UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

FIR FILTER DESIGN FOR AREA EFFICIENT IMPLEMENTATION

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirement for the

degree of

Doctor of Philosophy

By

XIAOJUAN HU
Norman, Oklahoma
2005

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

FIR FILTER DESIGN FOR AREA EFFICIENT IMPLEMENTATION

A Dissertation APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Linda DeBrunner
(Chair)

Dr. Victor DeBrunner
(Co-chair)

Dr. John E. Fagan

Dr. Joseph Havlicek

Dr. Murad Özaydin

Dedicated to my family and friends…

ACKNOWLEDGEMENTS

My first thanks go to my advisors Dr. Linda DeBrunner and Dr. Victor DeBrunner for giving me a very interesting research topic. Without their consistent guidance, support and encouragement during my research, the completion of this thesis would not have been possible.

I am also very grateful to Dr. John E. Fagan, Dr. Joseph Havlicek and Dr. Murad Özaydin for being the members of my dissertation defense committee and for their helpful suggestions concerning this work.

My special thanks should go to Actel, Inc. for their support and Andreas Schwarztrauber and Baoxian Tang for their assistance in the experiments.

I would also like to thank Dr. Longji Wang, Shuqing Liu, Meirong Huang, Dayong Zhou and many other colleagues for their help and friendship.

My heartfelt thanks should also go to my parents, brothers and sisters for their many years of understanding, patience and encouragement.

Finally, I would like to dedicate this dissertation to my beloved husband, Dr. Kaiming Xia for his unstoppable encouragement and my daughter, Jiajie Xia and my son Andrew Y. Xia. Their understanding, love, encouragement and patience were essential in the completion and success of my PhD study at The University of Oklahoma.

Xiaojuan Hu

Table of Contents

List of Tables

List of Figures

Abstract

FIR filters are preferred for many Digital Signal Processing applications as they have several advantages over IIR filters such as the possibility of exact linear phase, shorter required wordlength and guaranteed stability. However, FIR filter applications impose several challenges on the implementations of the systems, especially in demanding considerably more arithmetic operations and hardware components. This dissertation focuses on the design and implementation of FIR filters in hardware to reduce the space required without loss of performance.

In this dissertation, a variable precision algorithm based on sensitivity analysis is proposed for reducing the wordlength of the coefficients and/or the number of nonzero bits of the coefficients to reduce the complexity required in the implementation. Further space savings is possible if the proposed algorithm is associated with our optimal structures and derived scaling algorithm. We also propose a structure to synthesize FIR filters using the improved prefilter equalizer structure with arbitrary bandwidth, and our proposed filter structure reduces the area required. Our improved design is targeted at improving the prefilters based on interpolated FIR filter and frequency masking design and aims to provide a sharp transition-band as well as increasing the stopband attenuation. We use an equalizer designed to compensate the prefilter performance. In this dissertation, we propose a systematic procedure for designing FIR filters implementations. Our method yields a good design with low coefficient sensitivity and small order while satisfying design specifications. The resulting hardware implementation is suitable for use in custom hardware such as VLSI and Field Programmable Gate Arrays (FPGAs).

Chapter 1 Introduction

For a digital FIR filter realization, developing efficient algorithms with minimal hardware and negligible performance degradation is strongly desired from the viewpoint of IC implementation. The factors that impact the performance of an FIR filter in terms of the amount of hardware circuitry required are the number of operations such as additions and multiplications. Therefore, the development of the techniques to reduce the number of operations required has become very attractive, especially for FPGA and VLSI implementation. Finding efficient methods that maximize the performance while keeping the cost as low as possible is one of the goals of digital filter design.

This thesis focuses on the design and implementation of FIR filters as described above. Before starting the discussion in detail, some basic filtering concepts upon which our methods rely are reviewed.

1.1 Basic FIR Digital Filter Structure

A linear time invariant (LTI) FIR filter [1] is one of the basic building blocks common to most DSP systems. The output of an FIR filter is a sequence generated by convolving the sequence of the input samples with N filter coefficients. The filter expression can be described by

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)\,, \tag{1.1}$$

where N is the length of the filter (i.e. N-1 is the order), $h(k)$ denotes the k^{th} coefficient, and $x(n-k)$ denotes the sampled input data at time $n-k$. As depicted in [2][3], except

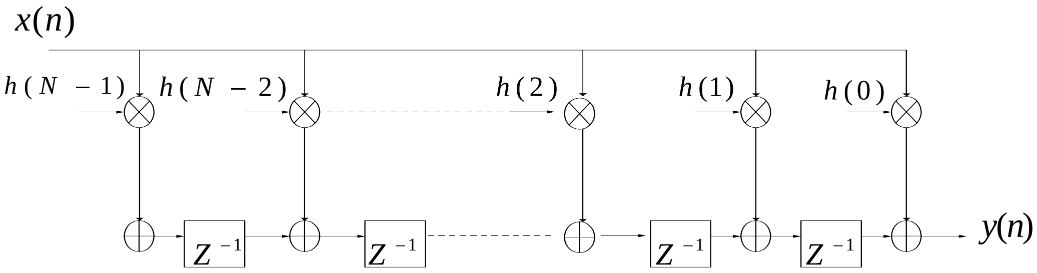


Figure 1-1 Transposed form FIR filter

for a global broadcast input, the transpose form FIR filter has several advantages over direct-form structures for high speed and parallel implementation. In this dissertation this form structure is used to realize our FIR filter. Figure 1-1 shows the block diagram for a transposed form FIR filter. From this diagram, we can see that *N* multiplications and *N-1* additions are required to compute each value in the output sequence.

1.2 Area Estimation of FIR Digital Filter Implementation

Before implementing a filter, it is helpful to know the computational complexity by estimating *N*, *b*, and *A*, where *N* is the length of the filter and thus the number of multipliers, *b* denotes the wordlength used for the multiplier coefficients, and *A* is the number of adders. As the multipliers usually contribute most to the area complexity [3], it is essential to keep them as simple as possible. This can be achieved by minimizing the wordlength and/or the number of the multiplications. Multiplications can be implemented in many ways. An efficient hardware realization of the multiplication algorithm with only shifts and additions is depicted in Figure 1-2. Obviously, its hardware complexity heavily depends on the parameters *N* and *b*.

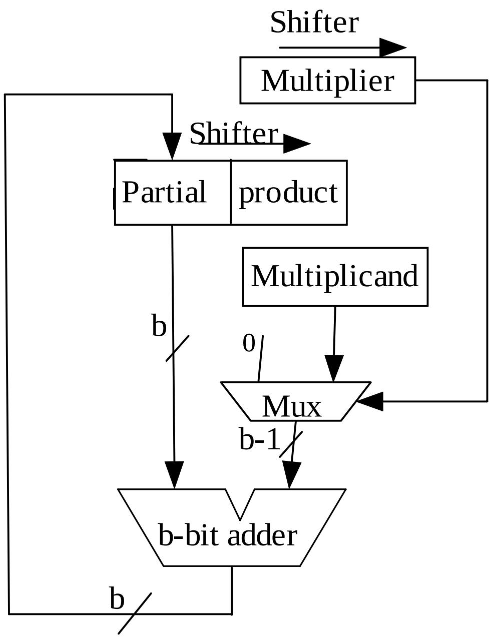


Figure 1-2 Basic Hardware Realization of Multiplication [5]

So, *Nb* is often used to characterize the hardware complexity of the desired digital filter implementation. As a result, implementation techniques can be designed to minimize *Nb*.

The conventional approach to FIR digital filter design is characterized by the following set of parameters:

δ_c - the passband ripple,

δ_s - the stopband ripple,

$\omega_c T$ - passband edge,

$\omega_s T$ - stopband edge.

The required number of multipliers for linear-phase FIR filters is estimated as [6]

$$N \approx 2\pi \frac{-20\log_{10}(\delta_c \delta_s) - 13}{14.6(\omega_s T - \omega_c T)} \tag{1.2}$$

Table 1-1 Multiplier Size of Virtex Family [7]

Multiplier Size	CLBs required in Virtex and Spartan II
8*8	24
10*10	35
12*12	48
16*16	80
32*32	300

For a direct realization of a linear-phase FIR filter, the number of multiplication is approximately half of the filter order because of coefficient symmetry. The number of additions and delay elements are both equal to the filter order.

For a straightforward implementation, a multiplier is needed for each tap. For FPGAs and VLSI, a reasonable hardware estimate can be developed for most applications. One can easily find how many multipliers are required for an FIR filter from equation (1.2) and from the published device characteristic description. For example, Table 1-1 shows how many (Configurable Logic Blocks) CLBs are needed for multipliers of the different multiplier sizes using Virtex devices. A CLB includes a pair of flip-flops and two independent 4-input function generators. These elements offer flexibility in implementing logic functions.

In general, a large number of coefficients are required for an FIR filter. As the filter length increases, the implementation demands more arithmetic operations and hardware components, which makes it hard to fulfill the performance and the silicon area requirements simultaneously. For FPGAs, using an estimate of the area of the desired filter, vendor supplied tables such as the one shown in Table 1-2 can be used to help in the selection of an appropriate device for a specified filter design.

Table 1-2 The Capacity of the Virtex Family of FPGAs [7]

Device	Gates	CLBs
XCV50	57,906	16*24=384
XCV100	108,904	20*30=600
XCV150	164,674	24*36=864
XCV200	236,666	28*42=1176
XCV300	322,970	32*48=1536
XCV400	468,252	40*60=2400
XCV800	888,439	56*84=4704

1.3 Approaches to Reduce the Implementation Complexity

As described in the previous section, the number of multipliers and the length of the multiplier coefficients are key factors in an FIR filter implementation. To reduce area, the implementation can be improved by modifying the filter structure and the design algorithm as well as by focus on implementation techniques. Many techniques[8]-[15] have been proposed in the past few years that focused on minimizing the number of the necessary operations to reduce the computational complexity of the hardware. Two main approaches are used. One approach is to reduce the complexity of each multiplication by minimizing the number of bits or nonzero bits used to represent the filter coefficients[16]-[22]. For a fixed-point implementation, the finite wordlength behavior in digital filters is extremely important since the cost and the complexity of a digital filter depend heavily on the necessary wordlength. Reducing the wordlength reduces the area consumed in the digital filter implementation at the cost of increasing the filter output quantization noise power. Several techniques [23]-[27] have been reported that attempt to reduce such noise while making the wordlength as short as possible. Another approach [28]-[30] tries to reduce the number of multiplications required in implementing the filtering operation of equation (1.1) by using more efficient structures for the filter design. Other different studies also have been proposed. For example, the well-known Distribution Arithmetic (DA) method [5] trades memory for combinational elements, resulting a design suitable for implementation in Look-UP Table (LUT) based FPGA, the smaller the order of the filter (i.e. the smaller number of multiplications), the smaller the LUT size needed. Thus, more space can be reduced.

Almost all methods of the above implementation use the same number of bits for each coefficient in the filter. Although these techniques using uniform coefficient lengths can give sufficiently accurate results to meet the filter requirements in practical filter design, many unnecessary bits are used in uniform quantization, which requires more area in the implementation. For example, to take advantage of their FPGA devices, Actel uses a constant multiplier method (CMULT) to implement FIR filters [31]. However, the implementations still always require relatively large area, which limits the applicability. Therefore, the question arises whether there is some way to reduce the area consumption further for digital filter implementations that can benefit computationally from the properties of the filter. A few researchers have used variable numbers of bits for the coefficients [32]-[36].

1.4 Scope of this Thesis

In this dissertation, a coefficient sensitivity measure is proposed to reduce area. We show that the frequency response of a filter has different sensitivities to different coefficients depending on the coefficient value, as well as the structure (i.e. the coefficient location). Thus, a higher precision can be used for some coefficients and a lower precision for others based on their different sensitivities. Consequently, the average wordlength required by the filter can be reduced, and hence the overall size and complexity of the system implementation can be reduced without impacting the overall filter performance.

Furthermore, relatively simple structures that reduce the arithmetic operations are also proposed. The scheme of pre-filter equalizer structure for narrow band filters is explored in [29][30]. The attractiveness of this cascade structure is that it has good performance and implementation. In this dissertation, an improved prefilter and equalizer structure is proposed to design arbitrary wide-band filters. New pre-filter structures are introduced that use the masking technique, interpolated filter and delay-complementary concepts [37][38][39] to improve the performance of the desired digital FIR filters and greatly reduce computational complexity of the filter realization.

