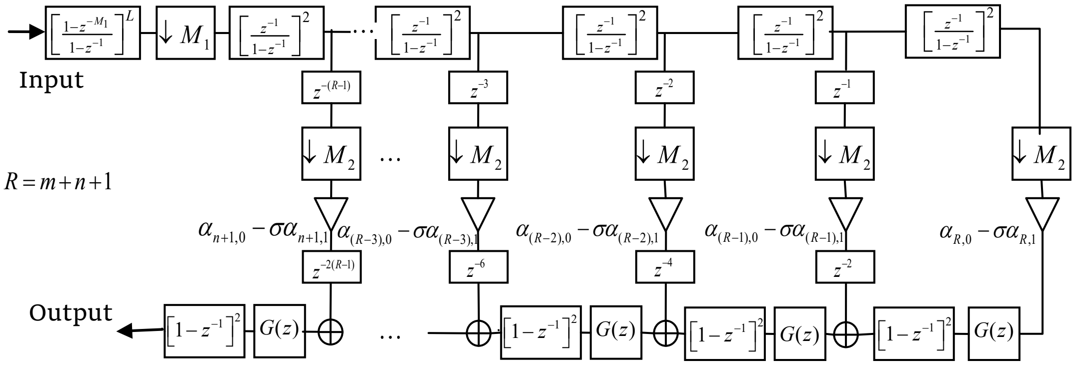
Applying the generalized sharpening technique to the compensated filter $H_{2C}(z)$ we obtain the proposed decimation filter whose transfer function is

$$H_P(z) = \left[H_1(z)\right]^L \cdot Sh\left\{H_{2C}(z)\right\}. \tag{3.83}$$

Using (3.77), (3.78), (3.80) and (3.83) we arrive at:

$$\begin{aligned} H_P(z) = &\left(\tfrac{1}{M_1}\tfrac{1-z^{-M_1}}{1-z^{-1}}\right)^L \cdot \sum_{j=n+1}^{n+m+1} (\alpha_{j,0} - \sigma\alpha_{j,1}) \times \\ &\sum_{j=n+1}^{n+m+1} \left\{ \left(\tfrac{1}{M_2}\tfrac{1-z^{-M_1 M_2}}{1-z^{-M_1}}\right)^K \left(-2^{-(b+2)}\left[1-(2^{b+2}+2)z^{-M_1 M_2} + z^{-2M_1 M_2}\right]\right)\right\}^j z^{-(n+m+1-j)\tau} \end{aligned} \tag{3.84}$$

where τ is equal to $M_1(M_2 - 1)K/2 + M_1M_2$. The coefficients $\alpha_{j,0}$ and $\alpha_{j,1}$ in (3.84) are calculated from (3.78). Thus, the design parameters are the tangencies m and n, the slope σ, and the compensator parameter b, along with the number of cascaded filters L for $H_1(z)$ and K for $H_2(z)$. An efficient structure for decimation is presented in Figure 3.21, straightforwardly derived from [25]. Note that the filter preceding the down-sampler by M_1 can be decomposed into polyphase components to avoid operations at high rate.



Input
$\left[\frac{1-z^{-M_1}}{1-z^{-1}}\right]^L$
$\downarrow M_1$
$\left[\frac{z^{-1}}{1-z^{-1}}\right]^2$
$z^{-(R-1)}$
z^{-3}
z^{-2}
z^{-1}
$\downarrow M_2$
$R = m+n+1$
$\alpha_{n+1,0} - \sigma\alpha_{n+1,1}$
$\alpha_{(R-3),0} - \sigma\alpha_{(R-3),1}$
$\alpha_{(R-2),0} - \sigma\alpha_{(R-2),1}$
$\alpha_{(R-1),0} - \sigma\alpha_{(R-1),1}$
$\alpha_{R,0} - \sigma\alpha_{R,1}$
$z^{-2(R-1)}$
z^{-6}
z^{-4}
z^{-2}
Output
$\left[1-z^{-1}\right]^2$
$G(z)$

Figure 3.21. Efficient structure for decimation.

- *Choice of design parameters*

The parameter K is closely related to the parameter n. By increasing either K or n, the stopband attenuation is enhanced. Nevertheless, it is preferable keeping K constant and as small as possible, whereas n is variable. Considering that K must be an even value, we set $K = 2$. As a consequence, the compensator parameter becomes $b = 2$ [15]. Furthermore, the slope σ controls the values of the ideal ACF that approximate the desired values necessary to compensate the passband droop introduced by the first-stage comb filter, $H_1(z)$. A simple way to assure multiplierless sharpening coefficients is by expressing the slope σ as $\sigma = 2^{-c}$. The constant c must be decreased as the droop introduced by $H_1(z)$ increases. Additionally, the tangency of the sharpening polynomial to the line with slope σ at the point (1, 1) is enhanced by increasing the parameter m. This results in a better passband characteristic but also in a higher complexity of the overall filter. Finally, the parameter L does not have implication in the improvement of the attenuation in the first folding band (where the worst-case attenuation occurs). However, L increases the droop of $H_1(z)$. For this reason, even though it is often considered arbitrary in most two-stage comb-based decimation filters, L should be kept as small as possible.

A simple design procedure for a given stopband specification is presented as follows:

1. Consider the decimation factor as $M = M_1M_2$, and that L and a residual decimation factor v are given. Set $K = 2$, $b = 2$, $\delta = 0$, $n = 0$, $c = 0$ and $m = 1$.

2. Increase n until the stopband requirement is satisfied.

3. Decrease c until an acceptable passband is obtained.

4. Increase m until the passband characteristic in step 2 can not be improved further.

Example 6

A design example to show the effectiveness of the proposal in comparison to other two-stage sharpening-based methods is presented below.

Consider a decimation process with overall decimation factor $D = M_1 M_2 \nu = 272$, with $M_1 = 4$, $M_2 = 17$ and $\nu = 4$. Assume that the passband edge frequency is $\omega_p = 0.9\pi/D$, and a desired stopband attenuation of 100 dB.

The polynomial used in this filter is $P_{\sigma,\delta,m,n}(x) = 5.125x^4 - 4.125x^5$, obtained with $m = 1$, $n = 3$, and $\sigma = -2^{-3}$. On the other hand, Stuart and Stephen use the traditional Kaiser and Hamming polynomial $P_{m,n}(x) = 3x^2 - 2x^3$, obtained with $m = 1$, $n = 1$, and their filter accomplishes the 100 dB attenuation with $K = 4$. Figure 3.22 shows the magnitude characteristics for both designs. Note that the proposed method achieves a much better passband characteristic.

For both designs, the first-stage comb filter can be decomposed in polyphase components, resulting in the same complexity. The second-stage comb filter of the proposed filter is implemented with the decimation architecture of Figure 3.21, whereas the one of [24] uses the structure of [25]. Note that the proposed filter has a lower computational complexity, as shown in Table 3.10.

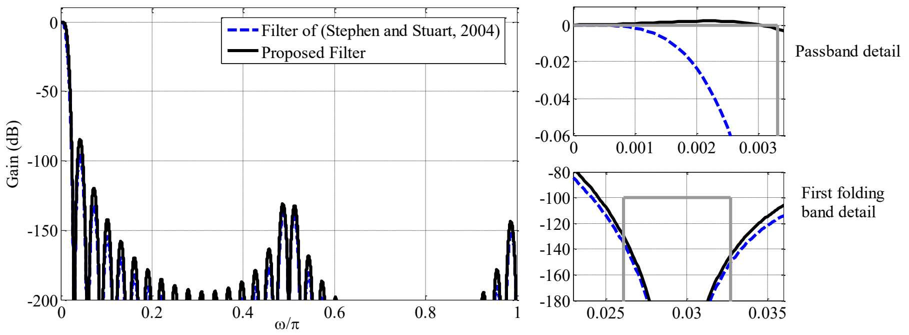


Figure 3.22. Gain in dB of the Example 6 applying the proposed method and the method of [24].

Table 3.10. Comparison of computational complexity of the sharpened filters in Examples 6.

Method	Additions Per Output Sample (APOS) in Example 6
Method [24]	$3KM_2+3K+3$ = 219
Proposed	$2RM_2+6R-1+$*coefficient adders* = 202

3.6 Sharpening of multistage comb decimator filter

A particular case of the above method, is the improvement of the comb decimators filters with decimation factor equal to power of two, i. e., M= 2^p. Namely, the use of p decimation stages. In this proposals to improve the worst case attenuation of the comb filter the improved sharpening is applied in last stage. In subsection 3.6.1 the filters of each stage are implemented in non-recursive form followed by a downsampler by 2. In order to improve both passband and stopband regions simultaneously it is convenient to apply the improved sharpening technique from [22]. Later, in subsection 3.6.2 an extension to the previous works a modification of the two-stage structure introduced in 3.6.1 is presented. The proposed scheme is a more regular CIC-based structure that provides also savings in chip area. A three-stage decimation structure for cases where M can be factorized in q = 3 arbitrary factors is proposed. The application of a compensator which works at the lower rate results in a passband improvement.

3.6.1 Sharpening of non-recursive comb decimation structure

We proposed to apply the improved sharpening described in Section 3.5, in last stage of the non-recursive structure,

$$H_{Sh}(z) = Sh\left\{[(1+z^{-1})/2]^2\right\}, \qquad (3.85)$$

where $Sh\{[(1+z^{-1})/2]^2\}$ denotes the improved sharpening to a filter (the cascade of 2 is chosen to avoid fractional delays and keeps the same to any value of cascades filter in all the stages).

Now, let us define L as the number of cascaded comb filters in the last stage as

$$L = 2N + l. \qquad (3.86)$$

where l has value 0 or 1. For l equal to 1, an additional comb filter is cascaded to the sharpened filter. This filter is shown in Figure 3.23 by the dashed box. As result, an odd number of cascaded filters is obtained.

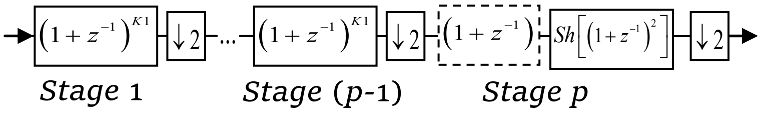


Figure 3.23. Proposed structure.

The number of cascaded comb filters in all stages, except in the last one, is K_1. Moreover, the number of extra comb filters that are cascaded in the last stage is $K_2 = L-K_1$.

The transfer function in the last stage becomes:

$$Sh\left\{[(1+z^{-1})/2]^2\right\} = \sum_{j=0}^{N} z^{-(N-j)} q_j [(1+z^{-1})/2]^{2j}, \tag{3.87}$$

$$q_j = \alpha_{j,0} - \sigma\alpha_{j,1} - \delta\alpha_{j,2}, \tag{3.88}$$

with $\alpha_{j,0}$, $\alpha_{j,1}$ and $\alpha_{j,2}$ given in (3.78).

In proposed structure the comb filter of the last stage is replaced by a filter with the following transfer function:

$$H_S(z) = \left(1 - l + l\cdot[(1+z^{-1})/2]\right)\cdot Sh\left\{[(1+z^{-1})/2]^2\right\}. \tag{3.89}$$

We write the transfer function of the proposed filter, at the input sampling rate as:

$$H_P(z) = \left[\prod_{i=0}^{p-2} 2^{-1}(1+z^{-2^i})\right]^{K1} H_S(z^{-2^{(p-1)}}). \tag{3.90}$$

Using multirate identity, some delays elements can be moved to lower rate. Figure 3.24 shows the obtained structure for this section by using (3.85), (3.86) and (3.87), where the dashed box indicates the case when the number of coefficients is odd, i.e. *N* is even, and the solid box indicates even coefficients, i.e. *N* is odd.

The total number of required APOS is given as:

$$APOS_P = 2\left[\left(2^p - 2\right)K_1 + (m+n+1)2 + l\right] + c \tag{3.91}$$

where *c* denotes the number of adders required for the multiplierless sharpening coefficients.

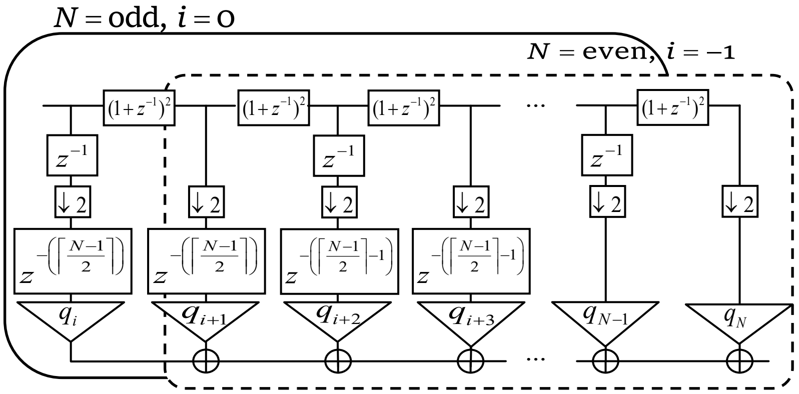

$N = \text{odd},\ i = 0$
$N = \text{even},\ i = -1$
$(1+z^{-1})^2$
$(1+z^{-1})^2$
$(1+z^{-1})^2$
$\cdots$
$(1+z^{-1})^2$
z^{-1}
z^{-1}
z^{-1}
$\downarrow 2$
$\downarrow 2$
$\downarrow 2$
$\downarrow 2$
$\downarrow 2$
$\downarrow 2$
$z^{-\left(\left\lceil \frac{N-1}{2} \right\rceil\right)}$
$z^{-\left(\left\lceil \frac{N-1}{2} \right\rceil\right)}$
$z^{-\left(\left\lceil \frac{N-1}{2} \right\rceil-1\right)}$
$z^{-\left(\left\lceil \frac{N-1}{2} \right\rceil-1\right)}$
q_i
q_{i+1}
q_{i+2}
q_{i+3}
q_{N-1}
q_N
$\cdots$

Figure 3.24. Structure of the sharpened section. Note that, if N is even, only the structure enclosed in the dashed box is used and in this case i = −1. If N is odd, the complete structure is used and i = 0.