In this dissertation, the problems of reducing the computational complexity of the FIR filter are considered at two levels: algorithm development and structure exploration. For an existing filter, low–space implementation requires reduction of the wordlength and/or the number of nonzero bits in the coefficients to reduce the complexity of the filter. Hence, at the algorithm level, we develop a variable precision method, as well as a scaling algorithm to achieve this goal. At the structure level, two issues are considered to assist in improving the performance of designed FIR filters and reducing the space required. We develop a variable precision equalizer and improve the prefilter structure for arbitrary bandwidth FIR filters.

In chapter 2, a method is proposed that allows an FIR filter to be efficiently implemented using variable wordlength coefficients. The resulting algorithm, which we call the Variable Precision Method (VPM) [40], maintains computational performance while reducing area required. The sensitivity analysis is provided, as well as the examples that demonstrate the feasibility of our VPM to realize filters using variable precision coefficients to reduce area are given.

In chapter 3, a scaling approach is proposed to improve the performance of the fixed-point filter and to minimize the quantization error. The goal of the scaling operation is to adjust the real filter coefficients in order to constrain the coefficient values to an appropriate range more suitable for finite precision arithmetic. We will consider how to determine the Scale Factor (SF) and how to constrain the SF so that it changes the filter coefficients into the optimal values, which can be represented using a small number of nonzero bits while keeping the error noise low.

In chapter 4, a prefilter and equalizer structure is presented. The realization is a cascade connection of a prefilter with good frequency response and relatively simple structure, and an equalizer with reduced order to achieve the low cost implementation. For a narrow filter design, an RRS based structure is presented. Furthermore, for the arbitrary wide band filter, two other kinds of structure based chebyshev polynomial prefilter and halfband prefilter are investigated. To demonstrate the effectiveness of our design methods, examples are given. In additional, we discuss the performance in terms of the coefficients’ sensitivity and the possibility of using shorter wordlength, and thus the resulting hardware area is reduced greatly.

In chapter 5, the novel architecture takes into account the interpolated FIR filter implementation, and a frequency response masking technique is proposed. The resulting architecture development is divided into two levels. The first level is targeted at improving the prefilters based on the frequency masking technique and aims to provide a sharp transition-band, as well as increasing the stopband attenuation. The second level is targeted at reducing the complexity of the equalizers required. As the result, our two-level design allows arbitrary width filters to be realized in simple circuit.

In chapter 6, our conclusions are given along with an outline of areas for future research.

Chapter 2 Variable Precision Method

When a digital filter is implemented using a digital system, it essentially involves quantization of signals and coefficients in the system. As the result, the consideration of finding good quantized coefficients attracts a great deal of attention. The conventional quantization methods [41] always use uniform wordlength coefficients. However, there is some inefficiency existing in the implementation, because not all the coefficients have the same influence on filter frequency response. To reduce the redundancy cost of digital filter implementation, an area efficient method referred to as the Variable Precision Method (VPM) is proposed.

Next, how to predict the variable precision of the quantized coefficients that are required to meet the specification becomes the key problem. Since the frequency response of a filter has different sensitivities to different coefficients depending on the coefficient value, as well as the filter structure, different coefficients can be set to different precisions based on the sensitivities. Thus, in this chapter a method is presented for finding appropriate variable precision fixed-point digital filters based on the coefficient sensitivity analysis to obtain the minimum wordlength for each coefficient. The method can provide benefits in reducing the space required for any existing FIR filter, especially to enhance ASIC and FPGA FIR filter designs since these implementations enable flexibility in the precision of the signal at each stage of the algorithm [42].

In this chapter, the energy of quantization error on the output is used as the method of measuring the filter performance and the analysis of coefficient sensitivity is used to determine the variable coefficient precision. The definition of the coefficient sensitivity and the variable precision coefficient estimation process for an FIR filter is explained in Section 2.1 and 2.2, respectively. In Section 2.3, an algorithm to determine the optimal set of variable coefficients is described. Section 2.4 shows examples with the comparisons of the computational complexity to other approaches. Section 2.5 provides our conclusions.

2.1 Sensitivity of the Frequency Response to Coefficient Quantization

To estimate the length of each coefficient, we start by evaluating the sensitivity $S_{en}(n)$ of the frequency response to each of the coefficients. In this chapter, we use the magnitude coefficient sensitivity $S_{en}(n)$, as the computed sensitivity analysis. This sensitivity reflects the degree of influence on the frequency response of a digital filter that any one of the coefficients will exert under small perturbations. To make straightforward computation and observation, the coefficient sensitivity [43] is defined by setting each coefficient, in turn, to some nearest fixed number, yielding in each case a response $\tilde{H}_n(e^{j\omega})$.

$$S_{en}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} | H(e^{j\omega}) - \tilde{H}_n(e^{j\omega}) |^2 \, d\omega \, , \qquad (2.1)$$

which is the sensitivity of the transfer function $H(e^{j\omega})$ with respect to the variation in one of the multiplier coefficients, where $H(e^{j\omega})$ is the frequency response with infinite-precision coefficients, and $\tilde{H}_n(e^{j\omega})$ is the frequency response with the nth coefficient changed to its nearest fixed number.

The real justification in this chapter for using the $S_{en}(n)$ measure is that the coefficient sensitivity is indicative of the choice of the coefficient wordlength. In other words, more bits can be assigned to sensitive coefficients, and fewer to others, allowing the average wordlength to be reduced and thus the area of implementation to be reduced. The detail of its application will be discussed in section 2.3.

2.2 Analysis of Coefficient Quantization Effects in FIR Filters

2.2.1 Estimation of Quantization Error Based on Uniform Wordlength

According to conventional statistical approaches, the coefficient wordlength is uniform and the quantiation step size for rounding of the coefficients in the fixed point arithmetic is *Q*, and quantization error will have a uniform probability of lying between *–Q/2* and *Q/2* for an arbitrary set of coefficients. The statistical errors in the coefficients can be characterized by zero mean and variance of $\sigma_0^2 = Q^2/12$. If the error of each coefficient quantization is independent, then the output quantization noise can be determined as [44]

$$\sigma_e^2 \geq \sigma_0^2 S \tag{2.2}$$

where σ_e^2 is the filter output quantization variance and σ_0^2 is the variance of the uniform coefficient quantization (determined by smallest quantization step size). This equation can be used as a predictor of the actual output noise power.

S is the filter sensitivity, which also reflects the influence on the filter frequency response when filter coefficients are varied for arbitrary structure, and is defined as

$$S = \sum_i \left(\frac{\partial \left|H(e^{j\omega})\right|}{\partial h_i}\right)^2 \tag{2.3}$$

Since some of the coefficients are more sensitive than others, the corresponding coefficient sensitivity can be given in terms of weighted units in the form of $w_i S_{en}(i)$. However, it is relatively difficult to select weight units to describe an actual digital filter, hence an improved error prediction is developed.

2.2.2 Quantization Error for Variable Precision Coefficient FIR Filter

A more meaningful method for examining the changes in the frequency response due to coefficient quantization can also be obtained as described in this section.

As noted in [41], the transfer function of a direct-form FIR filter can be written as

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \tag{2.4}$$

Quantization of the filter coefficients results in a new transfer function

$$\tilde{H}(e^{j\omega}) = \sum_{n=0}^{N-1} \tilde{h}[n] e^{-j\omega n} = \sum_{n=0}^{N-1} (h[n] + e[n]) e^{-j\omega n} \ , \tag{2.5}$$

Thus, the FIR filter with quantized coefficients can be modeled as a parallel connection of two FIR filters as shown in Figure 2-1, where $H(e^{j\omega})$ represents the desired FIR filter with unquantized coefficients, and $E(e^{j\omega})$ is the FIR filter representing the error in the transfer function due to coefficient quantization.

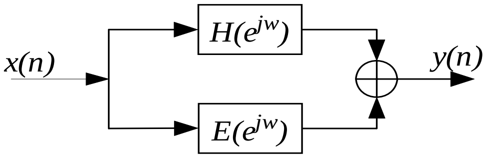


Figure 2-1 Model of the FIR filter with Quantized Coefficients [41]

For convenience, we consider the design of an odd length-N, symmetric FIR filter, hence, the filter coefficients satisfy,

$$h[n] = h[N-n-1], \quad 0 \le n \le N-1 \tag{2.6}$$

so that the frequency response may be rewritten as

$$H(e^{j\omega}) = \left[h[(N-1)/2] + 2\sum_{n=0}^{(N-3)/2} h[n]\cos(((N-1)/2-n)\omega) \right] e^{-j\omega(N-1)/2} \tag{2.7}$$

Substituting

$$g[n] = h[(N-1)/2-n], \quad 0 \le n \le (N-1)/2 \tag{2.8}$$

and canceling the delay factor yields the filter magnitude response as

$$G(e^{j\omega}) = g[0] + 2\sum_{n=1}^{(N-1)/2} g[n]\cos(n\omega) \tag{2.9}$$

Quantizing the coefficients of this filter leads to the new magnitude frequency response:

$$\tilde{G}(e^{j\omega}) = \tilde{g}[0] + 2\sum_{n=1}^{(N-1)/2} \tilde{g}[n]\cos(n\omega) \tag{2.10}$$

From (2.9) and (2.10), the magnitude response of the quantization error can be derived as

$$\begin{aligned} E(e^{j\omega}) &= G(e^{j\omega}) - \tilde{G}(e^{j\omega}) \\ &= e[0] + 2\sum_{n=1}^{(N-1)/2} e[n]\cos(n\omega) \end{aligned} \tag{2.11}$$

where

$$e[n] = g[n] - \tilde{g}[n], \quad n = 0, \ldots, (N-1)/2. \tag{2.12}$$

A realistic quantization error can be derived if we assume $e[n]$ are statistically independent random variables with expected value 0. The expression for the variance of $E(e^{j\omega})$, $\sigma_E^2(\omega)$, is simply given by

$$\begin{aligned}
\sigma_E^2(\omega) &= E\left[E(e^{j\omega})E^*(e^{j\omega})\right] \\
&= E\left[(e[0]+2\sum_{n=1}^{(N-1)/2} e[n]\cos(n\omega))^2\right] \\
&= E\left[e^2[0]+4\sum_{n=1}^{(N-1)/2} e[0]e[n]\cos(n\omega)\right] \\
&\quad + E\left[4\sum_{n=1}^{(N-1)/2} e^2[n]\cos^2(n\omega)\right] \\
&\quad + E\left[8\sum_{n=1}^{(N-3)/2}\sum_{m=n+1}^{(N-1)/2} e[n]e[m]\cos(n\omega)\cos(m\omega)\right]
\end{aligned} \tag{2.13}$$

Now define the mutual correlation in the form as

$$r_e(n,k) = E\left[e[n]e[n+k]\right], \qquad 0 \le n \le (N-1)/2 - k \tag{2.14}$$

Equation (2.13) is then rewritten as

$$\begin{aligned}
\sigma_E^2(\omega) &= r_e(0,0) + 4\sum_{n=1}^{(N-1)/2} r_e(n,0)\cos^2(n\omega)) \\
&\quad + 4\sum_{n=1}^{(N-1)/2} r_e(0,n)\cos(n\omega) \\
&\quad + 8\sum_{n=1}^{(N-3)/2}\sum_{m=n+1}^{(N-1)/2} r_e(n,m)\cos(n\omega)\cos(m\omega)
\end{aligned} \tag{2.15}$$

Here assume that the error due to the quantization of different coefficients is independent and uniformly distributed, that is, the correlation function $r_e(n,k)$ can be expressed in the form

$$r_e(n,k) = \begin{cases} \dfrac{2^{-2d_n}}{12}, & k = 0 \\ 0, & k \ne 0 \end{cases} \tag{2.16}$$

where d_n is the least important nonzero bit in the digit presentation of the n$^{\text{th}}$ coefficient. Thus, we can rewrite equation (2.15) to yield

$$\sigma_E^2(\omega) = \frac{2^{-2d_0}}{12} + \sum_{n=1}^{(N-1)/2} \frac{2^{-2d_n}}{3} \cos^2(n\omega) \tag{2.17}$$

Equation (2.17) will be used to evaluate the quantization error as described in Section 2.3. It is expressed by using variable precision coefficients. Of course it also can be used for uniform wordlength coefficient filters if the least important nonzero bit, d_n, in the digit representation of the coefficients are the same in each case.

Comparing equation (2.17) with equation (2.2), it can be seen that the quatization noise of the digital filter can be computed more efficiently by using equation (2.17), especially for the special case of variable coefficients since different quantization steps are implied in equation (2.17). Consequently, it is possible to choose variable wordlength coefficients, such that a few coefficients are quantized to greater precision while others can be chosen with fewer bits and thus reduce the filter average wordlength. Since the filter implementation complexity is related to the adder cost and the multiplier size as described in the previous chapter, a reduction in the average wordlength leads to a reduction in the hardware cost for the filter realization.

2.3 Description of the Algorithm

The proposed method is based on the fact that the frequency response of a filter has different sensitivities to different coefficients depending on the filter response itself, i.e. the coefficient values. It is wise to vary the wordlength so that more bits will be used for the highly sensitive coefficients while fewer bits are used for the less sensitive coefficients. Our procedure of acquiring an ideal digital filter with the variable precision coefficients starts by using one of the conventional design methods to obtain a set of real coefficients. The order of the filter in the beginning can be determined by using the method as introduced in equation (1.2) based on the filter specifications and making sure that the filter response still lies within the required specification after the coefficients are quantized. If not, some adjustment of the initial choice of the filter order or coefficient precision must be made. As a result, a low space filter implementation can be realized without introducing additional inaccuracy into the filter response.

Figure 2-2 illustrates the algorithm for determining the optimal set of variable length coefficients. The procedure for obtaining the optimal variable coefficients starts by evaluating the sensitivity of each coefficient. First, each coefficient is set to the closest power of two. Then, the frequency response of the new quantized filter is evaluated. Equation (2.1) is used to determine the sensitivity of each coefficient.

After determining the sensitivities of the coefficients, the quantized coefficient d_n is determined considering the rounding error obtained by equation (2.17). If the frequency response does not meet the specification at the end of this step, one more nonzero bit is added to the most sensitive coefficient, i.e. the most sensitive coefficient is set to the closest sum or difference of two powers of two. Then, the frequency response of the modified filter and the sensitivity of the corresponding coefficient are reevaluated. The procedure is repeated until all the coefficients are set to their optimal one or more nonzero bits with the specification satisfied. Additional details of the algorithm are shown in Figure 2-2.

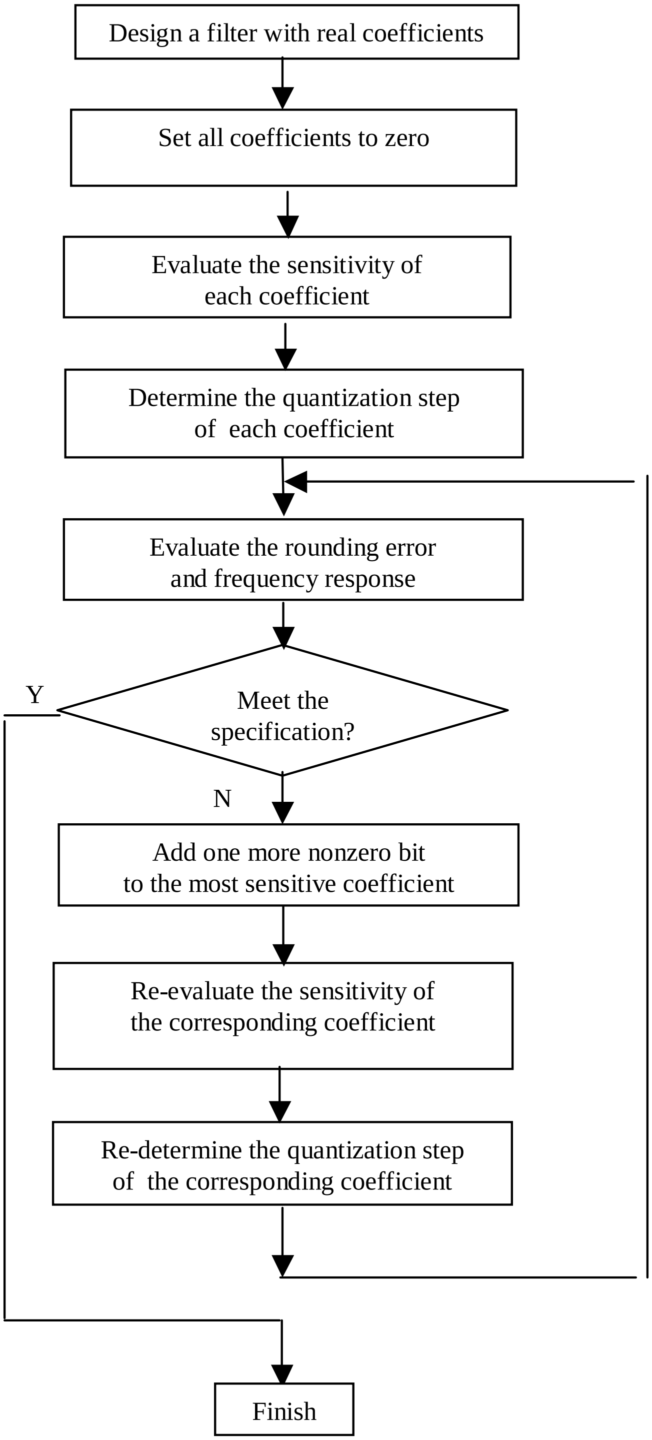

Design a filter with real coefficients
Set all coefficients to zero
Evaluate the sensitivity of each coefficient
Determine the quantization step of each coefficient
Evaluate the rounding error and frequency response
Meet the specification?
Y
N
Add one more nonzero bit to the most sensitive coefficient
Re-evaluate the sensitivity of the corresponding coefficient
Re-determine the quantization step of the corresponding coefficient
Finish

Figure 2-2 Algorithm for Variable Precision Coefficients [40]

2.4 Applications of the Proposed Algorithm to FIR Filter Designs

Example 2.1: A half-band FIR filter $H(z)$ with the passband and stopband edge frequency at $\omega_p T = 0.45$ $\omega_p T$ =o.45π and $\omega_s T$ =0.55π respectively, and the ripple in the passband not to exceed 0.1 db, while the stopband attenuation should be at least 50db. The required order is N-1=22.

The half-band filter was generated using the method given in [45]. The internal structure used to implement the fixed filter is the transposed structure.

The matched double precision coefficient filter is generated as shown in Table 2-1 as well as the resulting set of variable wordlength coefficients of the desired filter. The coefficients are initially quantized to 12 bits per coefficient. The frequency response of the corresponding fixed-point filter is shown in Figure 2-3. When the technique presented in this work is employed, which in turn reduces the average wordlength needed with the same performance and less noise introduced, the frequency response of the filter with a set of variable wordlength coefficients is observed (see Figure 2-3).

The results given in Table 2-2 show the comparison of VLSI implementations between the general FIR digital filter with 12 bits per coefficient and the variable wordlength coefficient FIR digital filter as implemented in the Actel device SXA72. The comparison indicates that the area being used on the device by using our proposed variable precision technique has been reduced by 16.52% over the uniform method. In addition, the critical path is reduced by the proposed method. It shows that variable precision method is feasible in practice and can improve the filter performance in terms of area, speed and power. There is no additional hardware required but the average coefficient length is reduced.

Table 2-1 The Results of Quantized FIR Filter & Wordlength of Each Tap [28]

Index of taps	Ideal value	Implementation value	Wordlength (bits)
h_0	-0.002315	-0.001953	11
h_1	0	0	0
h_2	0.005412	0.005859	11
h_3	0	0	0
h_4	-0.015866	-0.015625	8
h_5	0	0	0
h_6	0.038545	0.039063	9
h_7	0	0	0
h_8	-0.089258	-0.089844	10
h_9	0	0	0
h_{10}	0.312379	0.312500	6
h_{11}	0.500	0.5	3
h_{12}	0.312379	0.312500	6
h_{13}	0	0	0
h_{14}	-0.089258	-0.089844	10
h_{15}	0	0	0
h_{16}	0.038545	0.039063	9
h_{17}	00	0	0
h_{18}	-0.015866	-0.015625	8
h_{19}	0	0	0
h_{20}	0.005412	0.005859	11
h_{21}	0	0	0
h_{22}	-0.002315	-0.001953	11

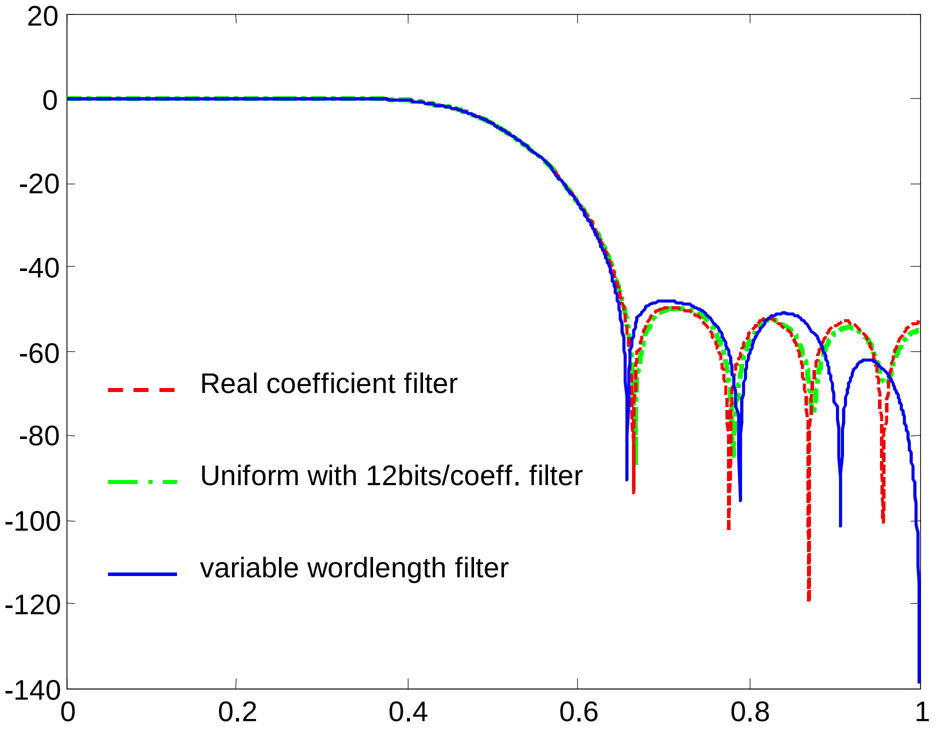


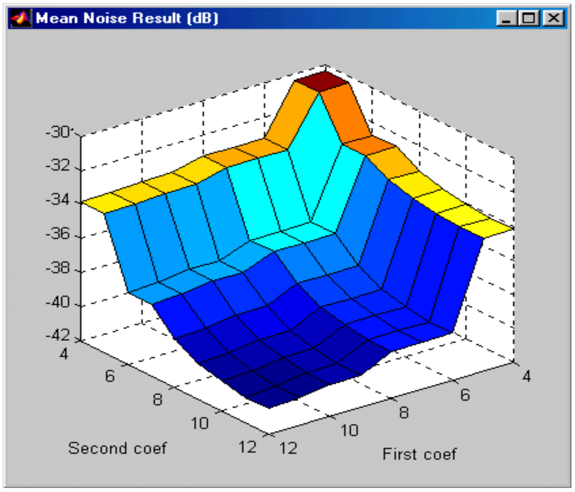
Figure 2-3 Example 2-1 Filter Design

Table 2-2 Comparison of VLSI Implementation

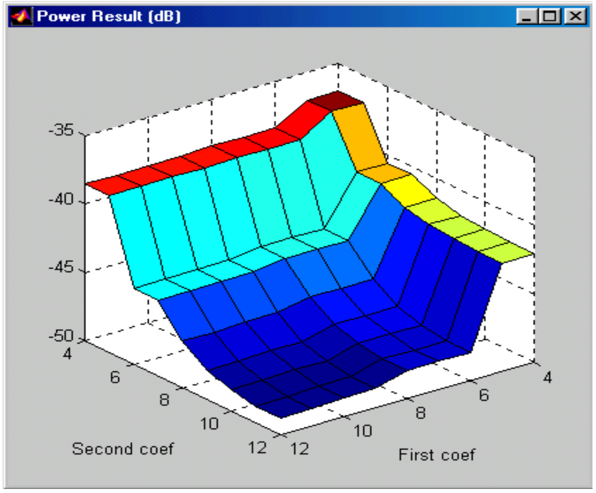
Measure	FIR_12	FIR_var
SEQ	1243 (61.78%)	1244 (61.83%)
COMB	2644 (65.71%)	2001 (49.73%)
LOGIC	3887 (64.40%)	3245 (53.76%)
t_pd	39.4	37.5

where SEQ refers to the sequential cell of the device, COMB is combinatorial cell., LOGIC means the cells being used (SEQ + COMB) and t_pd is the computation time of the critical path.