- *Choice of the Design parameters*

The design parameters are:

1) The sharpening parameters σ, δ, m and n (see 3.5).

2) The value l.

3) The number of cascaded filters in all the stages except for the last one, K_1.

Choice of parameters n and δ

The attenuation in all odd folding bands depends on the last stage of the structure. Let us refer to the desired ACF in Figure 2, specifically to the desired line with slope δ. By setting δ = 0 we observe that, as the tangency n increases, the polynomial $Q_{\sigma,\delta,m,n}(x)$ becomes closer to the line. The amplitude values of the last stage filter that are near to zero are mapped to new amplitude values closer to zero in the sharpened version of this filter, and its attenuation becomes better. Thus, we set δ = 0 and consequently n must be increased to improve attenuation.

Choice of parameters m and σ

It is possible to take advantage of the slope parameter σ to obtain a passband compensation by filters in the last stage. This can be seen by observing the line with slope σ in Figure 3.19. If this slope is chosen to be negative, the amplitude values of the last stage filter that are close to and less than 1 are mapped to new amplitude values closer to and greater than 1 in the sharpened version of this filter. As a consequence, a passband compensation is obtained. The tangency of the sharpening polynomial to the line with slope σ at the point (1, 1) is enhanced by increasing the parameter m. This results in a better passband characteristic, but also in higher complexity of the overall filter. Consequently, we set $m = 1$. For higher passband droops absolute value of slope σ must be increased.

Choice of parameter l

When the desired attenuation can not be accomplished by a given polynomial degree N, the parameter l is set to 1 before increasing N. The extra filter adds a zero into the first folding band and the attenuation can be slightly increased.

Choice of parameter K_1

To obtain a value of number of APOS less than in the corresponding traditional non-recursive structure, with parameter K, the parameter K_1 must be less than K. The smaller the value of K_1, the smaller attenuation in the second folding band is achieved in the proposed filter.

By substituting $m = 1$ and $\delta = 0$ in (3.77) we have:

$$Q_{\sigma,\delta,m,n}(x)=\sum_{j=n+1}^{n+2}(\alpha_{j,0}-\sigma\alpha_{j,1})x^{j}=q_{N-1}x^{N-1}+q_{N}x^{N}, \tag{3.92}$$

where the coefficients q_{N-1} y q_N are obtained from 3.78 as,

$$q_{N-1}=n+2-\sigma, \tag{3.93}$$

$$q_{N}=-\left(n+1-\sigma\right). \tag{3.94}$$

To assure multiplierless coefficients in the improved sharpening polynomial, the slope σ is expressed as,

$$\sigma=2^{-B}\cdot round\left(\frac{\sigma_{prec\_inf}}{2^{-B}}\right), \tag{3.95}$$

where σ_{prec_inf} is an infinite-precision value and B is an arbitrary word-length for the fractional part of σ.

The Worst-Case Passband (WCP) in the magnitude response of a comb filter occurs at the frequency, [25]:

$$\omega_{p}=\frac{\pi}{MR}, \tag{3.96}$$

where R is the residual decimation factor. Similarly, the Worst-Case Attenuation (WCA) among the odd folding bands occurs in the first folding band at the frequency, [25]:

$$\omega_{s}=\frac{2\pi}{M}-\omega_{p}. \tag{3.97}$$

The WCA in the even folding bands occurs in the second folding band at the frequency:

$$\omega_{s}=\frac{4\pi}{M}-\omega_{p}. \tag{3.98}$$

To assure a WCA equal or higher than a desired minimum attenuation A (given in dB), the factor K can be calculated as:

$$K = \left\lceil \frac{-|A|}{20\log_{10}\{H(\omega)\}} \right\rceil_{\omega=\omega_s}, \tag{3.99}$$

where $\lceil x \rceil$ is the nearest integer equal or greater than x.

Figure 3.25 shows the WCAs in dB, for different values of K_1 and K_2, along with the value K of an original cascaded-by-K comb filter, when R = 2 and M = 2^4. From this diagram we can choose the parameters on design. Suppose that we want to design a decimation filter with a minimum WCA equal to –60dB. Then using (3.99) the parameter K of the comb filter must be K=6. From Figure 3.25 we can find the set of possible values K_1 and K_2 for which the proposed structure achieves a WCA of –60dB. These values are found as the intersections of the horizontal line of –60dB with the plots of Figure 3.25, and they are presented in Table 3.11.

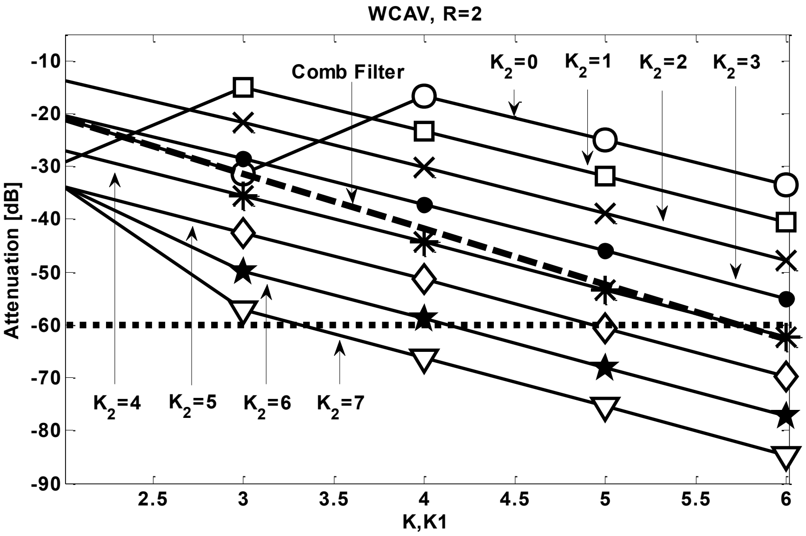


Figure 3.25. Worst case aliasing attenuation for comb filter and proposed filters.

Table 3.11. APOS for filters that accomplish WCA = –60 dB.

Structure	σ	l	WCP (dB)	APOS
Non- Recursive Comb				
(K=6)	-	-	-5.4318	180
Proposed				
(K1=4 y K2=7)	-3.6250	1	-0.5496	139
(K1=5 y K2=5)	-3	0	-0.4679	162
(K1=5 y K2=6)	-3.9375	1	-0.5842	166
(K1=5 y K2=7)	4	0	0.6668	167

- *Design Steps in the Proposed Method*

The residual decimation factor R and a desired WCA denoted as A are given. A simple design procedure is presented as follows:

5. Calculate an approximated value of K from (3.99) substituting ω_s from (3.97). Estimate also K_1 using (3.99), substituting ω_s from (3.98).

6. Set δ = 0, σ_{prec_inf} = 0, l = l and m = 1. Then estimate K_2 as $K_2 = K$ – K1 + l + 1 and obtain n = $\lceil (K_2 + K_1 - 4 - l)/2 \rceil$.

7. Compute the sharpening polynomial using (3.92)-(3.94) and form the transfer function $H_S(z)$ of (3.89).

8. Choose the value of B in (3.95). Obtain σ by decreasing σ_{prec_inf} using (3.95) until an acceptable passband is obtained.

9. If the desired attenuation is not achieved in the first folding band, increase n if l = 1 and reset l = 0, otherwise set l = 1, and repeat from step 3 until the WCA equal to A is accomplished in the first folding band.

Example 7

Consider a comb-based filter with the minimum attenuation given as $A = -80dB$ and $R=8$, with $M = 16$.

The resulting polynomial for this filter is $Q_{\sigma,\delta,m,n}(x) = 4x^2 - 3x^3$, where $m = 1$, $n = 1$, and $\sigma = -1$. Additionally, $l = 1$, $K_1 = 3$ and $K_2 = 4$. Figures 3.26 and 3.27 show the magnitude characteristics of the proposed design along with the solution of method [26], where $K_1 = 3$ and $K_2 = 1$. Note that the proposed method achieves a much better passband characteristic, with a slight increase of the computational complexity, as shown in Table 3.12.

Table 3.12. Comparison of computational complexity and magnitude characteristics for example 7.

Structure	APOS	WCA	WCP
Method, [26] (K1=3 and K2=1)	92	-89.2169	-0.2168
Proposed (K1=3 and K2=4)	100	-89.0144	-0.0044

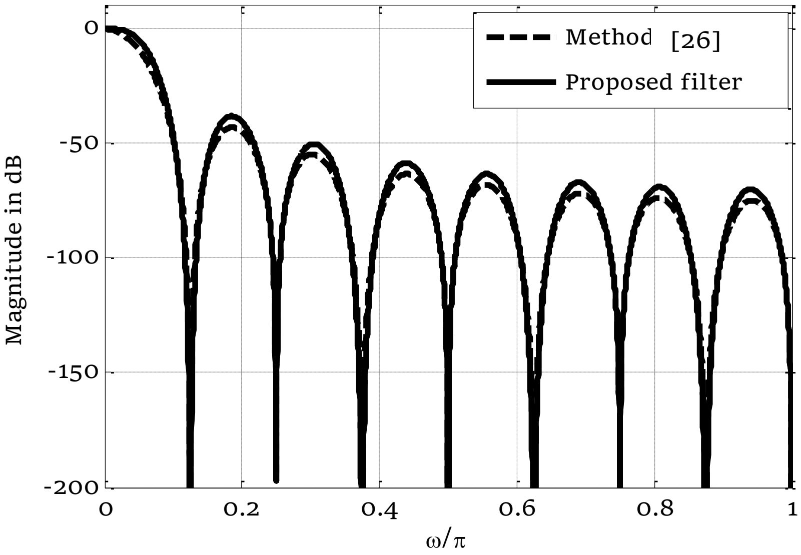


Figure 3.26. Magnitude responses in dB of filters in the Example 7.

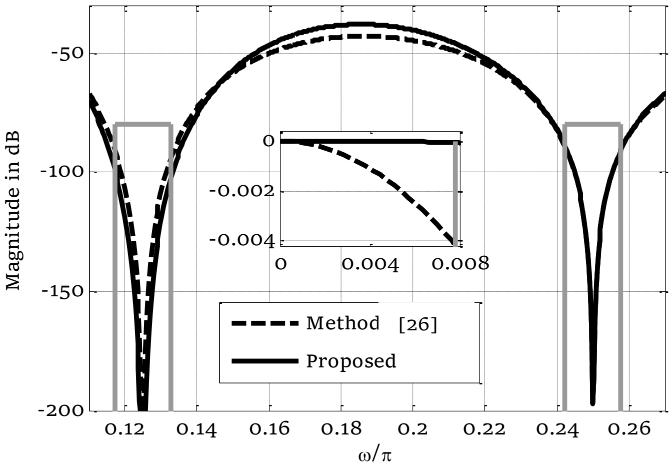


Figure 3.27. Detail of first and second folding bands with passband detail of the magnitude responses in dB of filters in the Example 7.

3.6.2 On compensated three-stages sharpened comb decimation filter

First, as started point consider the two-stage scheme, i.e., where $M = M_1 \times M_2$, with $\{M_1, M_2\} > 1$. The transfer function of the proposed decimation filter is

$$G(z) = H_1^{K_1}(z) \cdot H_2(z^{M_1}), \tag{3.100}$$

where

$$H_1(z) = H(z, M_1), \tag{3.101}$$

$$H_2(z) = z^{-NM_2} \cdot P_{\sigma,\delta,m,n}\left(z^{M_2-2}H^2(z, M_2)\right). \tag{3.102}$$

By substituting the following recursive form

$$H_{comb}(z) = \frac{1}{D} \cdot \frac{1 - z^{-D}}{1 - z^{-1}} = \frac{1}{D} \cdot \sum_{d=0}^{D-1} z^{-d}, \tag{3.103}$$

in (3.100) and (3.101) and using

$$\left|H_{comb}(e^{j\omega})\right| = \left|\frac{\sin(\omega D/2)}{D\sin(\omega/2)}\right| \tag{3.104}$$

, we have

$$\begin{aligned} G(z) = {} & \left(\frac{1-z^{-M_1}}{1-z^{-1}}\right)^{K_1} \cdot \Bigg[\delta \cdot z^{-(n+m)M_1M_2-2M_1} \times \\ & \left(\frac{1-z^{-M_1M_2}}{1-z^{-M_1}}\right)^{2} + \beta_{n+1} \cdot z^{-mM_1M_2-2(n+1)M_1} \times \\ & \left(\frac{1-z^{-M_1M_2}}{1-z^{-M_1}}\right)^{2(n+1)} + \ldots \\ & +\beta_{n+m+1} \cdot z^{-2(n+m+1)M_1} \left(\frac{1-z^{-M_1M_2}}{1-z^{-M_1}}\right)^{2(n+m+1)}\Bigg]. \end{aligned} \tag{3.105}$$

In order to map the amplitudes of the comb filter that are near to zero to values closer to zero after sharpening, the slope δ must be equal to zero. Thus, setting δ=0 in (3.104) and splitting the filter $H_1(z)$ in its integrator and comb parts, we obtain:

$$G(z, M_1, M_2) = H_I^{K_1}(z) \cdot H_C^{K_1}(z^{M_1}) \cdot G_S(z^{M_1}, M_2) \tag{3.106}$$

$$H_I(z) = \frac{1}{(1-z^{-1})}, \tag{3.107}$$

$$H_C(z) = (1-z^{-1}), \tag{3.108}$$

$$\begin{aligned} G_S(z, M_2) = {} & \beta_{n+1} \cdot z^{-mM_2} \cdot A^{(n+1)}(z) \cdot B^{(n+1)}(z^{M_2}) + \\ & \beta_{n+2} \cdot z^{-(m+1)M_2} A^{(n+2)}(z) \cdot B^{(n+2)}(z^{M_2}) + \ldots \\ & +\beta_{n+m+1} \cdot A^{(n+m+1)}(z) \cdot B^{(n+m+1)}(z^{M_2}) \end{aligned} \tag{3.109}$$

$$A(z) = \left(\frac{z^{-1}}{1 - z^{-1}} \right)^2 , \qquad (3.110)$$

$$B(z) = (1 - z^{-1})^2 . \qquad (3.111)$$

Splitting the downsampling M into to two factors M_1 and M_2, the filters $H_C{}^{K1}(z^{M1})$ and $G_S(z^{M1})$ can be moved after the downsampling by M_1, resulting in the structure shown in Figure 3.28.

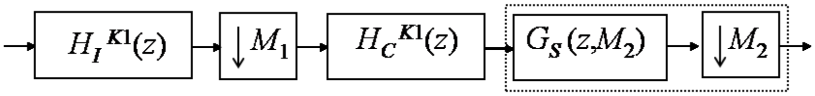


Figure 3.28. Two-stage decimation structure.

The efficient structure of the dashed block of Figure 3.32 is shown in Fig. 3.29.

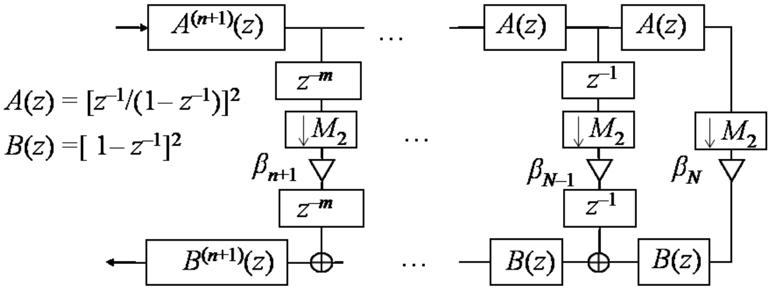


Figure 3.29. Efficient structure for the filter $G_s(z)$.

When the structure of Figure 3.29 is used in the dashed block in Figure 3.28, the filter $A^{(n+1)}(z)$ is cascaded with the filter $H_C{}^{K1}(z)$, forming an equivalent filter $D(z) = H_C{}^{K1}(z)A^{(n+1)}(z)$. From (3.108) and (3.110) we can see that this product results in an equivalent filter with transfer function:

$$D(z) = z^{-2(n+1)}[1/(1 - z^{-1})]^{2(n+1)-K1} = z^{-K1}A^{(n+1)-K1}(z). \qquad (3.112)$$

This structural modification allows us to save $2K_1$ adders compared to the original cascade $H_C^{K1}(z)A^{(n+1)}(z)$.

Finally, replacing the structure of Figure 3.29 in its corresponding equivalent dashed block of Figure 3.28, we arrive to the proposed two-stage structure presented in Figure 3.30. The corresponding coefficients β_i are obtained from (3.88) being q_j equal β_i and N from $n + m + 1$, whereas $H_I(z)$, $A(z)$ and $B(z)$ are respectively given in (3.107), (3.110) and (3.111).

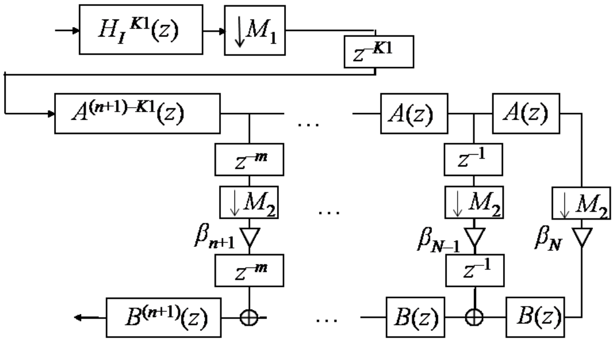


Figure 3.30. Proposed two-stage structure.

We consider here that the decimation factor M can be written as :

$$M= M_1M_2M_3. \tag{3.113}$$

The transfer function of the proposed decimation filter is given as

$$G_p(z) = H_1^{K_1}(z) \cdot H_2^{K_2}(z^{M_1}) \cdot H_3(z^{M_1M_2}) , \tag{3.114}$$

where $H_1^{K1}(z)$ is given as (3.101), and with

$$H_2(z) = H(z, M_2) , \tag{3.115}$$

$$H_3(z) = z^{-NM_3} \cdot P_{\sigma,\delta,m,n}\left(z^{M_3-2}[H^2(z,M_3)C(z^{M_3})]\right)., \tag{3.116}$$

where C(z) is the comb compensator proposed in [9].

The number of cascaded filters K_1 and K_2 can be chosen with different values. Using the form of (3.106) and setting δ=0 we arrive to the proposed transfer function,

$$\begin{aligned} G_p(z,M_1,M_2,M_3) = H_1^{K_1}(z)\cdot H_I^{K_2}(z^{M_1})\cdot H_C^{K_2}(z^{M_1M_2})\cdot \\ G_{pS}(z^{M_1M_2},M_3)\cdot C(z^M) \end{aligned} \tag{3.117}$$

where $H_I(z)$ and $H_C(z)$ are given in (3.107) and (3.108). Similarly, $G_{pS}(z, M_3)$ is expressed as,

$$\begin{aligned} G_{pS}(z,M_3) = \beta_{n+1}\cdot z^{-mM_3}\cdot A^{(n+1)}(z)\cdot \\ B^{(n+1)}(z^{M_3})\cdot C(z^M)+\beta_{n+2}\cdot z^{-(m+1)M_3}\cdot \\ A^{(n+2)}(z)\cdot B^{(n+2)}(z^{M_3})\cdot C(z^M)+\ldots \\ +\beta_{n+m+1}\cdot A^{(n+m+1)}(z)\cdot B^{(n+m+1)}(z^{M_3})\cdot C(z^M), \end{aligned} \tag{3.118}$$

where $A(z)$ and $B(z)$ are given in (3.110) and (3.111).

The filter $H_1^{K1}(z)$ is implemented in nonrecursive form. The polyphase decomposition can be applied to this stage. The filter $H_I^{K2}(z^{M1})$ can be moved after the downsampling by M_1 and the filters $H_C^{K2}(z^{M1M2})$ and $G_{pS}(z^{M1M2}, M_3)$ can be moved after the downsampling by M_2. Applying the compensator filter of [9] in the last stage, the resulting structure is given in Figure 3.31.

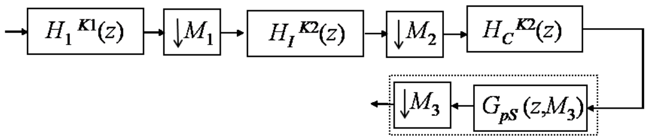


Figure 3.31. Proposed decimation structure with a CIC scheme used for H_1^{K1}(z) and $H_2^{K2}(z^{M1})$.

The dashed section of Figure 3.31 is implemented in a similar way as that of Figure 3.29, just replacing M_2 by M_3. In the same way, an equivalent filter $D_1(z)$ = $H_C^{K2}(z)A^{(n+1)}(z)$ is obtained. Using (3.112) we get:

$$D_1(z) = z^{-K2}A^{(n+1)-K2}(z). \qquad (3.119)$$

Finally, the resulting structure, obtained by replacing $H_C^{K2}(z)A^{(n+1)}(z)$ with $D_1(z)$, and using the non recursive form of $H_1^{K1}(z)$, is given in Figure 3.32.

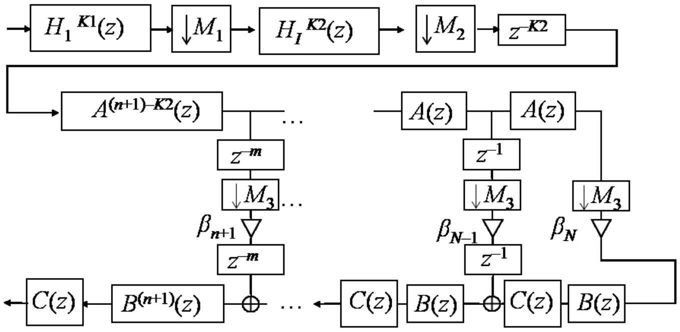


Figure 3.32. Proposed structure with all the filters working at low rate (the first nonrecursive comb filter is implemented in polyphase decomposition).

The filters $A(z)$ and $B(z)$ can be implemented with two adders and two delays. Thus, the proposed structure requires an amount of Additions per Output sample (APOS) given by:

$$\begin{aligned} N_{APOS} &= 2M_2M_3(M_1-1) + S(H_{1i}) + K_1M_1M_2 \\ &+2(N-K_1)M_1 + 2N + m + \sum\nolimits_{i=n+1}^{N} S(\beta_i) + S(C_i), \end{aligned} \qquad (3.120)$$

where $S(\beta_i)$ means the number of adders required to implement the coefficient β_i, $S(C_i)$ means the number of adders required to implement the coefficient of the compensator, and $S(H_{1i})$ means the number of adders required to implement the coefficient of the filter $H_1^{K1}(z)$.

The design steps of the proposed filter are:

1. Consider the decimation factor M expressed as (3.113). Choose $M_2 \geq M_3 \geq M_1$.

2. Set $K_1 = K_2 = 1$, $K_3 = 2$, $\delta = 0$, $n = 1$ and $m = 1$.

3. Design the compensator of [9] such that the passband deviation is as low as possible but preserving a monotonic passband characteristic.

4. Obtain $\sigma = 2^{-B}[\mathrm{round}(\sigma_{inf}/2^{-B})]$, where σ_{inf} is a positive slope if the passband characteristic is monotonically increasing or a negative slope if the passband characteristic is monotonically decreasing. Increase the absolute value of σ_{inf} proportionally to the passband deviation until the passband improvement is appropriate. Choose B as small as possible (it is usual to have $B < 6$).

5. Compute the sharpening polynomial given in (3.77) and design the filter $G_p(z)$ of (3.117).

6. If the attenuation in the first folding band is not satisfied, then increase n, K_1, K_2 and repeat the procedure until the desired attenuation is obtained.

Example 8

Consider the decimation process with residual factor equal to $v = 2$ and decimation factor $M = 81$. The minimum attenuation of 80 dB in the first folding band is required.

The decimation factor $M = 81 = 3^4 = 9 \cdot 3 \cdot 3$. We choose $M_1 = 3$, $M_2 = 9$, $M_3 = 3$.

The obtained sharpening polynomial is $P_{\sigma,\delta,m,n}(x)$=3.3750x^3 - 2.3750x^4. The parameters are: n = 2, K_1= 2, K_2= 4 and σ = 0.6250. The resulting compensator is given as $C(z^M)$= 2^{-2}[-1/2 + 5z^{-M} - 1/2z^{-2M}].

Figure 3.33 shows the magnitude response of the proposed filter along with the response of method [26]. The response of that filter is obtained using parameters K_1=4, K_2=4, K_3=4 and K_4=5 (it uses 4 stages) and it is shown with dashed line.

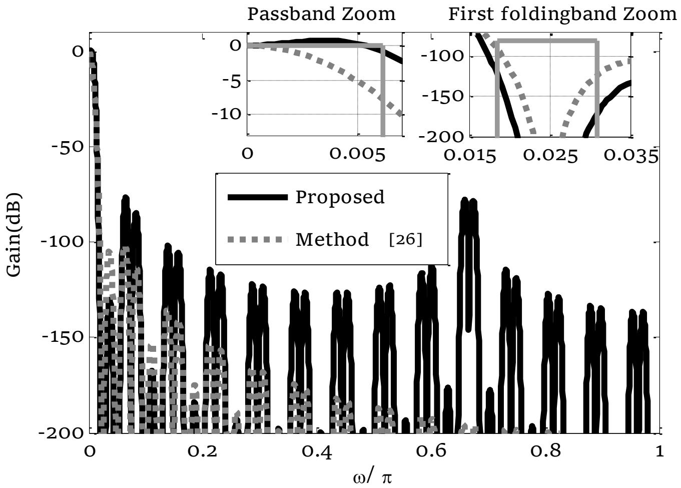


Figure 3.33. Magnitude response of the filter of Example 8. The resulting magnitude response by using the proposed design and the design by method [26].

Table 3.13. Comparison of characteristics and computational complexity for example 8.

Method	Worst case attenuation value	Worst case passband droop	Additions per output sample
Method [26]	-90.45	-7.8159	1224
Proposed	-122	-0.9048	235

3.7 References

[1] Oppenheim, A. V., and Schafer, R. W. *Discrete-Time Signal Processing*, N J:Prentice-Hall International, 1989.

[2] Aksoy, L., Flores, P., and Monteiro, J. "A tutorial on multiplierless design of FIR filters: algorithms and architectures," *Circ. Syst. Signal Process.* 2014.

[3] Coleman, J. O. "Chebyshev stopbands for CIC decimation filters and CIC-implemented array tapers in 1D and 2D," *IEEE Trans. on Circ. and Syst.-I*, vol. 59, no. 12, pp. 2956-2968, 2012.

[4] Rayes, M. O., Trevisan, V., and Wang, P. S. "Factorization properties of Chebyshev polynomials," *Computers and mathematics with applications*, no. 50, pp. 1231-1240, 2005.

[5] Dolecek, G. J., and Dolecek, V. "Application of Rouche's theorem for MP filter design," *Applied Mathematics and Computation*, no. 211, pp. 329-335, 2009.

[6] Kale, I., Cauin, G.D., and Morling, R.C.S. "Minimum-phase filter design from linear-phase start point via balanced model truncation," *IET Electronic Letters*, vol. 31, no. 20, pp. 1728-1729, 1995.