We still need to determine what the proper coefficient word-length should be. The algorithm developed by using coefficient sensitivity analysis helps us achieve this goal. Figure 2-4, generated by our software allows the user to consider the filter power noise when two correlated coefficients are changed. This also allows us to study the impact when more than one coefficient is varied. Figure 2-4 (a) shows the mean noise of the filter output. The vertical axes of both of (a) and (b) indicate the noise in db when wordlength of the coefficients is changed. In this example, we consider the effect of the wordlength of b2 and b4 on filter performance. Figure 2-4 (a) shows that only if the wordlength of b2 is greater than 6 bits and the wordlength of b4 is greater than 6 bits, will the mean noise output be less than -40 db (our target noise level). Thus, if we choose at least 6 bits for these two coefficients, then the filter performance will not be impacted significantly. Figure 2-4(b) shows the noise variance output if the wordlength of the coefficients is changed. In this case, the graph shows that if the wordlength is greater than



(a) Mean Noise for Variations in 2 Selected Coefficients



(b) Power Noise Analysis for Variations in 2 Selected Coefficients

Figure 2-4 Output Noise Analysis with the Combination of two Coefficients Considered

8 bits, the performance of the system will be acceptable. Combining the results of Figure 2-4 (a) and Figure 2-4 (b), we determine that the wordlength of the coefficients should be greater than 8 bits.

***Example 2.2**:* To illustrate the advantage of the proposed variable coefficient algorithm based on sensitivity analysis over the other methods, the filter provided by Nielsen [32] with the following specification is adopted: the filter order N, is 67, the passband frequency is $0.2 * f_s$, the ripple on the passband is no more than 0.1db, the stopband frequency is $0.3 * f_s$, and the attenuation of the stopband should be below 90db.

The double precision filter, $h(n)$, is generated by the Parks-McClellan method as given in the Matlab "remez" (or "firpm") method. The second column of Table 2-3 shows the values generated when each coefficient is set to its closest power of two. The third column gives the initial sensitivity of each coefficient, which is the basis of determining the variable precision coefficients. The results of the Nielsen algorithm are given in the fourth column, and the results from the proposed sensitivity algorithm are given in the last column for comparison. Figure 2-5 shows the performance of the corresponding frequency responses of the quantized digital filters with variable length coefficients produced using both Nielsen's algorithm and our proposed algorithm, as well as the frequency response of the filter with double precision coefficients.

Table 2-3 The Results of Proposed Algorithm as well as the Results of Nielsen Method

Real Coefficient	Closest power of two	Sensitivity	Nielsen method coefficients representation	Variable precision coefficients representation
h(34)=32765.0156	2^{15}	4.0251e-06	$2^{15}-2^{1}$	2^{15}
h(35,33)=21386.3712	2^{14}	0.0067	$2^{14}+2^{12}+2^{10}-2^{7}+2^{4}-2^{3}-2^{0}$	$2^{14}+2^{12}+2^{10}-2^{7}+2^{3}+2^{1}$
h(36,32)=992.9455	2^{10}	4.1883e-05	$2^{10}-2^{5}+2^{0}$	$2^{10}-2^{5}$
h(37,31)=-6826.4739	-2^{13}	0.0018	$-2^{13}+2^{11}-2^{9}-2^{7}-2^{5}-2^{3}-2^{1}$	$-2^{13}+2^{11}-2^{9}-2^{7}-2^{5}-2^{3}-2^{1}$
h(38,30)=-941.9129	-2^{10}	1.1071e-04	$-2^{10}+2^{6}+2^{4}+2^{0}$	$-2^{10}+2^{6}+2^{4}+2^{1}$
h(39,29)=3753.0633	2^{12}	4.6252e-04	$2^{12}-2^{9}+2^{7}+2^{5}+2^{3}$	$2^{12}-2^{9}+2^{7}+2^{5}+2^{3}$
h(40,28)=862.0648	2^{10}	2.1840e-04	$2^{10}-2^{7}-2^{5}-2^{1}$	$2^{10}-2^{7}-2^{5}-2^{1}$
h(41,27)=-2346.7999	-2^{11}	4.0299e-04	$-2^{10}-2^{8}-2^{5}-2^{3}-2^{1}$	$-2^{10}-2^{8}-2^{5}-2^{3}-2^{1}$
h(42,26)=-760.4611	-2^{9}	3.3510e-04	$-2^{10}+2^{8}+2^{3}$	$-2^{10}+2^{8}+2^{3}$
h(43,25)=1522.7395	2^{11}	6.7265e-04	$2^{11}-2^{9}-2^{4}+2^{1}$	$2^{11}-2^{9}-2^{4}+2^{1}$
h(44,24)=645.6166	2^{9}	1.8021e-04	$2^{9}+2^{7}+2^{2}$	$2^{9}+2^{7}+2^{3}-2^{1}$
h(45,23)=-986.6937	-2^{10}	5.0315e-05	$-2^{10}+2^{5}+2^{2}$	$-2^{10}+2^{5}+2^{2}$
h(46,22)=-526.4166	-2^{9}	1.9444e-05	$-2^{9}-2^{4}+2^{1}$	$-2^{9}-2^{4}+2^{1}$
h(47,21)=624.1586	2^{9}	1.5127e-04	$2^{9}+2^{7}-2^{4}$	$2^{9}+2^{7}-2^{4}$
h(48,20)=411.0701	2^{9}	1.3612e-04	$2^{9}-2^{7}+2^{5}-2^{2}-2^{0}$	$2^{9}-2^{7}+2^{5}-2^{2}-2^{0}$
h(49,19)=-378.5124	-2^{9}	1.6523e-04	$-2^{9}+2^{7}+2^{2}+2^{0}$	$-2^{9}+2^{7}+2^{2}+2^{0}$
h(50,18)=-306.2521	-2^{6}	6.7775e-05	$-2^{8}-2^{6}+2^{4}-2^{2}+2^{0}$	$-2^{8}-2^{6}+2^{4}-2^{1}$
h(51,17)=215.8401	2^{8}	5.4164e-05	$2^{8}-2^{5}-2^{3}-2^{0}$	$2^{8}-2^{5}-2^{3}$
h(52,16)=216.5513	2^{8}	5.3204e-05	$2^{8}-2^{5}-2^{3}$	$2^{8}-2^{5}-2^{3}$
h(53,15)=-112.6784	-2^{7}	2.0664e-05	$-2^{7}+2^{4}-2^{0}$	$-2^{7}+2^{5}$
h(54,14)=-144.2876	-2^{7}	2.1967e-05	$-2^{7}-2^{4}-2^{0}$	$-2^{7}-2^{5}$
h(55,13)=51.3508	2^{6}	1.7060e-05	$2^{6}-2^{4}+2^{1}$	$2^{6}-2^{4}+2^{2}$
h(56,12)=89.6510	2^{7}	3.4595e-05	$2^{7}-2^{5}-2^{3}$	$2^{7}-2^{5}-2^{0}+2^{1}$
h(57,11)=-18.1667	-2^{4}	2.9222e-06	$-2^{4}-2^{1}-2^{0}$	$-2^{4}-2^{1}$
h(58,10)=-51.1279	-2^{6}	1.7361e-05	$-2^{6}+2^{4}-2^{4}+2^{0}$	$-2^{6}+2^{4}-2^{2}$
h(59,9)=2.6714	2^{1}	9.0552e-07	$2^{2}-2^{0}$	2^{1}
h(60,8)=26.0694	2^{5}	7.9986e-06	$2^{5}-2^{3}+2^{1}$	$2^{5}-2^{3}+2^{1}$
h(61,7)=2.7570	2^{1}	1.0210e-06	$2^{2}-2^{0}$	2^{1}
h(62,6)=-11.3053	-2^{3}	4.4579e-06	$-2^{3}-2^{1}$	$-2^{4}+2^{2}$
h(63,5)=-3.2459	-2^{2}	1.0171e-06	-2^{0}	-2^{2}
h(64,4)=3.8829	2^{2}	1.5793e-07	$2^{3}-2^{1}$	2^{2}
h(65,3)=2.4990	2^{1}	6.7300e-07	2^{2}	2^{1}
h(66,2)=0.0785	0	2.1579e-08	2^{1}	0
h(67,1)=-0.2136	0	4.9093e-08	0	0

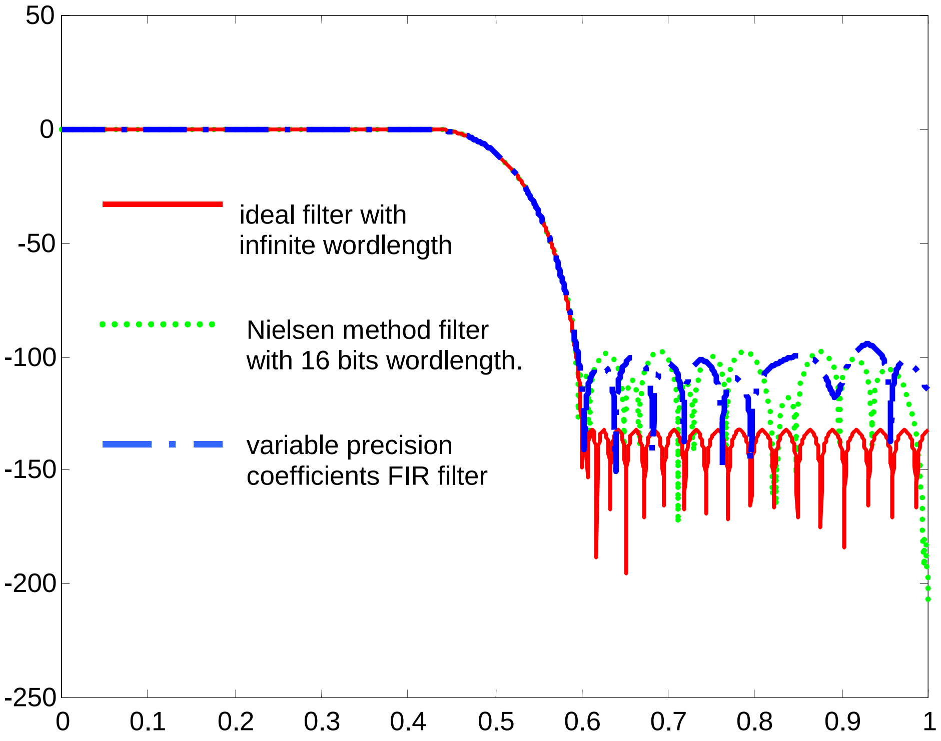


Figure 2-5 The FIR filter Frequency Responses

From Table 2-3 and Figure 2-5, it is clear that the proposed variable precision method produces the same performance as Nielsen's method, while the variable precision method requires fewer adders. For a multiplierless filter implementation, the number of the adders required is one of the important factors to measure the complexity of the digital filter [10][15]. Compared with Nielsen's example presented here, the area required by the variable precision method is reduced by 12.269% over the Nielsen's method in terms of the number of adders required, respectively.

2.5 Improvement on CSD Application

For more efficient hardware usage, a general method for performing multiplication by a constant value is by using a sequence of shifts and adds. To this end, the canonical signed digit (CSD) representation can be used to reduce the number of nonzero digits, where a coefficient is represented with a summation and subtraction of power of terms. Applying this approach to FIR filters, it is possible to greatly reduce the number of additions needed, and thus reduce the area consumption [3][8][11]. Since CSD format often requires fewer nonzero bits, CSD representation has the advantage of decreasing the number of additions/subtractions needed.

2.5.1 CSD Representation

The definition of the CSD form [46] is given by:

$$x = \sum_{r=0}^{N} a_r 2^{-r}, \qquad (2.18)$$

where $a_r \in \{-1,\ 0, +1\}$ and no two consecutive digits are nonzero, i.e.

$$a_r \cdot a_{r+1} = 0, \quad 0 \le r \le N \ .$$

Because no 2 consecutive digits in a CSD number are nonzero, CSD numbers contain the minimum possible number of nonzero bits, which is about $N/3$ on average as opposed to $N/2$ in the usual 2's complement numbers. This reduced number of nonzero bits is the reason that CSD is widely applied in digital filter implementation [47]. If each coefficient of equation (1.1) is expressed in CSD format as $h_k = h_{k,M-1}h_{k,M-2}...h_{k,1}h_{k,0}$, then the filter equation becomes

$$y(n) = \sum_{k=0}^{N-1}\sum_{i=0}^{M-1} h_{k,i}x(n-k) >> (M-i)\,, \tag{2.19}$$

where, $x(n-k) >> (M-i)$ denotes shifting the input item to the right for $(M-i)$ bits.

To show how the filter may be implemented using a CSD constant multiplier, for a choice of filter coefficients as an example $h_0 = 0.1000100\bar{1}, \quad h_1 = 0.0\bar{1}00010\bar{1}$, then the filter can be implemented as

$$\begin{aligned} y(n) &= x(n)h_0 + x(n-1)h_1 \\ &= x(n) >> 1 + x(n) >> 5 - x(n) >> 8 \\ &\quad - x(n-1) >> 2 + x(n-1) >> 6 - x(n-1) >> 8 \end{aligned} \quad ,$$

which requires 5 additions.

The conventional filter can be expressed as $h_0 = 0.10000111, \quad h_1 = 1.11000011$, and the filter is implemented in the way

$$\begin{aligned} y(n) &= x(n)h_0 + x(n-1)h_1 \\ &= x(n) >> 1 + x(n) >> 6 + x(n) >> 7 + x(n) >> 8 \\ &\quad + x(n-1) >> 1 + x(n-1) >> 2 + x(n-1) >> 7 + x(n-1) >> 8 \end{aligned} \quad .$$

Compared with conventional implementation, which requires 7 additions, the operations are reduced by 28.57%.

2.5.2 Distribution of CSD Quantization

However, the quantization error for the fixed number filter is larger than that of the 2'complement number since the set of numbers represented by a CSD code with a fixed number of nonzero digits is very non-uniformly distributed [8]. This non-uniform distribution is shown as Figure 2-6 for the case of a 6 digits CSD code as well as a 7 digits CSD code. Both of the cases have no more 2 nonzero digits, that is their fixed number of quantization bits is 2 nonzero bits.

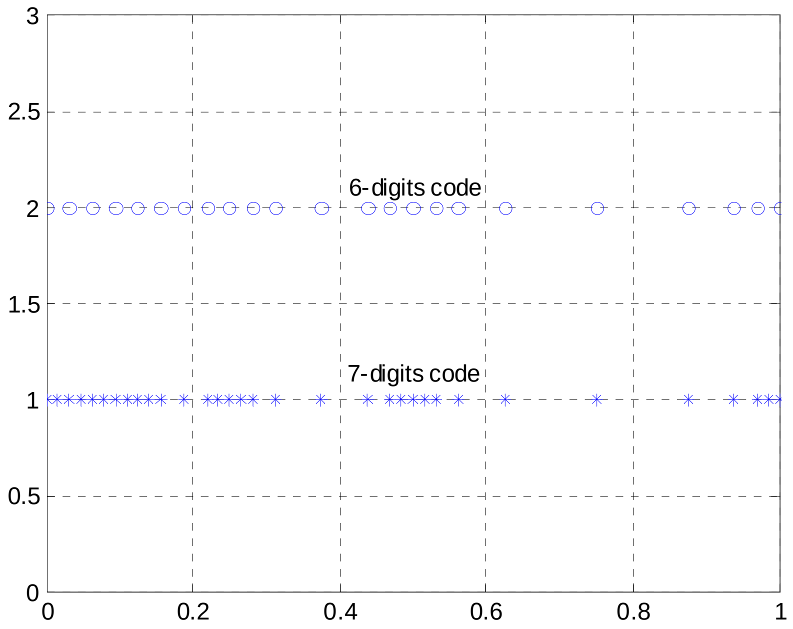


Figure 2-6 Distributon of CSD Coefficient Set for 6 Digits Code as well as 7 Digits Code with 2 Nonzero Bits

2.5.3 Sensitivity Method on CSD Improvement

Samueli [8] proposed a method that adds one additional nonzero digit to the CSD representation to the impulse response coefficients whose magnitudes exceed 0.5. This design is based on the observation that when rounding a set of filter coefficients to the nearest CSD code, the magnitude of the worst quantization error always occurs for the larger valued coefficients. However, this method does not consider the sensitivity of the coefficient with respect to the frequency response. In some cases, although the magnitude of some coefficient exceeds 0.5, the coefficient has little effect on the filter performance. Therefore, the method of adding an additional nonzero bit to the coefficient cannot provide better filter performance in these instances.

In other words, simply adding an additional nonzero bit to the coefficients whose magnitude exceeds 0.5 is not the best way to reduce the filter quantization error. From our experiments, we have seen that the magnitude of the worse–case quantization error usually occurs for the most sensitive coefficient in the CSD code. Hence, we propose a strategy to reduce the quantization error without increasing the complexity as follows:

One additional nonzero bit in the CSD representation is allocated to the most sensitive coefficient until the specification is met. The method to calculate the coefficient sensitivity is the same as we proposed in this chapter in equation (2.1).

Example 2.3: Design an FIR filter with normalized passband and stopband edge frequencies of 0.15* f_s and 0.25* f_s, respectively. The desired filter passband and stopband ripples are $\delta_p = \delta_s = 0.005\,(-46dB)$.

Initially, we quantized the coefficients to the set of numbers with wordlength 12 bits. Next, we apply the proposed algorithm to generate the optimized CSD coefficients shown in Table 2-4. Samueli's CSD coefficients are also shown in Table 2-4 for comparison. The normalized frequency responses of the quantized coefiicients are illustrated in Figure 2-7. A total of 44 adders are needed to realize the proposed CSD digit filter, which is the same as Samueli's method. However, the stopband attenuation of the proposed filter is 47dB, which is better than the requirement. For comparison, the stopband attenuation of Samueli's is 43.8dB, which is 2.2dB less than required. Therefore, with the same area requirement, the performance of the filter implemented is improved using our proposed method.

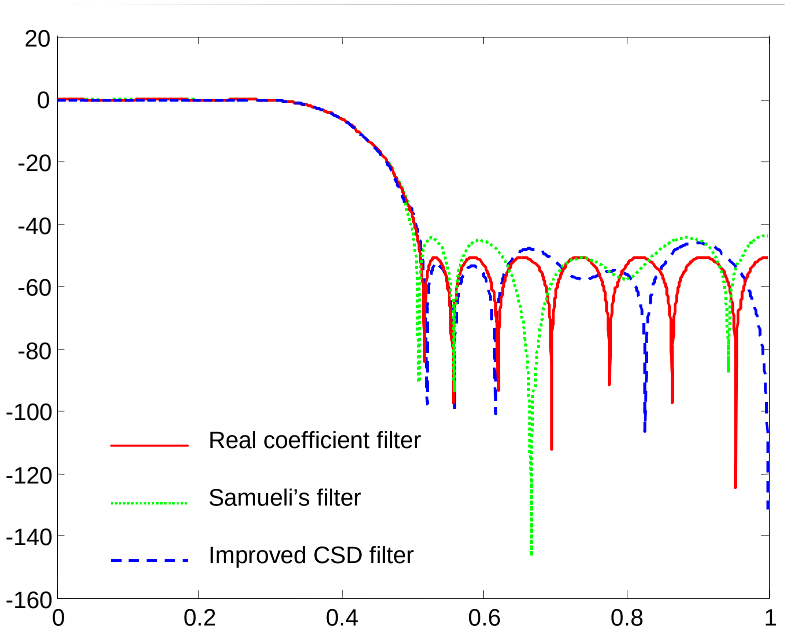


Figure 2-7 Frequency Response of CSD Coefficients FIR Filters

Table 2-4 Comparision of quantized CSD Coefficients

Coefficients	Samueli's CSD (before normalized)	Improved variable CSD
h(0) , h(24)	2^{-8}	2^{-9}
h(1) , h(23)	$2^{-7}+2^{-8}$	$2^{-8}+2^{-10}$
h(2) , h(22)	-2^{-8}	-2^{-10}
h(3) , h(21)	-2^{-5}	$-2^{-6}+2^{-8}$
h(4) , h(20)	$-2^{-5}+2^{-8}$	$-2^{-7}-2^{-9}$
h(5) , h(19)	$2^{-5}+2^{-7}$	2^{-6}
h(6) , h(18)	$2^{-4}+2^{-6}$	2^{-5}
h(7) , h(17)	-2^{-8}	-2^{-9}
h(8) , h(16)	$-2^{-3}-2^{-5}$	-2^{-4}
h(9) , h(15)	$-2^{-3}-2^{-7}$	$-2^{-4}+2^{-7}$
h(10) , h(14)	$2^{-2}-2^{-5}$	$2^{-3}-2^{-5}$
h(11) , h(13)	$2^{-1}+2^{-2}-2^{-5}$	$2^{-2}+2^{-4}-2^{-6}$
h(12)	$2^{0}-2^{-5}-2^{-7}$	$2^{0}-2^{-2}+2^{-4}-2^{-6}$

2.6 Concluding Remarks

An algorithm is presented for selecting variable precision coefficients for FIR filters that produces a reduced space implementation with no degradation in frequency response. The method in this chapter is based on the fact that the frequency response of a filter has different sensitivities to different coefficients. The novelty of the algorithm is that it is able to reduce the number of bits required by using the variable precision of the quantized coefficients to meet the specification with low space. The examples in this chapter show that using variable precision to exploit redundancy across the coefficients results in significant reductions in complexity and area over the uniform wordlength method and over other known variable precision methods. The use of variable precision coefficients has opened an exciting implementation possibility that allows solutions to trade-off between performance and area just as wordlength reduction, which can result in great reduction in the number of additions and storage blocks in the chip area. The proposed method is highly appropriate for the design of multiplierless filters and VLSI or FPGA implementations.

Chapter 3 Scaling Algorithm for Pre-quantization

Scaling is the process of adjusting the real filter coefficients in order to constrain the coefficient values to an appropriate range more suitable for finite precision arithmetic. As addressed in [1][8][20][28], the resulting implemented filter has significantly better frequency shaping characteristics if the filter coefficients are scaled before the quantization process is performed. This improved and enhanced performance is attributed to the multiplication of an appropriate scale factor (SF), which can optimize the coefficient values and minimize the error noise.

3.1 Effect of Scaling Filter Coefficients

The incorporation of a scale factor has a significant affect on the coefficient optimization process when the coefficients are represented by CSD code. Scaling the filter coefficients prior to rounding them can always reduce the quantization error significantly and improve the frequency response. The benefit of scaling is shown in the following example.

Example 3.1: Given an ideal filter with coefficients [0.26, 0.131,0.087, 0.011], consider a uniform wordlength of 7 bits, in which 1 bit is for the integer part and 6 bits represents the fractional part. If the coefficients are quantized directly, the quantized filer coefficients are [0.25, 0.125, 0.0781, 0]. The worst case of difference between the quantized and unquantized coefficients is less than $1/64$. The quantization error is [0.01, 0.006,0.0089, 0.011].

If a scaling factor *SF*=1.923 (i.e. 0.5/0.26), which makes the absolute value of the largest coeffcient to be 0.5, is applied prior to quantization, then the scaled filter coefficients are modified as to be [0.5, 0.2519,0.1673,0.0212], which can be quantized in the same wordlength (7 bits) digits by [0.5,0.25, 0.1563,0.0156]. In this case, the difference between the quantized and unquantized coefficients is also less than $1/64$. However, the quantization error is reduced to

$$\frac{1}{SF}[0,\ 0.0019,\ 0.0110,\ 0.0056]=[0,\ 0.0010,\ 0.0057,\ 0.0029]$$

Therefore, the scaling operation minimizes the quantization error and improves the performance of the fixed-point filter.

3.2 An Improved Scaling Algorithm

Our proposed method for selecting the scale factor is to account for the correlation between the coefficient quantization error and the frequency response errors. We want to select the scale factor so that it has the shortest length (or less nonzero bits) and results in the minimum value of output quantization error. We constrain the scale factor so that after scaling, the magnitude of the largest coefficient is between 0.5 and 1. Our proposed iterative procedure to choose the preferred *SF* is as follows:

Step 1. The real filter is initially normalized so that the magnitude of the largest coefficient is equal to 1, and the initial scaled factor is chosen as,

$$SF=\frac{1}{\max\left\{abs\left(\text{ideal filter coefficients}\right)\right\}}$$

Step 2. For the scaled filter, choose a maximum wordlength ($W\_\max$) of the coefficients such that the peak ripple of the fixed-point filter is strictly less than the ripple of the filter specification.