[7] Dam, H. H., Nordebo, S., and Svensson, L. "Design of minimum-phase digital filters as the sum of two allpass functions using the cepstrum technique," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 726-731, 2003.

[8] Pei, S.-C., and Lin, H.-S. "Minimum-phase FIR filter design using real cepstrum," *IEEE Trans. Circ. and Syst.-II*, vol. 53, no. 10, pp. 1113-1117, 2006.

[9] Romero D. E. T., and Dolecek, G. J. “Application of amplitude transformation for compensation of comb decimation filters,” *Electronics Letters*, vol. 49, no. 16, 2013.

[10] Lyons, R. “Sample Rate Conversion,” in *Understanding Digital Signal Processing*, 2nd ed. New Jersey, USA, Prentice Hall, 2004.

[11] Dolecek, G. J., and Mitra, S. K. “Simple method for compensation of CIC decimation filter,” *Electronics Letters*, vol. 44, no. 19, pp. 1162–1163, 2008.

[12] Pecotic, M. G., Molnar G. , and Vucic, M. “Design of CIC compensators with SPT coefficients based on interval analysis,” in *Proc. The 35th IEEE Int. Convention MIPRO 2012*, pp. 123–128, 2012.

[13] Kaiser, J., and Hamming, R. “Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter,” *IEEE Trans. Acoust. Speech and Signal Process.*, vol. 25, no. 5, pp. 415-422, 1977.

[14] Jiang, Z., and Wilson, A. N. “Efficient digital filtering architectures using Pipelining/Interleaving,” *IEEE Transactions on Circuits and Systems- II: Analog and Digital Signal Processing*, vol. 44, no. 2, pp. 110-119, 1997.

[15] Dolecek, G. J., and Mitra, S. K. “Novel two-stage comb decimator,” *Computación y Sistemas*, vol. 16, no. 4, pp. 481-489, 2012.

[16] Dolecek, G. J. “Simple wideband CIC compensator,” *Electronics Letters*, vol. 45, no. 24, pp. 1270–1272, 2009.

[17] Dolecek G. J., and Dolecek, L. “Novel multiplierless wide-band CIC compensator,” in *Proc. IEEE ISCAS 2010*, pp. 2119–2122, 2010.

[18] Milic, D. J., and Pavlovic, V. D. "A new class of low complexity low-pass multiplierless linear-phase special CIC FIR filters," *IEEE Signal Processing Letter*, vol. 21, no.12, pp. 1511-1515, 2014.

[19] Fa-Long, L. (Editor), *Digital Front-End in Wireless Communications and Broadcasting: Circuits and Signal Processing*, Cambridge University Press, New York, USA, 2011.

[20] Meyer-Baese, U. "Chapter 2: Computer Arithmetic," in *Digital Signal Proccessing with Field Programmable Gate Arrays*, Springer, 4th Edition, pp. 142, 2014.

[21] Stosic, B. P., and Pavlovic, V. D. "Design of new selective CIC filter functions with passband-droop compensation," *Electronics Letters*, vol. 52, no. 2, pp. 115-117, 2016.

[22] Hartnett, R. J., and Boudreaux-Bartels, G. F. "Improved filter sharpening," *IEEE Trans. on Signal Process*, vol. 43, no. 12, pp. 2805-2810, 1995.

[23] Samadi, S. "Explicit formula for improved filter sharpening polynomial," *IEEE Trans. on Signal Process*, vol. *9*, pp. 2957–2959, 2000.

[24] Stephen, G., and Stuart, R. "High-speed sharpening of decimating CIC filter," *Electronics Letters*, vol. 40, pp.1383-1384, 2004.

[25] Kwentus, A., Jiang, Z., and Willson, N. "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Procesing*, 45, pp. 457-467, 1997.

[26] Dolecek, G. J., and Molina, G. "Low-power non-recursive comb-based decimation filter design*," in Proc. Int. Symp. on Communications, Control and Signal Process. ISCCSP 2012*, pp. 1-4, 2012.

Chapter 4

Theoretical lower bounds for parallel pipelined shift-and-add constant multiplications

Multiplication with constants is a regular operation in Digital Signal Processing (DSP) systems. In hardware, a multiplication is demanding in terms of area and power consumption. However, the Single Constant Multiplication (SCM) and Multiple Constant Multiplication (MCM) operations can be implemented by using only shifts, additions and subtractions, with the last two being usually referred in general form as additions [1]-[36].

Theoretical lower bounds for the number of adders and for the number of depth levels, i.e., the maximum number of serially connected adders (also known as the critical path), in SCM, MCM and other constant multiplication blocks that are constructed with two-input adders under the shift-and-add scheme have been presented in [3]. Tighter lower bounds, as well as a new bound, namely, the one for the number of extra adders required to preserve the lowest number of depth levels, were presented in [4] for the SCM case. Nevertheless, there are no theoretical lower bounds for the case of constant multiplication blocks that include multiple-input additions/subtractions and pipeline registers in the involved arithmetic operations. This type of operations has become very important mainly when the pipelined constant multiplication blocks are implemented in the increasingly demanded Field Programmable Gate Array (FPGA) platforms. This is due to the fact that logic blocks of FPGAs include memory elements, and thus pipelining results in low extra cost [5]-[12]. Currently, the use of three-input adders has started to gain importance, since the logic blocks of the newest families of FPGAs are bigger and allow to fit more complex adders using nearly the same amount of hardware resources [10]-[12].

Particularly, in the last two decades many efficient high-level synthesis algorithms have been introduced for the multiplierless design of constant multiplication blocks. The common cost function to be minimized in these algorithms is given by the number of arithmetic operations (additions and subtractions) needed to implement the multiplications. Nevertheless, the critical path has the main negative impact in the speed and power consumption [13]-[18]. Therefore, substantial research activity has been carried out currently targeting both, Application-Specific Integrated Circuits (ASICs) [19]-[21] and FPGAs [5]-[10], [22]-[25], where the minimization of the number of arithmetic operations subject to a minimum number of depth levels is the ultimate goal.

This chapter introduces the theoretical lower bounds for the number of operations necessary to implement Pipelined Single Constant Multiplication (PSCM) and Pipelined Multiple Constant Multiplication (PMCM) blocks that are constructed with the shift-and-add scheme. For the derivation of these bounds we consider that either an n-input (where n is an integer) pipelined addition/subtraction or a single pipeline register have the same cost. As mentioned earlier, recently this assumption fits particularly well for cases where n is set equal to 3 and the target platforms for implementation are the newest FPGAs from the two most dominant manufacturers, Xilinx and Altera. However, it is worth highlighting that n = 2 is still under common use in many applications. This contribution is important because the optimality of different algorithms that reduce the number of operations in PSCM and PMCM blocks can be tested using appropriate theoretical lower bounds. Additionally, these bounds can be useful to develop new algorithms.

This chapter is organized as follows. In the next section, definitions and methods needed to address the proposal are given. Section 4.2 presents the new theoretical lower bounds along with theorems and proofs to support the derivation of these bounds. Comparisons with previous theoretical lower bounds from [3] and [4] are provided in Section 4.3. Finally, conclusions are given in Section 4.4.

4.1 Definitions

Let us express the n-input A-operation, i.e., the n-operand addition/subtraction along with shifts, as follows,

$$A_q(u_1,\ldots,u_n) = \left| 2^{l_1} u_1 + \sum_{i=2}^{n} (-1)^{s_i} 2^{l_i} u_i \right| 2^{-r}\ , \tag{4.1}$$

where $l_i \geq 0$ for $i = 1, \ldots, n$ are left shifts, $r \geq 0$ is a right shift, $s_2, \ldots, s_n$ are binary values, $q = \{l_1, \ldots, l_n, s_2, \ldots, s_n, r\}$ is the configuration of the A-operation and $u_1,\ldots, u_n$ are odd integers.

It is important to mention that a *multiplicative graph* is the graph obtained by cascading subgraphs, and the union point between two cascaded subgraphs in a multiplicative graph is called *articulation point* [33]. This is illustrated in Figure 4.1(a) for n-input A-operations. A particular case is the *completely multiplicative graph*, where each cascaded subgraph is composed by one *A*-operation, as shown in Figure 4.1(b). Other graphs without articulation points are referred as *non-multiplicative graphs* [33]. A cascaded interconnection of a completely multiplicative graph with a non-multiplicative graph is called *generalized graph*, see Figure 4.1(c).

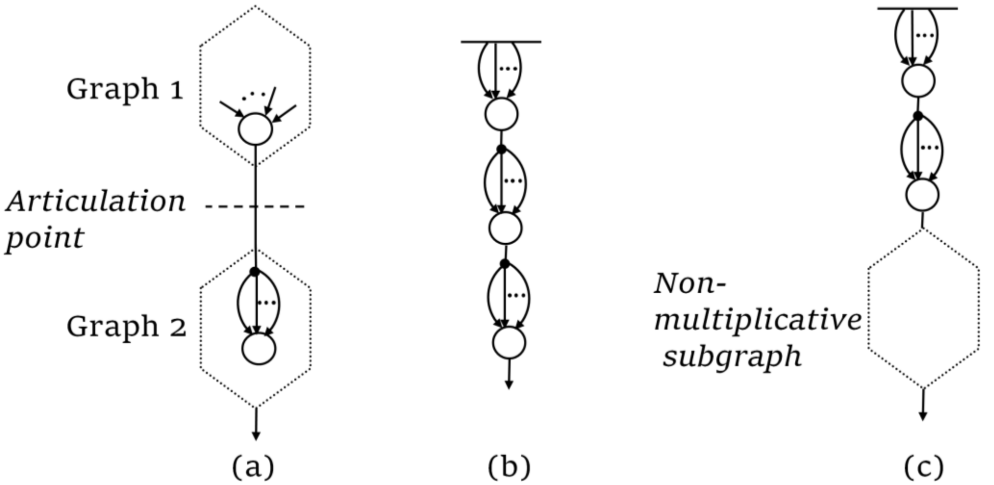


Figure 4.1. (a) multiplicative graph, (b) completely multiplicative graph, and (c) generalized graph.

The speed of a design is restricted by the critical path. The pipelining technique allows the reduction of a critical path introducing registers along the data path [34]. In FPGA implementations the constant multiplications involving shifts-and-add operations can be made fully-pipelined with a low extra cost. Pipelining has a small overhead due to the fact that the logic blocks in FPGAs include memory elements, which are otherwise unused [28], [35]-[36]. For example, Table 4.1 shows the amount of logic elements used to implement the multiplier 45*X* (for an 8-bit input) in an Altera Cyclone IV EP4CE115F29C7 FPGA. We observe that only 3 extra logic elements are needed in the pipelined implementation, which represents an increase of 9.7% in resources utilization compared with the non-pipelined case. Nevertheless, the frequency of operation is increased by 31.7%.

Table 4.1. Pipelined and Non Pipelined implementations of a 45*X* multiplier.

Pipelined	Total logic elements (LE)	Maximum frequency of operation (MHz)
No	31	285.47
Yes	34	376.08

Due to the aforementioned observation, the implementation cost will be accounted by the number of *registered operations*, called hereafter *R*-operations, where an *R*-operation is either an *A*-operation plus a register (an addition-register pair) or a single register. Two *R*-operations with the same cost are illustrated in a simplified way in Figure 4.2. Hence, the PSCM problem consists in finding the pipelined array of *A*-operations that form a single-constant multiplier using the minimum number of *R*-operations. Similarly, the PMCM problem consists in finding the pipelined array of *A*-operations that form a multiple-constant multiplier using the minimum number of *R*-operations.

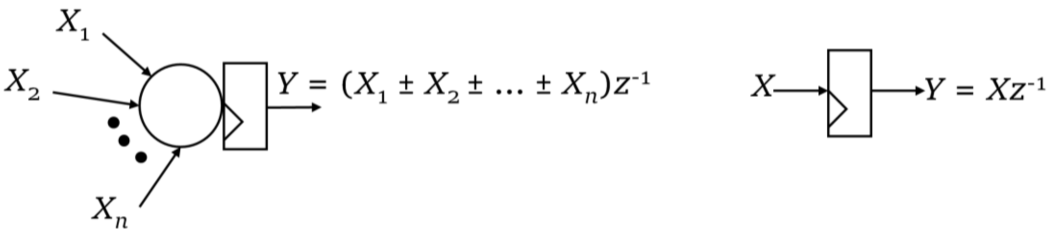


Figure 4.2. *R*-operations with the same cost.

To calculate the lower bounds for the number of R-operations required to implement PSCM and PMCM blocks, we need the following information from a constant:

1) Its Minimum Number of Signed Digits (MNSD), denoted by S. We will also refer to this number in a more informal manner as "the number of non-zero digits".

2) Its number of prime factors (it does no matter if these prime factors are repeated). This number is denoted by Ω.

4.2 Proposed lower bounds

In the following we state, in sub-section 4.2.1, Theorems 1 to 8 to derive the lower bounds of R-operations in PSCM, and in sub-section 4.2.2 Theorems 9 and 10 for PMCM, along with their corresponding proofs. The pipelining operation, which has not been alluded in the previous works [3] and [4], is explicitly included in the proposed lower bounds with the R-operations.

4.2.1 PSCM case

Whenever a constant c is mentioned in the theorems of this sub-section (Theorem 1 to 8), we consider that the MNSD of that constant is S and its number of prime factors is Ω.

Theorem 1 provides the upper limit of non-zero digits that can be generated by any graph with a given number of depth levels, regardless of its number of R operations. From this, we can know the minimum number of depth levels that a graph must have to implement a constant with a given S.

Theorems 2 and 3 prove the properties of the completely multiplicative graphs, namely, generating the upper limit of non-zero digits mentioned in Theorem 1 with the minimum possible number of R operations. From them, we have that the completely multiplicative graph is a solution with the lower bound for the number of R operations. However, as it is known, this graph has articulation points, and every articulation point represents the union between two cascaded subgraphs, i.e., the product of two smaller constants. Therefore, Theorem 4 uses Ω to identify what constants can be implemented with the completely multiplicative graph (for example, prime constants can not be factorized into smaller constants, thus they can not be implemented by a completely multiplicative graph).

Theorem 5 identifies the minimum number of R operations needed in any non-multiplicative graph with a given number of depth levels, and Theorem 6 proves that non-multiplicative graphs can generate the upper limit of non-zero digits mentioned in Theorem 1 with its minimum number of R operations. Then, Theorem 7 establish the lower bound for the number of R operations needed to implement a prime constant ($\Omega = 1$).

Finally, Theorem 8 completes the information of Theorems 4 and 7, namely, the lower bound of R operations needed to implement non-prime constants that have fewer number of factors than the number of sub-graphs used in a completely multiplicative graph.

Theorem 1. *A graph with p depth levels can provide at most n^p non-zero digits for a constant.*

Proof. The proof is given by induction (see proof of Theorem 6.9 in [35] for the case of 2-input A-operations):

1) The base case corresponds to the first depth level, where a n-input A-operation can form a constant with at most n non-zero digits. This is true since the input of any graph has one non-zero digit [3]-[4], [35].

2) As inductive step we assume that, in the p-th level, there are n^p non-zero digits at most. In the (p+1)-th level an A-operation can form a constant whose number of non-zero digits is the sum of the numbers of non-zero digits at every input of that A-operation. This is at most n times the maximum number of non-zero digits available in the previous level, i.e., $n \times n^p = n^{p+1}$ non-zero digits.

Since assuming that the theorem is true for p implies that the theorem is also true for p+1, and since the base case is also true, the proof is complete. The aforementioned observations are presented graphically in Figure 4.3. Note that an adder, regardless of its number of inputs, can not generate more non-zero digits than the sum of the numbers of non-zero digits in every one of its inputs. Thus, the MNSD can be, at most, n-plicate if the inputs of the n-input adder placed in any depth level come from the immediately previous depth level. ■

Theorem 2. *A completely multiplicative graph with p A-operations can generate n^p non-zero digits.*

Proof. This proof is an straightforward extension of the proof of Theorem 6.8 in [35], which corresponds to completely multiplicative graphs with 2-input A-operations. As stated earlier, the input of a graph has one non-zero digit. In the completely multiplicative graph, there are at most n non-zero digits after the A-operation placed at the 1st depth level. Cascading an A-operation to that output yields at most $n \times n$ non-zero digits, and so on. The number of non-zero digits at the depth level p is at most the n-tuple of the number of non-zero digits of a fundamental at the (p–1)-th depth level. Consequently, the maximum number of non-zero digits at the p-th depth level is n^p. Figure 4.4 illustrates an example. ■

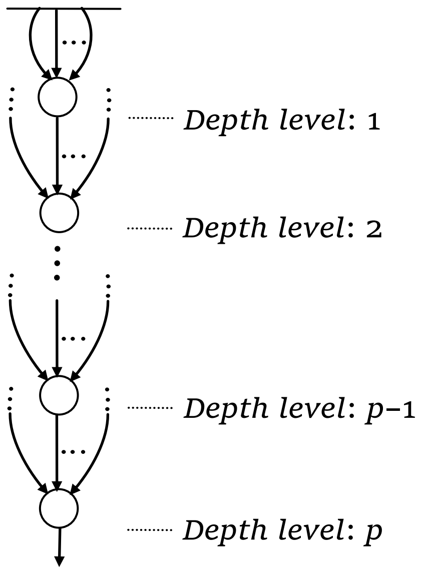


Figure 4.3. In the p-th depth level, a graph can not generate more than n^p non-zero digits.

Theorem 3. *A completely multiplicative graph with p depth levels needs only p R-operations.*

Proof. The completely multiplicative graph with p depth levels has p A-operations, and every A-operation forms a subgraph. Pipelining between two subgraphs needs only one register, according to [34], because the pipelining occurs on the articulation point. This results in every A-operation being followed by a register. Since an A-operation followed by a register is considered an R-operation, there are only p R-operations in total. This is illustrated in Figure 4.5. ■

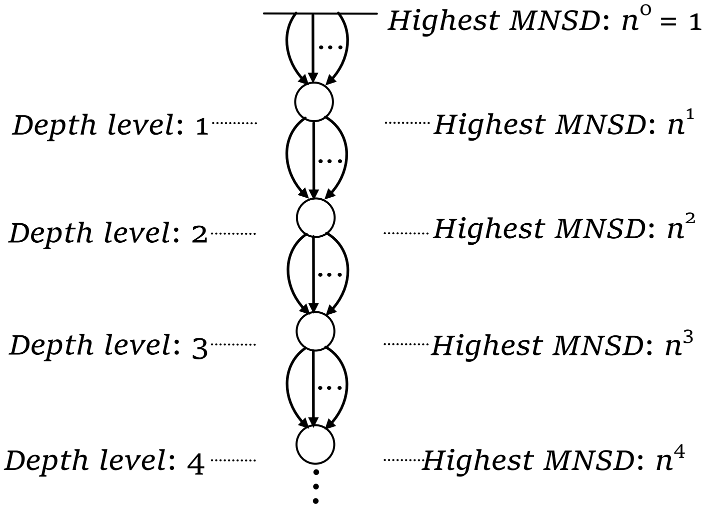


Figure 4.4. The completely multiplicative graph achieves n^p non-zero digits with the minimum number of n-input adders, p, and the minimum number of depth levels, p.

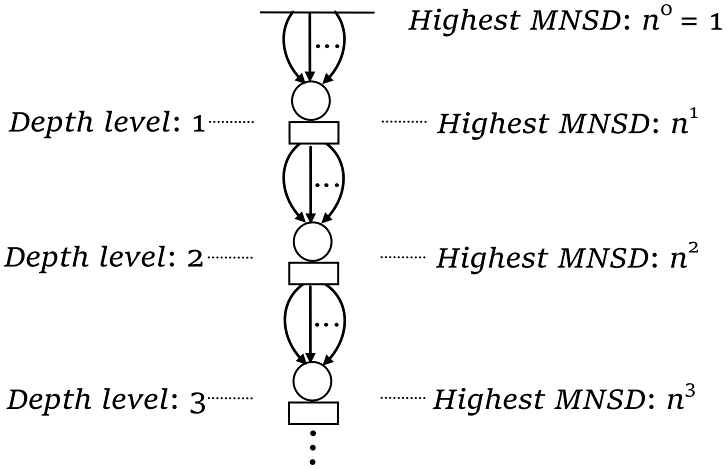


Figure 4.5. The pipelined completely multiplicative graph achieves n^p non-zero digits with the minimum number of n-input R-operations, p, and the minimum number of depth levels, p.

Theorem 4. *A constant with* $(n^{p-1}+1) \leq S \leq n^p$ *and* $\Omega \geq p$ *needs at least p R-operations.*

Proof. From Theorem 2 we have that a constant with $(n^{p-1}+1) \leq S \leq n^p$ non-zero digits can be implemented with at least p depth levels, which implies at least p A-operations. From Theorem 3 we have that a completely multiplicative graph can generate those values for S with only p R-operations. The completely multiplicative graph with p R-operations consists of p cascaded subgraphs, thus a constant implemented with that graph must have at least p prime factors. Since $\Omega \geq p$ holds, the completely multiplicative graph can be employed to implement that constant using p R-operations. ■

Theorem 5. *A non-multiplicative graph with p depth levels needs at least (2p – 1) R-operations.*

Proof. According to Theorem 3, if a graph with p depth levels has only p R-operations in total, it must be a pipelined completely multiplicative graph. According to Theorem 2, that graph can generate the maximum possible number of non-zero digits, namely, n^p. To make non-multiplicative that optimal graph, the $(p - 1)$ articulation points must be eliminated. From [34], it is known that at least one additional R-operation must be added for every eliminated articulation point. Therefore, at least $(2p - 1)$ R-operations are required, i.e., the original p minimum number of R-operations in the form of addition-delay pairs plus the additional $(p - 1)$ R-operations in the form of pure delays. Figure 4.6 shows an example with $p = 3$. ■

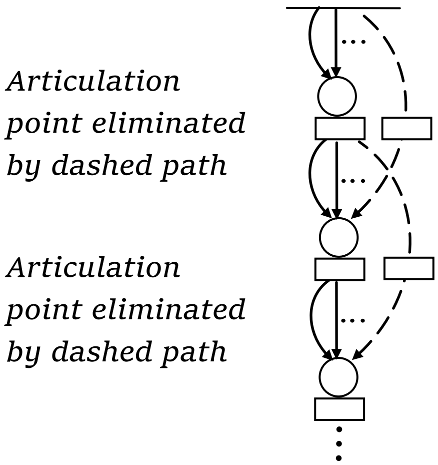

Articulation point eliminated by dashed path
Articulation point eliminated by dashed path

Figure 4.6. Non-multiplicative graph with p = 3 depth levels and p–1 extra R-operations in the form of pure delay.

Theorem 6. *A non-multiplicative graph with p depth levels and (2p – 1) R-operations can generate n^p non-zero digits.*

Proof. Consider a graph with p depth levels formed by two completely multiplicative graphs of (p–1) levels each, connected in parallel from the input of the graph, and one A-operation placed in the p-th level summing up the outputs of the aforementioned graphs. The output of one of these graphs is connected to the n – 1 inputs of the last A-operation and the output of the other graph is connected to the remaining input of the last A-operation. This is a non-multiplicative graph because it is not formed by cascading subgraphs, and it is composed by (2p –1) A-operations. According to Theorem 2 we can obtain n^{p-1} non-zero digits from the completely multiplicative graphs and according to Theorem 3 these graphs can be pipelined without requiring extra registers. Since the last A-operation can add n times the n^{p-1} non-zero digits in each one of its inputs and can be pipelined without extra cost, the resulting graph generates n^p non-zero digits using (2p – 1) R-operations. An example of this is shown in Figure 4.7. ∎

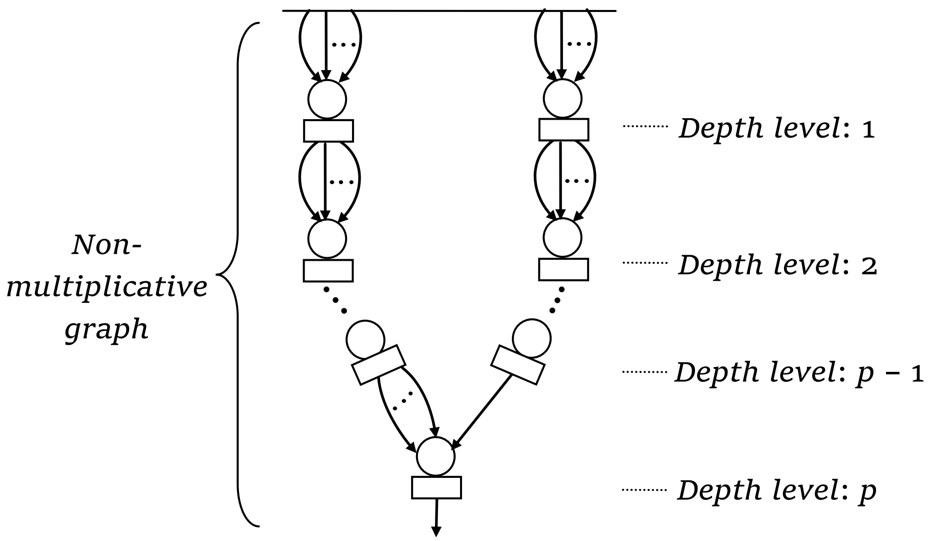


Figure 4.7. Non-multiplicative graph that generates the maximum number of non-zero digits, n^p, with the minimum number of R-operations in non-multiplicative graphs.

Theorem 7. *A constant with* $(n^{p-1}+1) \leq S \leq n^p$ *and* $\Omega = 1$ *needs at least* $2p - 1$ *R-operations.*

Proof. Since $\Omega = 1$ holds, the non-multiplicative graph must be employed to implement that constant. From Theorem 6 we have that a constant with $(n^{p-1}+1) \leq S \leq n^p$ non-zero digits can be implemented with at least p depth levels and at least $2p - 1$ R-operations. This is a lower bound for the number of R-operations, since from Theorem 5 we have that a non-multiplicative graph with p-levels needs at least $2p - 1$ R-operations. ■

Theorem 8. *A constant with* $(n^{p-1}+1) \leq S \leq n^p$ *and* $1 < \Omega < p$ *needs at least* $(2p - \Omega)$ *R-operations.*

Proof. From Theorem 1 we have that p depth levels are necessary to achieve the values of S in the specified range. Since $\Omega < p$ holds, we can take advantage of a completely multiplicative graph with $\Omega-1$ R-

operations at most, which, according to Theorem 2, generates $n^{\Omega-1}$ non-zero digits at most, and represents the product of Ω−1 factors. The last factor can be formed with a non-multiplicative subgraph with $[p-(\Omega-1)]$ depth levels. According to Theorem 5, this subgraph needs at least $2[p-(\Omega-1)] - 1$ *R*-operations, and according to Theorem 6 it can generate $n^{[p-(\Omega-1)]}$ non-zero digits. The total graph, illustrated in Figure 4.8, can generate at most $n^{\Omega-1}\times n^{[p-(\Omega-1)]} = n^p$ non-zero digits and uses at least $(\Omega-1) + 2[p-(\Omega-1)] - 1 = 2p -2(\Omega-1) + (\Omega-1) - 1 = 2p -(\Omega-1) - 1 = (2p - \Omega)$ *R*-operations. ■

Finally, from Theorem 1 we have that the number of depth levels necessary to achieve *S* is $p = \lceil \log_n(S) \rceil$. Substituting this value for *p* and using Theorems 4, 7 and 8, we obtain the lower bound for the number of *R*-operations needed to form a PSCM block as follows,

$$L_{PSCM} = \begin{cases} 2\lceil \log_n(S) \rceil - \Omega; & \Omega < \lceil \log_n(S) \rceil, \\ \lceil \log_n(S) \rceil; & \Omega \geq \lceil \log_n(S) \rceil. \end{cases} \tag{4.2}$$

4.2.2 PMCM case

The theorems in this section are stated for *N* constants $c_1, c_2, \ldots, c_N$, whose respective MNSDs are $S_1, S_2, \ldots, S_N$, and their respective numbers of prime factors are $\Omega_1, \Omega_2, \ldots, \Omega_N$, such that $S_1 \leq S_2 \leq \ldots \leq S_N$.