Step 3. Set the appropriate wordlength of each coefficient using the following procedure (where *K* is the order of the filter)

```
For tap = 0 to $K$ {
    For $w = 1$ to $W\_\max$ {
        Quantize the coefficient to $w$ bits;
        If the quantization error < $2^{-(W\_\max+1)}$ then
            Set the coefficient wordlength to $w$ and break;
    }
}
```

Step 4. After all of the coefficients are quantized, a re-scaled factor (RSF) is used to properly re-modify the magnitude of the resulting fixed filter coefficients. The value of the RSF is determined as follows:

$$RSF = \frac{\max\left\{abs\left(\text{ideal filter coefficients}\right)\right\}}{SF}$$

Step 5. Estimate the output error of the filter.

Step 6. Change the *SF* with the step size of 2^{-k}, where *k* is an arbitrarily chosen integer, and repeat steps 2-6. Choose the appropriate *SF* such that the output error of the filter is minimum.

The flow chart of the above scaling procedure is shown as Figure 3-1.

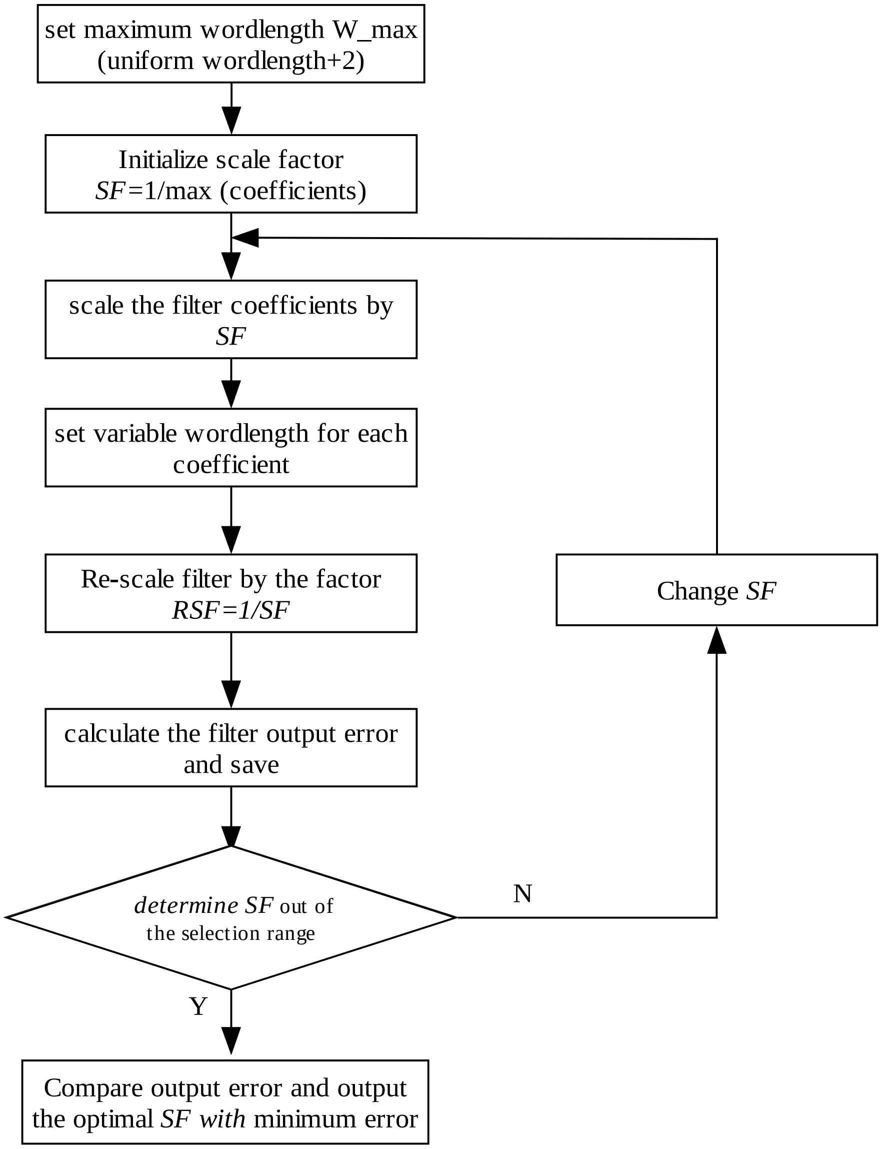

set maximum wordlength W_max
(uniform wordlength+2)
Initialize scale factor
SF=1/max (coefficients)
scale the filter coefficients by
SF
set variable wordlength for each
coefficient
Re-scale filter by the factor
RSF=1/SF
calculate the filter output error
and save
determine SF out of
the selection range
N
Change *SF*
Y
Compare output error and output
the optimal *SF with* minimum error

Figure 3-1 Scaling procedure

3.3 Numerical Example

To show the effect of the proposed scaling algorithm, we consider example 2.1, that is, the filter with coefficients

$$\begin{aligned} h(n) = [&-0.0023\ 0\ 0.0054\ 0\ -0.0159\ 0\ 0.0385\ 0\ -0.0893 \\ &0\ 0.3124\ 0.5000\ 0.3124\ 0\ -0.0893\ 0\ 0.0385\ 0\ -0.0159\ 0\ 0.0054\ 0\ -0.0023] \end{aligned}$$

The frequency response of ideal filter and the un-scaled quantized filter are shown in Figure 3-2. Employing the technique presented in this chapter, the filter is quantized with scaled variable precision coefficients. The scaling factor, *SF*, is determined by $(SF)=1/\max(abs(h))=2$. The scaled coefficients are $hs = SF * h = 2h$, and the maximum wordlength, *MW*, is 10 bits. The resulting filter and its variable precision coefficients are shown in Table 3-1. To recover the actual desired filter, the quantized FIR filter can be multiplied by a rescaling factor, $RSF = 0.5$. Then, the FIR filter will be implemented $h = hs * RSF = hs/2$.

The improvement in this case shows not only that the quantization error of the scaled filter is reduced to some extent by the scaling process, but also that the wordlength of the coefficients can be reduced. Consequently, the scaling method used to preprocess the designed filter will reduce the space required and decrease the quantization error as well.

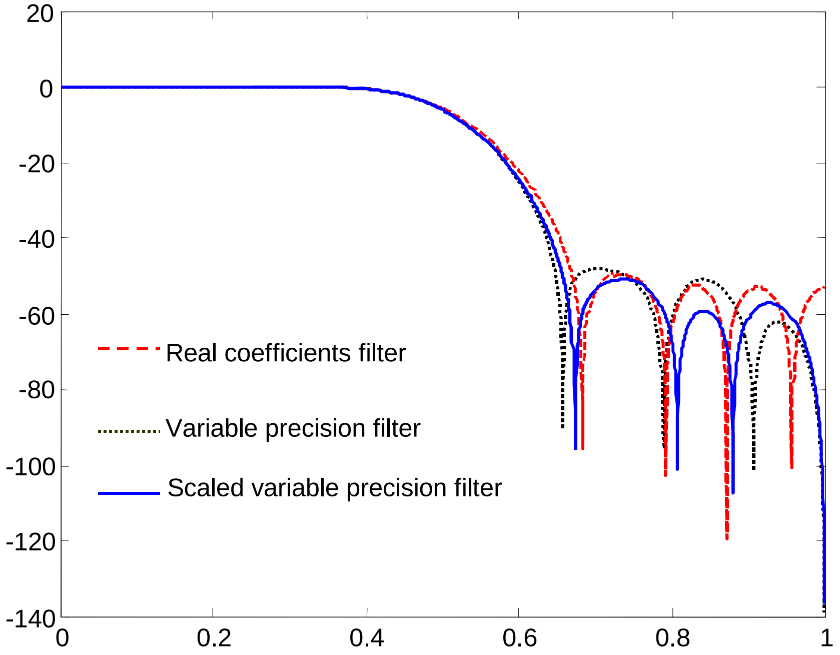


Figure 3-2 The Performance of Scaled Filter as well as that of the Unscaled Filters

Table 3-1 The Resulting Scaled Filter and its Coefficients

Index of taps	Ideal coefficients	Quantized coefficients scaled by 2	Wordlength (bits)
h_0	-0.0023	-0.00390625	10
h_1	0	0	0
h_2	0.0054	0.01171875	10
h_3	0	0	0
h_4	-0.0159	-0.03125	7
h_5	0	0	0
h_6	0.0385	0.078125	8
h_7	0	0	0
h_8	-0.0893	-0.1796875	9
h_9	0	0	0
h_{10}	0.3124	0.625	5
h_{11}	0.500	1.0	2
h_{12}	0.3124	0.625	5
h_{13}	0	0	0
h_{14}	-0.0893	-0.1796875	9
h_{15}	0	0	0
h_{16}	0.0385	0.078125	8
h_{17}	0	0	0
h_{18}	-0.0159	-0.03125	7
h_{19}	0	0	0
h_{20}	0.0054	0.01171875	10
h_{21}	0	0	0
h_{22}	-0.0023	-0.00390625	10

Chapter 4 Improved Prefilter-Equalizer Filter

The factors that impact the performance in terms of the amount of hardware circuitry and resources required are the number of adders and multipliers, which is highly related to the order of the filer. The required order for linear phase finite impulse response (FIR) filters can be estimated by equation (1.2). From the equation, we see that the order of the FIR filter is inversely proportional to the width of the transition band and highly related to the filter ripple. This indicates that a filter with sharp transition band and huge attenuation requires much more complexity in implementation. In the past few years, several alternative FIR filter designs have been proposed to improve the performance of the filter and reduce the number of arithmetic operations. One attractive way to reduce the hardware cost is to reduce the order of the FIR filter by using a relatively simple structure [29][30] based on a prefilter. In this chapter, an improved efficient multiplierless structure of FIR filter synthesis is proposed based on the prefilter-equalizer structure, which allows arbitrary bandwidth and great area reduction.

Section 4.1 introduces the prefilter equalizer structure used for space efficient FIR design based on RRS prefilter. The effect of the coefficient quantization on filter frequency response is discussed in Section 4.2. Compared with the conventional structure, the superior performance inreducing area consumption is also demonstrated in this section. Section 4.3 explores a new prefilter structure to make it possible to extend the simple prefilter to wideband FIR filter design. Concluding remarks are also offered in Section 4.4.

4.1 The Prefilter Equalizer Structure

The scheme of a prefilter-equalizer was explored in [29]. Unlike the direct structure, the realization is a cascade connection of a prefilter with reasonable frequency response and a relative simple structure, and the equalizer has of reduced order to achieve the low cost implementation. The technique can be outlined as follows: Given a set of specifications of the frequency response in terms of cutoff frequencies and attenuation tolerances, the transfer function $H(z)$ is obtained as a cascade of two transfer functions $H_1(z)$ and $H_2(z)$. The function $H_1(z)$, called the prefilter, is extremely simple to implement, requiring very few additions and multiplications. This prefilter, $H_1(z)$, provides reasonable stopband attenuation but has a poor pass-band response. The filter $H_2(z)$, called the equalizer, compensates for this problem, leading to an overall filter $H(z)$ that meets all the specifications. The order of $H_2(z)$ is much lower than that of a filter designed directly from the initial specification. This resulting order reduction is partially obtained from the prefilter because $H_1(z)$ has already contributed some attenuation in the stopband. Therefore, the overall implementation is simpler than that of a direct conventional approach.

A very simple and effective prefilter is the recursive running sum (RRS) filter [29], which requires only two adders and no multipliers at all, regardless of the filter length.

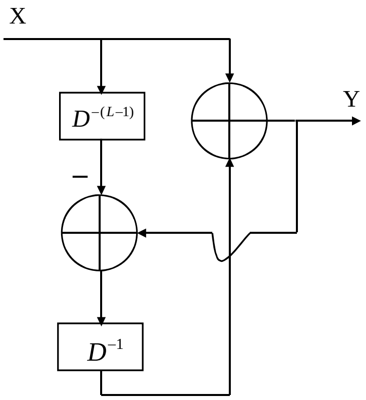


Figure 4-1 Implementation of RRS Structure

The implementation of RRS is shown in Figure. 4-1. The frequency response of an RRS filer is given by

$$H(e^{jw}) = \frac{\sin(\omega L/2)}{\sin(\omega/2)} e^{-j\omega(L-1)/2}, \tag{4.1}$$

where L is the length of the filter. Substituting $z = e^{j\omega}$, the equivalent transfer function can be written as:

$$\begin{aligned} H(z) &= 1 + z^{-1} + \ldots + z^{-(L-1)} \\ &= \frac{1 - z^{-L}}{1 - z^{-1}} \end{aligned} \tag{4.2}$$

Thus, we see that the RRS is just an FIR digital filter with unity coefficients. However, it yields a relatively simple lowpass response and relieves some of the burden from the equalizer, especially from making a sharp transition and by reducing the attenuation required in the stopband. Thereby, it will minimize the order of equalizer.