Theorem 9 indicates the lower bound for the number of *n*-input *A*-operations needed to form an MCM block. If pipelining is added, more *R*-operations than the aforementioned lower bound may be needed because the constants with fewer prime factors may use non-multiplicative graphs, which require extra *R*-operations (see Theorems 5 to 8). Besides, all the outputs of the PMCM block must have equal number of depth levels to balance the input-output delay, which also may require extra *R*-operations. Based on these observations, Theorem 10 extends the lower bound provided in Theorem 9 by identifying at least how many extra *R*-operations would be needed. From these theorems we obtain the lower bound for the number of *R*-operations needed to form a PMCM block.

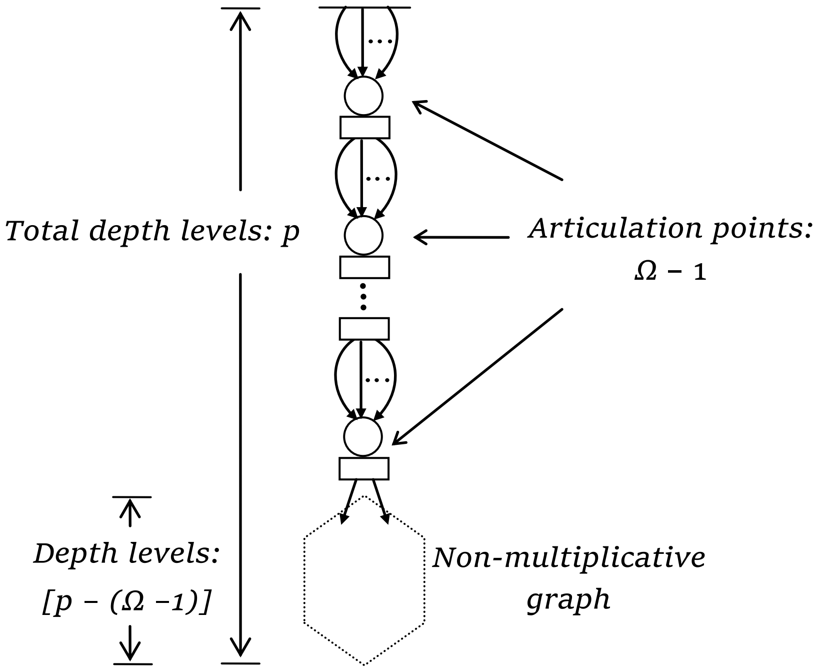


Figure 4.8. Generalized graph that generates the maximum number of non-zero digits, n^p, with the minimum number of *R*-operations in a multiplicative graph for constants with less prime factors than the minimum number of depth levels.

Theorem 9. *At least K n-input A-operations are needed to build an MCM block, where K is given by*

$$K = \lceil \log_n(S_1) \rceil + \sum_{i=1}^{N-1} E(S_i, S_{i+1}), \tag{4.3}$$

with

$$E(S_i, S_{i+1}) = \begin{cases} 1; & S_i = S_{i+1}, \\ \left\lceil \log_n \dfrac{S_{i+1}}{S_i} \right\rceil; & S_i < S_{i+1}. \end{cases} \quad (4.4)$$

Proof. Recall that every A-operation has only one possible configuration and therefore can generate only one fundamental. Simply shifted (i.e., scaled by a power of two) versions of that fundamental can be obtained from that A-operation. Since the target constants are integer and odd by definition, it is not possible to obtain two target constants from the same A-operation. Therefore, there must be at least N n-input A-operations for the N constants. Note that, since the terms S_i are sorted in ascendant order, S_1 corresponds to the simplest constant, i.e., the one with the smallest number of non-zero digits. From Theorem 1 we have that with p depth levels we can obtain n^p non-zero digits at most. By using the relation $n^p \geq S_1$, we have that the minimum number of levels necessary to generate S_1 non-zero digits is $\lceil \log_n(S_1) \rceil$, which implies the existence of at least $\lceil \log_n(S_1) \rceil$ A-operations for that constant. Finally, if $S_{i+1} > n \times S_i$ holds, we have that a single A-operation is not able to generate the constant c_{i+1} if there are only coefficients with at most S_i digits available because the number of non-zero digits at the output of an A-operation is at most the sum of the number of non-zero digits at its inputs. Therefore, at least $\lceil \log_n(S_{i+1} / S_i) \rceil$ A-operations will be required. This proof is an straightforward extension of the proof given in [3] for the lower bound of 2-input A-operations that form an MCM block. ■

Theorem 10. *At least L R-operations are needed to build a PMCM block, where L = K + F + G, with*

$$F = \begin{cases} \max_i\{\lceil \log_n(S_i) \rceil - \Omega_i\}; & \forall\ i \text{ such that } \Omega_i < \lceil \log_n(S_i) \rceil, \\ 0; & \text{otherwise.} \end{cases} \tag{4.5}$$

$$G = \sum_{i=1}^{N-1} \lceil \log_n(S_N) \rceil - \lceil \log_n(S_i) \rceil \tag{4.6}$$

and K given in (4.3).

Proof. Consider that there is a constant c_m that satisfies $\Omega_m < \lceil \log_n(S_m) \rceil$ and, if there are more constants that satisfy such condition, c_m has the greatest difference $[\lceil \log_n(S_m) \rceil - \Omega_m]$. From Theorem 8 we have that the constant can be formed by cascading a non-multiplicative graph with a completely multiplicative graph, where the non-multiplicative graph needs $2[\lceil \log_n(S_m) \rceil - (\Omega_m - 1)] - 1$ R-operations. Since Theorem 9 has not taken into consideration the number of prime factors, only $[\lceil \log_n(S_m) \rceil - (\Omega_m - 1)]$ A-operations have been accounted in that theorem, under the assumption that the constant c_m can be constructed with the optimal completely multiplicative graph. Therefore, at least $[\lceil \log_n(S_m) \rceil - (\Omega_m - 1)] - 1$ extra R-operations must be included when pipelining is applied, which explains the term F. The term G is explained by the fact that extra R-operations may be needed to achieve the same number of pipelined stages from input to output in every constant. Since the minimum depth level of a constant is given by $\lceil \log_n(S) \rceil$, the differences between the minimum depth level of the constant c_N (which has the greatest depth level among other constants) and the minimum depth levels of the other constants are accumulated in the term G. ■

From Theorem 10, we can express the lower bound for the number of R-operations in the PMCM case as

$$L_{PMCM} = \left\lceil \log_n(S_1) \right\rceil + \sum_{i=1}^{N-1} \left(\left\lceil \log_n(S_N) \right\rceil - \left\lceil \log_n(S_i) \right\rceil \right) + \sum_{i=1}^{N-1} E(S_i, S_{i+1}) + F \,, \qquad (4.7)$$

with $E(S_i, S_{i+1})$ given in (4.4) and F given in (4.5).

4.3 Results and comparisons

In this section, comparisons of the proposed lower bounds with the lower bounds currently available in literature are presented, detailing PSCM and PMCM cases in Subsections 4.3.1 and 4.3.2, respectively. In all cases, two and three-input additions were considered.

First, the PSCM case is addressed for n = 2 (i.e., 2-input additions) with an illustration of the lower bounds averaged over all the constants with a wordlength of B bits, where B goes from 1 to 14. This illustration compares the proposed lower bound with the existing lower bounds from [3] and [4], showing that the proposed lower bound is tighter. An example is also included, where the pipelined shift-and-add multipliers for constants 11467, 11093 and 13003 are constructed with 2-input and 3-input additions.

The effectiveness of the PMCM lower bound is demonstrated by examples, where pipelined shift-and-add multiple constant multiplication blocks are constructed using the algorithms from [7] —Output Fundamental Last (OFL)—, [8] —Optimal Pipelined Adder Graph (Optimal PAG), [22] —Reduced Slice Graph (RSG)—, [26] —Heuristic with Cumulative Benefit ($\mathrm{H_{cub}}$)— and [32] —Reduced Adder Graph (RAG)— for the case of 2-input additions, and the algorithm from [10] —Optimal Pipelined Adder Graph Ternary (Optimal PAGT)— for the case of 3-input additions. The proposed lower bound is compared with the lower bound from [3] in the case of 2-input additions and, in most of the cases, it provides better estimation of the number of required *R*-operations. For n = 3 (i.e., 3-input additions), there are no theoretical lower bounds currently available in literature. Thus, the proposed lower bound is only compared with the solution from [10]. In that case, the proposed lower bound falls short only by one *R*-operation.

4.3.1 SCM case

The lower bounds from methods [3] and [4], as well as the proposed lower bound L_{PSCM} from (4.2) are averaged for all constants with *B* bits, where *B* is between 1 and 14. These averages are shown in Figure 4.9. We can observe the tightening of the proposed lower bound, i.e., the proposed lower bound in general is greater than the lower bounds currently available in literature. Table 4.2 presents, for n = 2, the percentage of constants with improved lower bounds among 10,000 14-bits random constants and among 10,000 *B*-bits random constants, with *B* between 15 and 32.

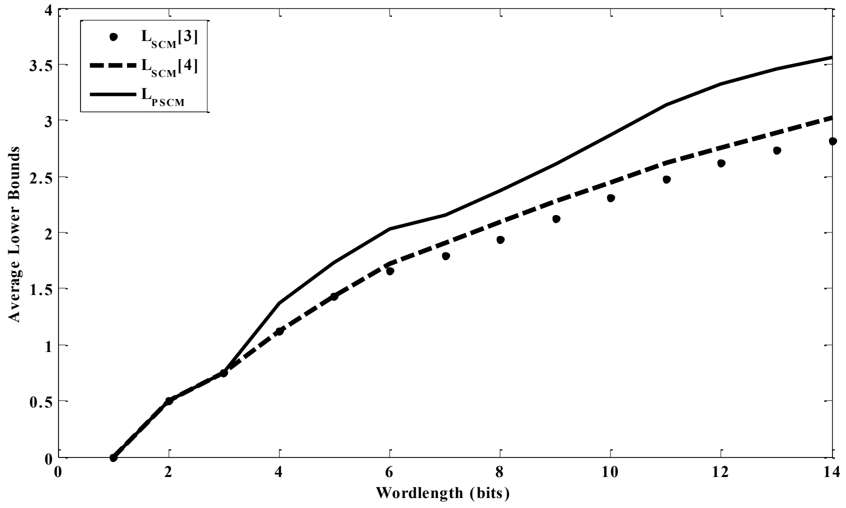


Figure 4.9. Average lower bounds for PSCM cases.

Table 4.2. Percentage of constants with improved lower bounds.

Word-length	L_{SCM} [3]	L_{SCM} [4]
B = 14 bits	54%	45%
14< B $\leq$ 32	63%	55%

Example 1 presents the pipelined shift-and-add multipliers for constants 11467, 11093 and 13003, constructed with 2-input additions (shown in Figures 4.10(a), 4.10(c) and 4.10(e), respectively) and 3-input additions (shown in Figures 4.10(b), 4.10(d) and 4.10(f), respectively). In all the cases, the optimal solutions have the number of *R*-operations predicted by the proposed lower bound. Besides, for the case of two-input additions, the proposed lower bound outperforms the ones from [3] and [4] because the lower bound from [3] falls short by 2 *R*-operations and the lower bound from [4] falls short by one *R*-operation.

Example 1. The constants 11467, 11093 and 13003 have similar graph and the same lower bounds as shows in Table 4.3. The corresponding graphs are presented in Figure 4.10.

Table 4.3. Number of *R*-operations.