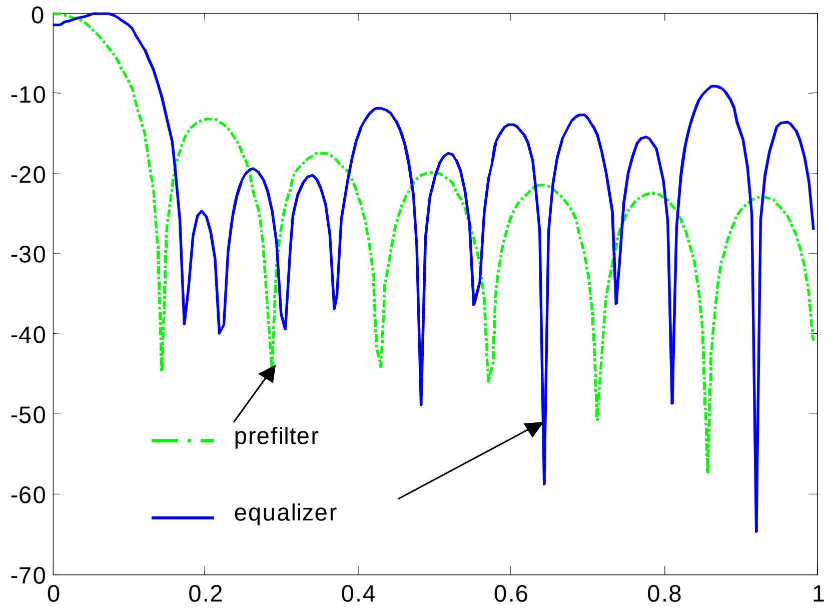
In order to demonstrate the advantage of such structure implementation, a specific simulation is provided here.

Example 4.1: Consider an FIR filter design with the following filter requirement: the edge of the passband and the stopband are at 0.042π and 0.14π, and the ripple of the passband and the stopband are $0.2dB$ and $35dB$, respectively.

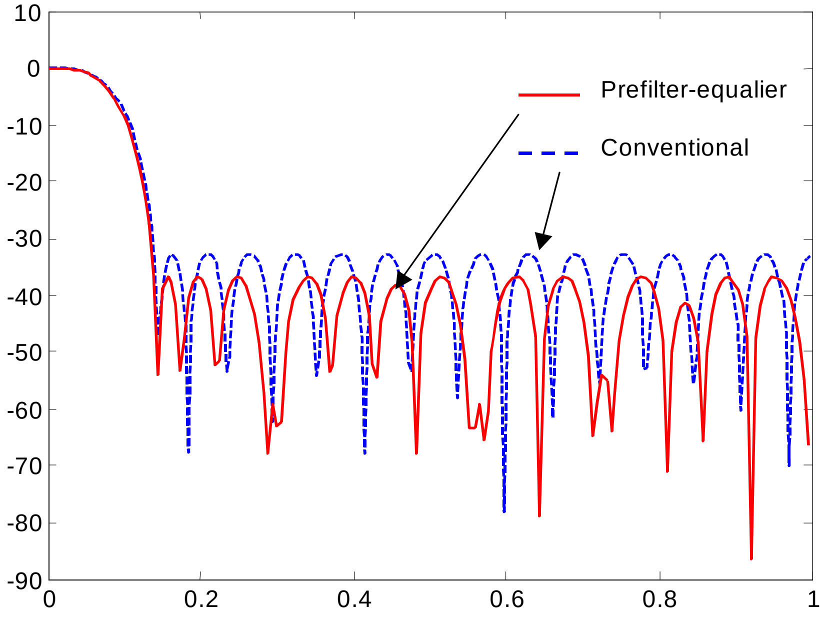
Figure 4-2(a) shows the magnitude response of the prefilter and equalizer. From the figure, we can easily notice that not only a wider transition band is allowed for the equalizer, but also the stopband attenuation requirements on the equalizer are relaxed. As a result, Figure 4-2(b) provides the performance of the prefilter and equalizer cascade as well as the comparison to a conventional filter designed to meet the same specification. The filter performance of the cascade structrue is improved significantly. A summary of the hardware requirements for the prefilter and equalizer structure filter and the conventional filter is given in Table 4-1. The required order of the conventional filter is 35, much longer than that of the equalizer, 23. In this case, it is assumed that both the conventional filter and the equalizer are implemented using a standard linear phase FIR filter. The prefilter equalizer FIR filter uses 5.7 percent more delay but 40 percent fewer adders and 50 percent fewer multipliers than the conventional filter. The prefilter and equalizer structure filter design has significant saving of the hardware.

Table 4-1 The Summary of the Hardware Requirement for Different Structure Filters

	Prefilter	Equalizer	Total of Cascade	Conventional filter
Delays	14	23	37	35
Adders	2	23	25	35
Multipliers	0	12	12	18



(a) The individual magnitude responses of the prefilter and equalizer



(b) The result of cascading the prefilter and equalizer, as well as the result of direct conventional filter design

Figure 4-2 RRS prefilter based FIR filter

4.2 Quantization Error of the Prefilter-Equalizer Structure

The realization of the digital FIR filter involves the quantization of signals and the coefficients. Thus, the coefficient quantization error incurred in the prefilter-equalizer cascade is considered here.

Let $h(n)$ and $\tilde{h}(n)$ denote the infinite precision and finite precision coefficients, respectively. Then, the quantization error $e(n)$ is given by:

$$e(n) = h(n) - \tilde{h}(n)\,. \tag{4.3}$$

$E(e^{j\omega})$ represents the frequency responses of the sequences defined above and is given as:

$$E(e^{j\omega}) = H(e^{j\omega}) - \tilde{H}(e^{j\omega})\,. \tag{4.4}$$

Assuming that the $e(n)$ are uncorrelated, then $E(e^{j\omega})$ should be an approximately noise-like spectrum.

For a prefilter equalizer cascade filter, $P(e^{j\omega})$ and $Q(e^{j\omega})$ denote the frequency responses of the real coefficients of the prefilter and the equalizer; the cascade filter is denoted as:

$$H(e^{j\omega}) = P(e^{j\omega})Q(e^{j\omega}) \tag{4.5}$$

The frequency response corresponding to the quantized coefficients are denoted by $\hat{H}(e^{j\omega})$, $\tilde{P}(e^{j\omega})$ and $\tilde{Q}(e^{j\omega})$, respectively. The quantization error for the equalizer is defined as $E_q(e^{j\omega})$. Therefore, we have

$$\hat{Q}(e^{j\omega}) = Q(e^{j\omega}) + E_q(e^{j\omega})\,. \tag{4.6}$$

Assume that we choose the prefilter so that its performance is immune to coefficient quantization. For example, using the RRS structure prefilter, we have $P(e^{j\omega}) = \hat{P}(e^{j\omega})$. After the coefficients are quantized, the transfer function of the prefilter and equalizer cascade filter can be expressed as:

$$\begin{aligned}\tilde{H}(e^{j\omega}) &= \tilde{P}(e^{j\omega})\tilde{Q}(e^{j\omega}) \\ &= P(e^{j\omega})(Q(e^{j\omega}) + E_q(e^{j\omega})) \\ &= H(e^{j\omega}) + P(e^{j\omega})E_q(e^{j\omega})\end{aligned} \tag{4.7}$$

From equation (4.7), we see that the prefilter attenuates the quantization error of the equalizer in the stopband. Based on this observation, we conclude that the prefilter equalizer at the stopband should be less sensitive to the quantization of the filter coefficients. Figure 4-3 shows the block diagrams representing the above relationship for the conventional filter and the prefilter equalizer cascade filter.

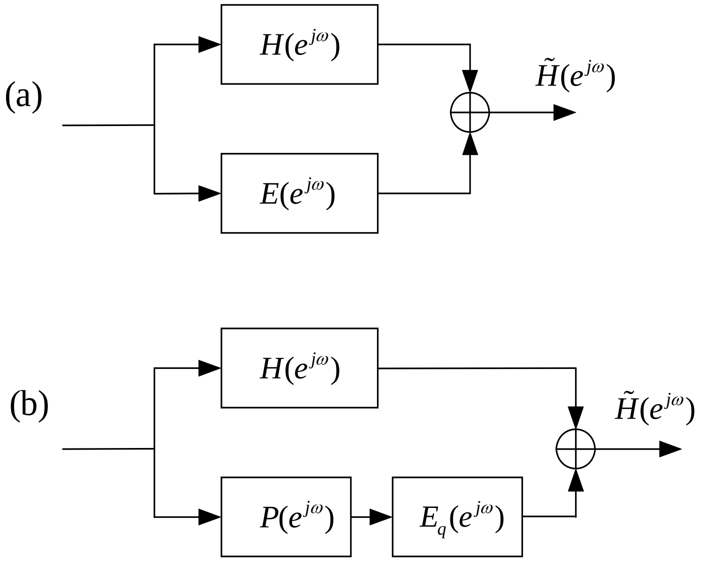
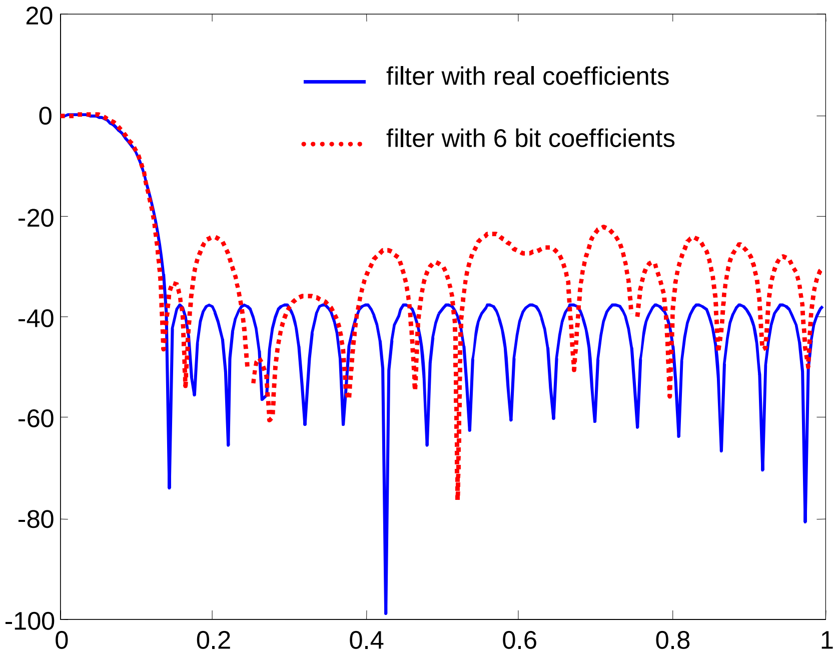
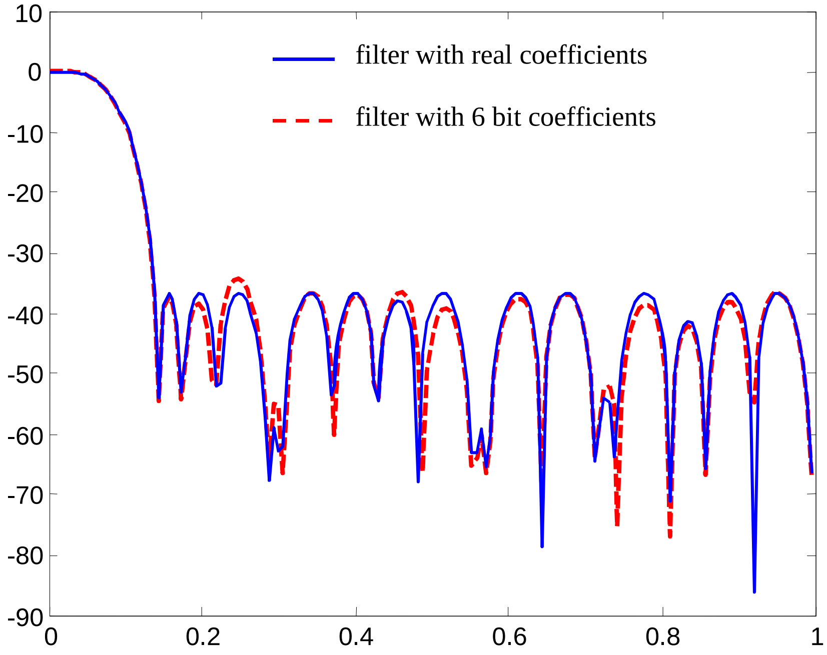