Constant	Estimated number of *R*- operations (n = 2)			Estimated number of *R*-operations (n = 3)
	L_{SCM}[3]	L_{SCM}[4]	L_{PSCM}	L_{PSCM}
11467	3	4	5	3
11093	3	4	5	3
13003	3	4	5	3

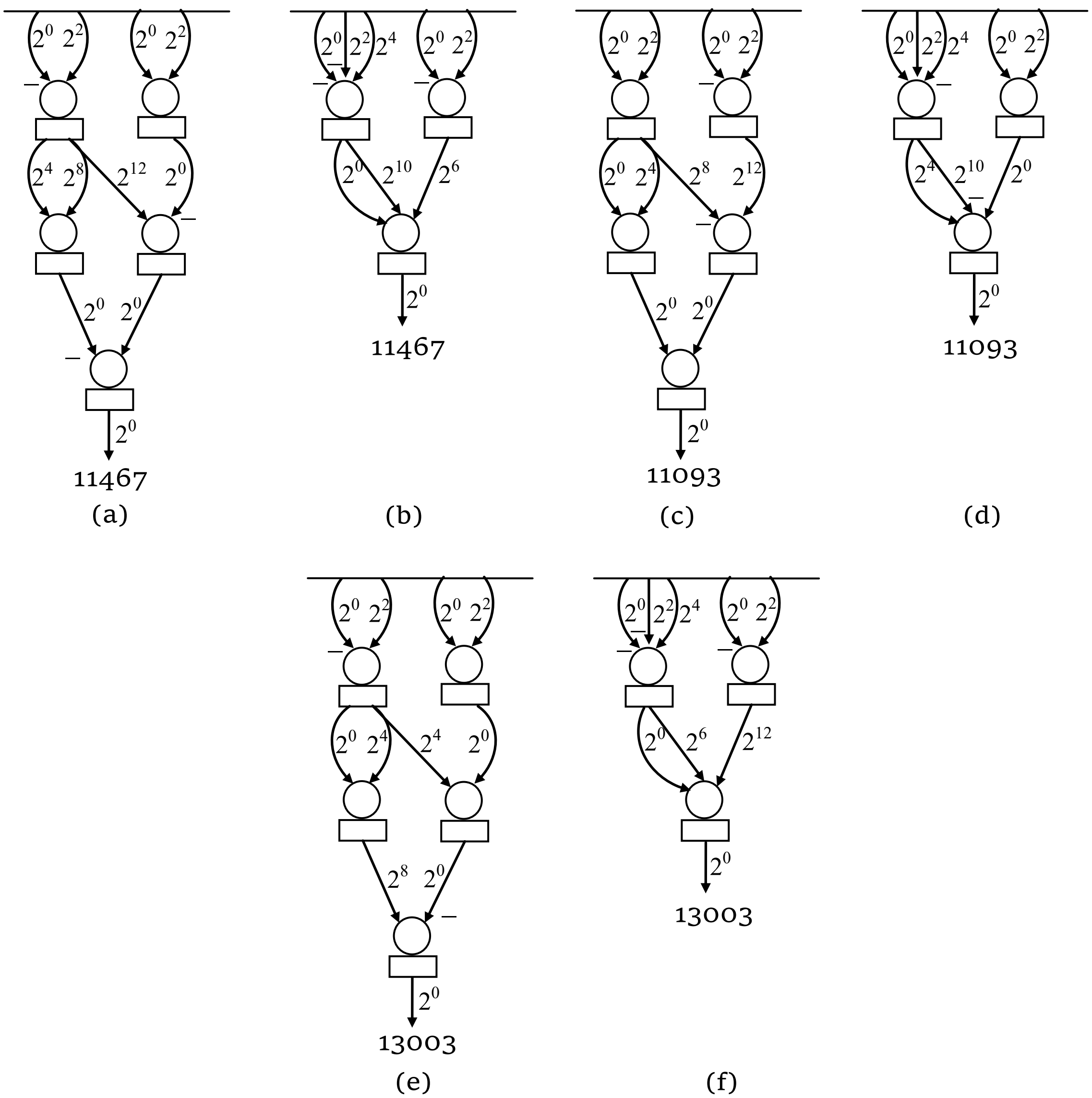


Figure 4.10. (a) Two-input adder graph of constant 11,467, (b) Three-input adder graph of constant 11,467, (c) Two-input adder graph of constant 11,093, (d) Three-input adder graph of constant 11,093, (e) Two-input adder graph of constant 13,003, and (f) Three-input adder graph of constant 13,003.

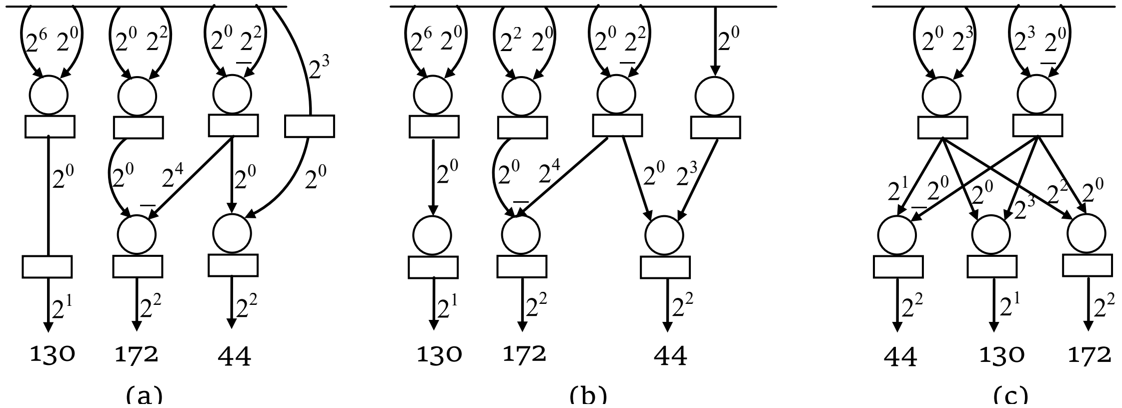
4.3.2. MCM case

Example 2. The multiplier block with constants from the set {44, 130, 172} (example given in [8]) has the estimate number of *R*-

operations as shown in Table 4.4. The resulting graphs are shown in Figure 4.11. The proposed lower bound outperforms the bound from [3].

Table 4.4. Resulting *R*-operations for example 2.

Algorithm	*R*- operations
H_{cub} (method [26] with additional pipelining)	7
PAG using heuristic pipelining (preliminary solution from [8])	7
Optimal PAG (method [8])	5
L_{MCM} [3]	3
L_{PMCM}	4



2^6 2^0 2^0 2^2 2^0 $\underline{2^2}$ 2^3
2^0 2^0 $\underline{2^4}$ 2^0 2^0
2^1 2^2 2^2
130 172 44
2^6 2^0 2^2 2^0 2^0 $\underline{2^2}$ 2^0
2^0 2^0 $\underline{2^4}$ 2^0 2^3
2^1 2^2 2^2
130 172 44
2^0 2^3 2^3 $\underline{2^0}$
2^1 $\underline{2^0}$ 2^0 2^3 2^2 2^0
2^2 2^1 2^2
44 130 172

(a) (b) (c)

Figure 4.11. (a) MCM block obtained by H_{cub} algorithm with pipelining, (b) MCM block obtained by PAG algorithm, and (c) MCM block obtained by Optimal PAG algorithm.

Example 3. The multiplier block with constants from the set {3, 13, 21, 37} (Example given in [7]) has the estimate number of *R*-operations as shown in Table 4.5. The resulting graphs are shown in Figure 4.12. The proposed lower bound outperforms the bound from [3].

Table 4.5. Resulting *R*-operations for example 3.

Algorithm	*R*- operations
RAG (method [32] with additional pipelining)	13
RSG (method [22])	7
OFL (method [7])	6
L_{MCM} [3]	4
L_{PMCM}	6

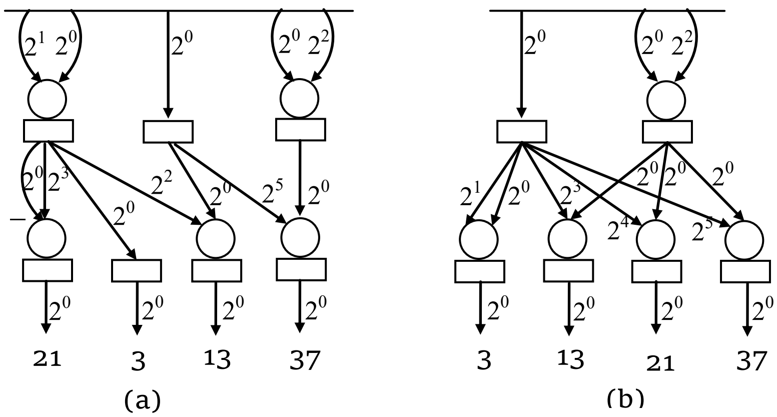


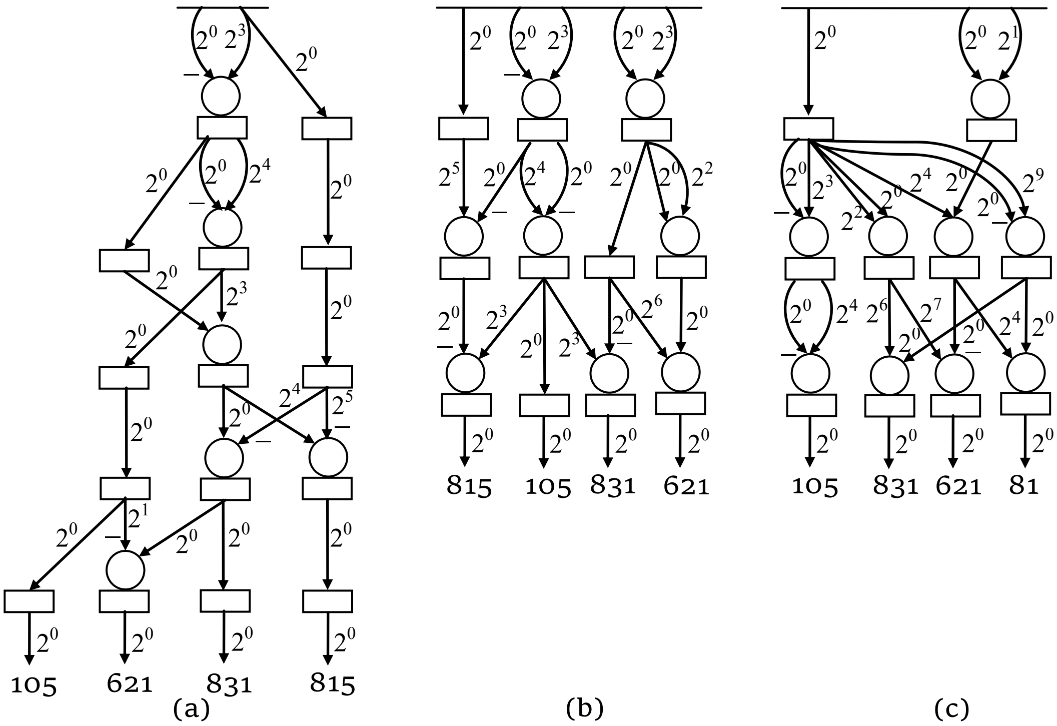
Figure 4.12. (a) MCM block obtained by RSG algorithm, and (b) MCM block obtained by OFL algorithm.

Example 4. The multiplier block with constants from the set {815, 621, 831, 105} (Example given in [7]) has the estimate number of *R*-operations as shown in Table 4.6, the resulting graphs are shown in Figure 4.13. The proposed lower bound outperforms the bound from [3].

Table 4.6. Resulting *R*-operations for example 4.

Algorithm	*R*- operations
RAG (method [32] with additional pipelining)	15

H_{cub} (method [26] with additional pipelining)	11
OFL (method [7])	10
L_{MCM} [3]	5
L_{PMCM}	8



(a) (b) (c)

Figure 4.13. (a) MCM block obtained by RAG algorithm with pipelining, (b) MCM block obtained by H_{cub} algorithm, and (c) MCM block obtained by OFL algorithm.

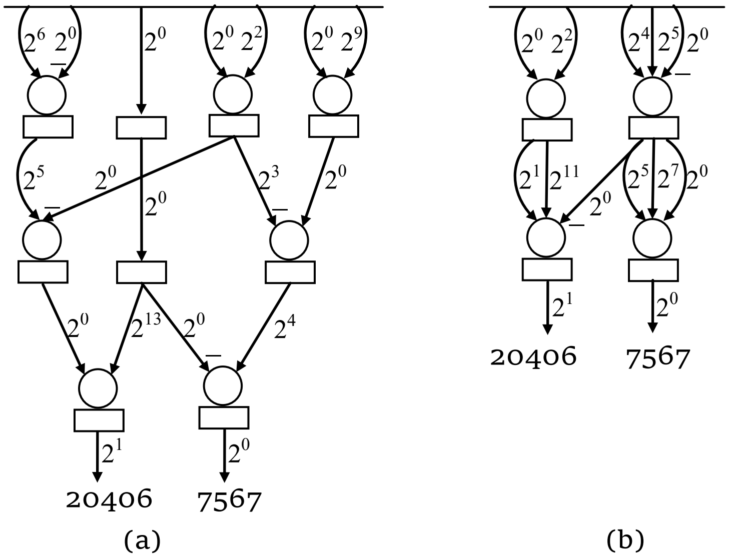
Example 5. The multiplier block with constants from the set {7567, 20406} (example given in [10]) has the estimate number of *R*-operations as shown in Table 4.7 for two-input adders and Table 4.8 for three-input adders. The corresponding graphs are shown in Figure 4.14.

Table 4.7. Using two-input adders

Algorithm	*R*- operations
PAG (method [8])	9
L_{MCM} [3]	4
L_{PMCM}	4

Table 4.8. Using three-input adders

Algorithm	*R*- operations
PAGT (method [10])	4
L_{PMCM}	3



(a)

(b)

Figure 4.14. (a) Two-input adder graph by PAG algorithm, and (b) Three-input adder graph by PAGT algorithm.

4.4 Conclusions

New theoretical lower bounds for the number of *R*-operations in the fully pipelined Single Constant Multiplication (SCM) and the fully pipelined Multiple Constant Multiplication (MCM) cases for *n*-input adders have been presented. The increase of the number of operations due to the use of pipelining registers was considered to develop the new lower bounds. It was observed that the use of articulation points allows a rapid increase of the number of non-zero digits from a depth level to the next depth level. The new theoretical lower bounds achieve better estimation of the number of required operations needed to implement an SCM block or an MCM block in comparison to theoretical lower bounds previously introduced in literature.

4.5 References

[1] Guo, R., DeBrunner, L. S., and Johansson, K. “Truncated MCM using pattern modification for FIR filter implementation,” *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 3881-3884, 2010.

[2] Aksoy, L., Günes, E. O., and Flores, P. “Search algorithms for the multiple constant multiplication problem: Exact and approximate,” *Microprocessors and Microsystems*, vol. 34, no.5, pp. 151-162, 2010.

[3] Gustasson, O. “Lower bounds for constant multiplication problems,” *IEEE Trans. Circuits and Syst. II: Express briefs*, vol. 54, no.11, pp. 974-978, 2007.

[4] Romero, D. E. T., Meyer-Baese, U., and Dolecek, G. J. “On the inclusion of prime factors to calculate the theoretical lower bounds in

multiplierless single constant multiplications," *EURASIP Journal on Advances in Signal Processing*, 122, pp. 1-9, 2014.

[5] Mirzaei, S., Kastner, R., and Hosangadi, A. "Layout Aware Optimization of High Speed Fixed Coefficient FIR Filters for FPGAs," *Int. Journal of Reconfigurable Computing*, pp. 1 – 17, 2010.

[6] Kumm, M. "High speed low complexity FPGA-based FIR filters using pipelined adder graphs," *Int. Conference on Field Programmable Technology (FPT)*, pp. 1-4, 2011.

[7] Meyer-Baese, U., Botella, G., Romero, D. E. T. and Kumm, M. "Optimization of high speed pipelining in FPGA-based FIR filter design using Genetic Algorithm," *Proc. SPIE 8401, Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering X,* 2012.

[8] Kumm, M., Zipf, P., Faust, M., and Chang, C. H. "Pipelined adder graph optimization for high speed multiple constant multiplication," *IEEE Int. Symp. on Circuits and Systems*, pp. 49-52, 2012.

[9] Kumm, M., Fanghanel, D., Moller, K., Zipf, P., and Meyer-Baese, U. "FIR filter optimization for video processing on FPGAs," *EURASIP Journal on Advances in Signal Processing*, 2013.

[10] Kumm, M., Hardieck, M., Willkomm, J., Zipf, P., and Meyer-Baese, U. "Multiple constant multiplications with ternary adders," *International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-8, 2013.

[11] Kumm, M., and Zipf, P. "Pipelined compressor tree optimization using integer linear programming," *24th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-8, 2014.

[12] Kumm, M., and Zipf, P. "Efficient high speed compression trees on Xilinx FPGAs," MBMV, pp. 171-182, 2014.

[13] Aksoy, L., Costa, E., Flores, P., and Monteiro, J. "Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications," *IEEE Trans. Comput.-Aided Des. Integr. Circuits*, vol. 27, no.6, pp. 1013 – 1026, 2008.

[14] Aksoy, L., Costa, E., Flores, P., and Monteiro, J. "Finding the optimal tradeoff between area and delay in multiple constant multiplications," *sevier J. Microprocess. Microsyst.*, vol. 35, no. 8, pp. 729 – 741, 2011.

[15] Dempster, A. G., Dimirsoy, S. S., and Kale, I. "Designing multiplier blocks with low logic depth," *in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 5, pp. 773 – 776, 2002.

[16] Faust, M., and Chip-Hong, C. "Minimal logic depth adder tree optimization for multiple constant multiplication," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 457 – 460, 2010.

[17] Johansson, K., Gustafsson, O., DeBrunner, L. S., and Wanhammar, L. "Minimum adder depth multiple constant multiplication algorithm for low power FIR filters," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1439 – 1442, 2011.

[18] Dempster, A. G., and Macleod, M. D. "Using all signed-digit representations to design single integer multipliers using subexpression elimination," *in Proceedings of the IEEE International*

Symposium on Circuits and Systems (ISCAS), vol. 3, pp. 165 – 168, 2004.

[19] Aksoy, L., Costa, E., Flores, P., and Monteiro, J. *Multiplierless design of linear DSP transforms*, in VLSI-SoC: Advanced Research for Systems on Chip, Springer, Chap. 5, pp. 73 – 93, 2012.

[20] Ho, Y. H., Lei, C. U, Kwan, H. K., and Wong, N. "Global optimization of common subexpressions for multiplierless synthesis of multiple constant multiplications," *in Proceedings of Asia and South Pacific Design Automation Conference*, pp. 119 – 124, 2008.

[21] Hosangadi, A., Fallah, F., and Kastner, R. "Simultaneous optimization of delay and number of operations in multiplierless implementation of linear systems," *in Proceedings of International Workshop on Logic Synthesis*, 2005.

[22] Macpherson, K., and Stewart, R. "Rapid prototyping—area efficient FIR filters for high speed FPGA implementation," *IEE Proc. Vision Image Signal Process.*, vol. 153, no.6, pp. 711 – 720, 2006.

[23] Meyer-Baese, U., Chen, J., Chang, C.H., and Dempster, A. "A comparison of pipelined RAGn and DA FPGA-based multiplierless filters," *in Proceedings of IEEE Asian-Pacific Conference on Circuits and Systems*, pp. 1555 – 1558, 2006.

[24] Aksoy, L., Costa, E., Flores, P., and Monteiro, J. "Design of low-complexity digital finite impulse response filters on FPGAs," *in Proceedings of Design, Automation and Test in Europe Conference*, pp. 1197 – 1202, 2012.

[25] Faust, M., and Chip-Hong, C. "Bit-parallel Multiple Constant Multiplication using Look-Up Tables on FPGA," *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pp. 657 – 660, 2011.

[26] Voronenko, Y., and Püschel, M. "Multiplierless multiple constant multiplication," *ACM Trans. Algorithms*, vol. 3, no.2, 2007.

[27] Oh, W. J., and Lee, Y. H. "Implementation of programmable multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. 42, no.8, pp. 553 – 556, 1995.

[28] Meyer-Baese, U. *Digital Signal Processing with Field Programmable Gate Arrays*, Springer, 2014.

[29] Bull, D. R., and Horrocks, D. H. "Primitive operator digital filters," in *IEE Proceedings G - Circuits, Devices and Systems*, vol. 138, no.3, pp. 401-412, 1991.

[30] Johansson, K., Gustafsson, O. and Wanhammar, L. "Switching activity estimation for shift-and-add based constant multipliers," *2008 IEEE International Symposium on Circuits and Systems*, pp. 676-679, 2008.

[31] Chen, J., and Chang, C. H. "High-Level Synthesis Algorithm for the Design of Reconfigurable Constant Multiplier," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 12, pp. 1844-1856, 2009.

[32] Dempster, A. G., and Macleod, M. D. "Use of minimum-adder multiplier blocks in FIR digital filters," in *IEEE Trans. Circuits and Systems II – Analog Digital Signal Process.*, vol. 42, no.9, pp. 569-577, 1995.

[33] Gustafsson, O., Dempster, A. G., Johansson, K., Macleod, M. D., and Wanhammar, L. "Simplified design of constant coefficient multipliers," *Circ. Syst. Signal Process*, vol. 25, no.2, pp. 225–251, 2006.

[34] Parhi, K. K. *VLSI digital signal processing systems:design and implementation*, John Wiley & Sons, 2007.

[35] Guftasson, O. *Contributions to Low-complexity digital filters*, Linköping Studies and technology dissertations, 2003, No. 837.

[36] Kastner, R., Hosangadi, A., and Fallah, F. *Arithmetic optimization techniques for hardware and software design*, Cambridge University Press, 2010.

Conclusions

Novel methods to design low-complexity linear-phase Finite Impulse Response (FIR) filters have been introduced in this thesis, as well as efficient architectures derived from these methods. Two specific cases have been investigated here: low-pass filtering for decimation processes and digital filters with constant coefficients implemented under the shift-and-add approach. The reason is that these cases are particularly useful for applications in digital communications.

We have observed that splitting the filters into simple subfilters allows to achieve low-complexity solutions especially useful in the design of decimators. The comb and cosine subfilters have been employed here due to their low computational complexity and low utilization of hardware resources. First, a simple heuristic has been introduced to design low-pass FIR filters using a cascade of comb and cosine subfilters to provide the desired attenuation, along with a cascaded subfilter optimized to obtain a band-edge shaping characteristic and to correct the passband droop of the comb-cosine prefilter. Taking this method as starting point, we have found that using cosine filters sharpened with Chebyshev polynomials is an interesting alternative to the comb-cosine cascade when low delay is desired. We have presented the mathematical demonstration that the application of Chebyshev sharpening to cosine and expanded cosine filters results in filters with zeros on the unit circle, that is, with Minimum Phase (MP) characteristic. Thus, they can form useful prefilters that can provide the attenuation for an overall Linear Phase (LP) filter or for an MP FIR filter. Moreover, these filters are a general case where the cascaded expanded cosine filters are a subset. Besides, the aforementioned prefilters have a low computational complexity because they do not need multipliers.

The design of comb-based decimators has been addressed from two approaches. In both cases, the objective has been correcting the passband droop and improving the worst-case attenuation with an as low as possible augmentation in the complexity of the resulting architecture. In the first approach, we have taken advantage of the improved sharpening of Harnett and Boudreaux to enhance the magnitude characteristics of previously compensated comb filters. The resulting proposed structures achieve better trade-offs in magnitude response improvement and computational complexity in comparison with other similar schemes where the traditional Kaiser-Hamming sharpening has been employed. In the second approach, we have taken advantage of the Chebyshev sharpening to improve uniquely the stopband attenuation of comb filters, whereas the passband-droop correction is performed at a low rate via compensation filtering. Using the Chebyshev sharpening as starting point, we have derived an efficient comb-based decimation architecture which improves the aliasing rejection and simultaneously consumes less power, uses less hardware resources and operates at higher rates in comparison with other recent methods from literature. Moreover, we have found that, in comparison with the state-of-the-art second-order compensators, the proposed fourth-order compensators, applied in wide passbands, can improve the correction of the droop by nearly four times, and the complexity of these compensators increases less than twice, which is a useful trade-off. Between the two aforementioned approaches, the one based in Chebyshev sharpening offers better results.

Finally, novel theoretical lower bounds for the number of pipelined operations that are needed in Single Constant Multiplication (SCM) and Multiple Constant Multiplication (MCM) blocks have been proposed. These lower bounds can be calculated for *n*-input additions/subtractions, for any *n*. In comparison to theoretical lower bounds previously introduced in literature, the proposed bounds achieve better estimation of the number of required operations needed to implement a fully pipelined SCM block or a fully pipelined MCM block, and this is because the pipelining registers were considered as costly elements, along with the *n*-input additions/subtractions. The proposed lower bounds are particularly important because they fit well for the implementation of pipelined SCM or MCM blocks on the newest families of Field Programmable Gate Arrays (FPGAs), which currently are a preferred platform for DSP algorithms.

Publications

Journals (JCR)

[3] **M. G. C. Jimenez**, U. Meyer-Baese and G. J. Dolecek, “Theoretical lower bounds for parallel pipelined shift-and-add constant multiplications with n-input arithmetic operators,” Submitted to EURASIP Journal on Advances in Signal Processing, Springer.

[2] **M. G. C. Jimenez**, U. Meyer-Baese and G. J. Dolecek, “Computationally efficient CIC-based filter with embedded Chebyshev sharpening for the improvement of aliasing rejection,” *Electronics Letters*, IET, online December 2016.

[1] **M. G. C. Jimenez**, D. E. T. Romero and G. J. Dolecek, “Minimum phase property of Chebyshev-sharpened cosine filters,” *Mathematical Problems in Engineering*, Hindawi, vol. 2015, pp. 1-14, 2015.

Conferences in journals or books

[2] **M. G. C. Jimenez** and G. J. Dolecek, “On compensated three-stages sharpened comb decimation filter,” *Applied Engineering Sciences: Proceedings of the 2014 AASRI International Conference on Applied Engineering Sciences*, Edited by Wei Deng, CRC Press, LA, USA, Chapter 4, pp. 17-21, 2014.

[1] **M. G. C. Jimenez** and G. J. Dolecek, “Application of generalized sharpening technique for two-stage comb decimator filter design,” *Procedia Technology*, Elsevier, vol. 7, pp. 142-149, 2013.

BEST PAPER AWARD AT THE CONFERENCE CIIECC 2013, APRIL 2013.

Proceedings

[6] **M. G. C. Jimenez**, D. E. T. Romero and G. J. Dolecek, “An efficient design of baseband filter for mobile communications,” *IEEE International Conference on Electro/Information technology, EIT 2016*, Grand Forks, North Dakota, USA, pp. 368-371, 2016.

[5] **M. G. C. Jimenez**, D. E. T. Romero and G. J. Dolecek, “On simple comb decimation structure based on Chebyshev sharpening,” *IEEE Latin American Symp. on Circuits and Systems, LASCAS*, Montevideo, Uruguay, pp. 1-4, 2015.

[4] **M. G. C. Jimenez**, D. E. T. Romero, G. J. Dolecek, and M. Laddomada “Wide-band CIC Compensators Based on Amplitude Transformation,” *9th IEEE International Caribbean Conference on Devices, Circuits and Systems, ICCDCS*, Playa del Carmen, Mexico, pp. 100-103, 2014.

[3] D. E. T. Romero, **M. G. C. Jimenez** and G. J. Dolecek “Design of Chebyshev Comb Filter (CCF)-based decimators with compensated passband,” *5th IEEE Latin American Symposium on Circuits and Systems, LASCAS*, Santiago, Chile, pp. 1-4, 2014.

[2] **M. G. C. Jimenez** and G. J. Dolecek, “On the design of very sharp narrowband FIR filters by using IFIR technique with time-multiplexed subfilters,” *2013 IEEE International Conference on Advances in Computing, Communications and Informatics, ICACCI*, Mysore, India, pp. 2002-2006, 2013.

[1] **M. G. C. Jimenez**, V. C. Reyes and G. J. Dolecek, “Sharpening of non-recursive comb decimation structure,” *13th IEEE International Symposium on Communications and Information Technologies, ISCIT*, Surat Thani, Thailand, pp. 458-463, 2013.

Book Chapters

[2] **M. G. C. Jimenez**, D. E. T. Romero and G. J. Dolecek, “Comb filters: Characteristics and current applications,” *Encyclopedia of Information Science and Technology*, 4ta. Ed., IGI Global Publishing, Julio 2017.

[1] **M. G. C. Jimenez**, D. E. T. Romero and G. J. Dolecek, “Comb filters: Characteristics and applications,” *Encyclopedia of Information Science and Technology*, 3ra. Ed., IGI Global Publishing, 2014.