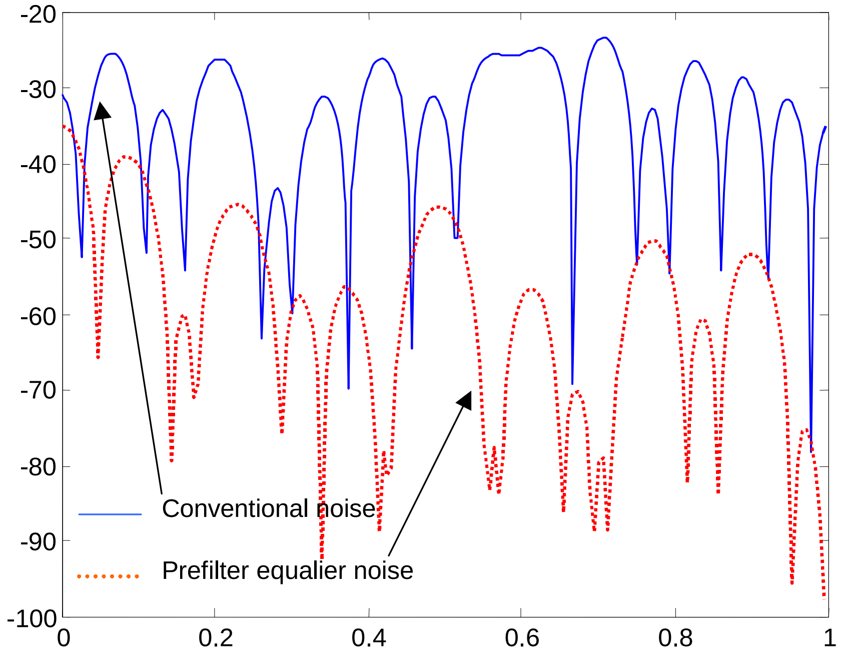
Figure 4-3 (a) Equivalent for Conventional Filter [41] (b) Equivalent for Prefilter Equalizer Structure


20
0
-20
-40
-60
-80
-100
0
0.2
0.4
0.6
0.8
1
filter with real coefficients
filter with 6 bit coefficients

(a)


10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
0
0.2
0.4
0.6
0.8
1
filter with real coefficients
filter with 6 bit coefficients

(b)



(c)

Figure 4-4 The Effect of the Coefficient Quantization on the Filter Magnitude Frequency Responses of: (a) Conventional Filter (b) Prefilter Equalizer Cascade (c) Quantization Noise for Both the Conventional Filter and the Prefilter Equalizer Cascade Filter.

Figure 4-4 (a)-(c), which are generated by using the case of example 4.1, show the effect of the coefficient quantization on the filter magnitude frequency responses of the conventional filter and the prefilter equalizer cascade filter. In this figure, the prefilter equalizer cascade filter exhibits superior performance over the conventional filter, and the sensitivity of the frequency response of the prefilter equalizer to the coefficients is reduced. This attribute allows the coefficients to be quantized to fewer bits without introducing more degradation as shown in Figure 4-4 (b). Thus, the complexity of the filter can be reduced.

4.3 Optimization of Prefilter Structure

From equation (4.1) or (4.2), we know that the only factor that must be well designed for the RRS prefilter is the length of the filter, *L*. However, according to equation (4.1), to increase the sharpness of the frequency response on the transition from the passband to the stop-band, the first null of the response of the prefilter should be placed at a point that is slightly above the stopband frequency, ω_s. This implies that the length of the prefilter, L, should be chosen to meet the criterion:

$$L < 2\pi / \omega_s , \tag{4.8}$$

For a fixed stopband frequency, ω_s, the restraint of equation (4.8) prevents the RRS prefilter from obtaining sufficient attenuation because a larger L is needed for large attenuation. Especially when the desired filter is wide-band, it is impossible to acquire an ideal prefilter that provides sufficient sharpness of transition and large attenuation at the same time. In other words, there exists a limitation to the RRS prefilter structure to be used for wide-band filter design. Therefore, the RRS structure is a very elegant prefilter that can be used to cope with a narrow band filter. However, for an arbitrary bandwidth filter, this method cannot be used in a straightforward manner. Thus, other kinds of prefilters will be investigated in the next section to meet the arbitrary bandwidth requirement.

4.3.1 Principle of Finding an Optimal Prefilter

For practical purposes, the design of the equalizer is less important, while the design of prefilter is very important. We know that the equalizer can be designed by appropriately modifying the existing algorithm to compensate for the prefilter performance with lower order compared to conventional methods. There are some principles for designing the prefilter. One of the principles is that the prefilter should be optimal in the sense that the stopband offers the largest attenuation for a given specification. The second principle is that the implementation of the prefilter should be extremely simple so that we can view the implementation as combinations of powers of two. Thus, the complexity of the implementation should be as low as possible, and the performance of the prefilter should be immune to coefficient quantization. Third, it is essential to find such an optimal prefilter so that we can extend the application of the prefilter equalizer cascade structure to the design of wide-band filters.

4.3.2 Prefilter Based on Chebyshev Function

For the sake of the area-efficient implementation, the desired filter is expected to offer quite nice results in saving the number of arithmetic operations. Here, a Chebyshev function is applied to generate an optimal prefilter. One of the considerations in choosing a Chebyshev function as the efficient prefilter is that the implementation of a Chebyshev polynomial involves only the coefficients that are extremely simple combinations of powers of two. It can be implemented by shifting and adding operations where no multiplier is required.

The first kinds of Chebyshev polynomials of order $n \geq 0$, is defined as [41]:

$$T_n(x) = \begin{cases} \cos(n\cos^{-1} x) & |x| \leq 1 \\ \cosh(n\cosh^{-1} x) & |x| > 1 \end{cases} \tag{4.9}$$

Simplifying the manipulation of equation (4.9), and introducing $x = \cos(\theta) \quad |x| < 1$ and $x = \cosh(\varphi) \quad |x| > 1$, we have

$$T_n(x) = \begin{cases} \cos(n\theta) & |x| \leq 1 \\ \cosh(n\varphi) & |x| > 1 \end{cases} \tag{4.10}$$

Then, we can easily obtain an alternative to the Chebyshev polynomial $T_n(x)$ of order n with $T_0(x) = 1$, $T_0(x) = x$ and

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)\,. \tag{4.11}$$

The above equation is often used to denote the Chebyshev polynomial, rather than using the explicit formula (4.9). Since the coefficients of the polynomial can be expressed by the combination with fewer terms of powers of two. For example, the 5th order Chebyshev polynomial can be expressed as:

$$T_5(x) = (2^2 + 2^0)x - (2^4 + 2^2)x^3 + 2^4 x^5 \tag{4.12}$$

Hence, the corresponding implementation of such a coefficient function involves only shifts and adders, which turns out to be extremely simple. The implementation is shown in Figure 4-5.

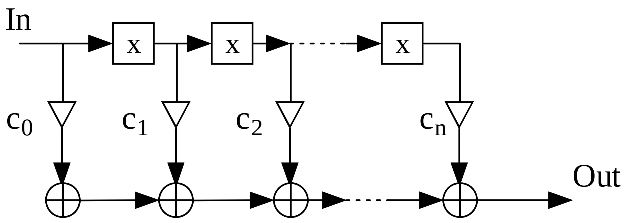


Figure 4-5 Implementation of Chebyshev Polynomial $T_n(x)$

The second most important factor for using the Chebyshev function is that such a function can offer large attenuation, which is helpful to simplify the model filter design by relieving the equalizer from increasing attenuation in the stopband.

One of the filters constructed using the Dolph Chebyshev function [41], which can be designed with any specified relative sidelobe level while simultaneously adjusting its passband width by choosing the parameter appropriately. The Dolph Chebyshev function is defined in the frequency domain as [30]:

$$h(\omega) = \frac{T_N(x_0 \cos(\omega/2))}{T_N(x_0)}, \tag{4.13}$$

where $x_0 > 1$, and ω_s is the cutoff frequency of the lowpass filter $h(\omega)$ and satisfies the condition $x_0 \cos(\omega_s/2) = 1$.

We applied equation (4.13) to form our proposed Chebyshev prefilter of N^{th} order and cutoff frequency ω_s:

$$P_{N,\omega_s}(\omega) = T_N(x)/T_N(x_s), \tag{4.14}$$

where

$$x = \cos(\omega/2)/\cos(\omega_s/2) \tag{4.15}$$

$$x_s = 1/\cos(\omega_s/2). \tag{4.16}$$

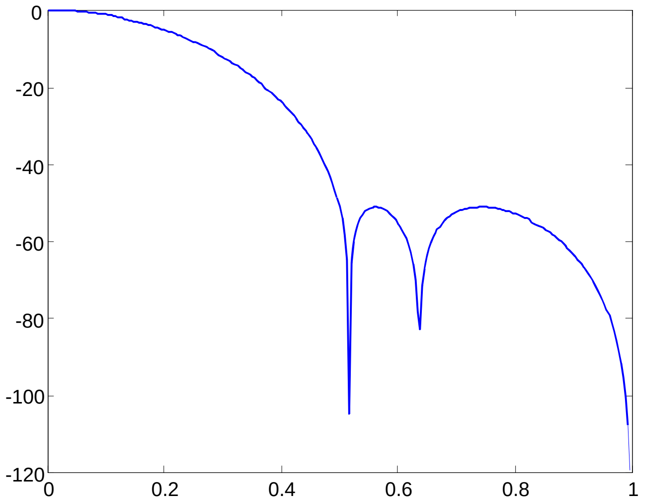


Figure 4-6 Magnitude Response of Proposed Prefilter with the Order of 5

Figure 4-6 shows the magnitude response of the prefilter $P_{5,\pi/2}(\omega)$ of the 5th order and $\pi/2$ cutoff frequency. It is clear that a Chebyshev prefilter provides a relatively larger attenuation and wider bandwidth, which is suitable to extend the prefilter equalizer cascade structure to arbitrary bandwidth filter design.

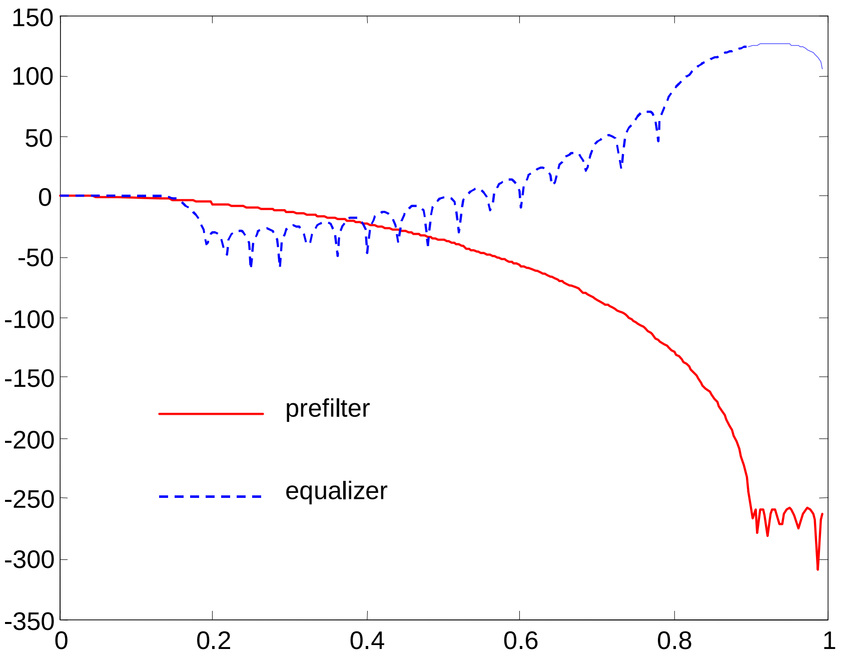
From equation (4.14), it can be noted that the minimum stopband attenuation is given by

$$\min(\delta_s) = 20\log_{10}(T_N(x_s)), \tag{4.17}$$

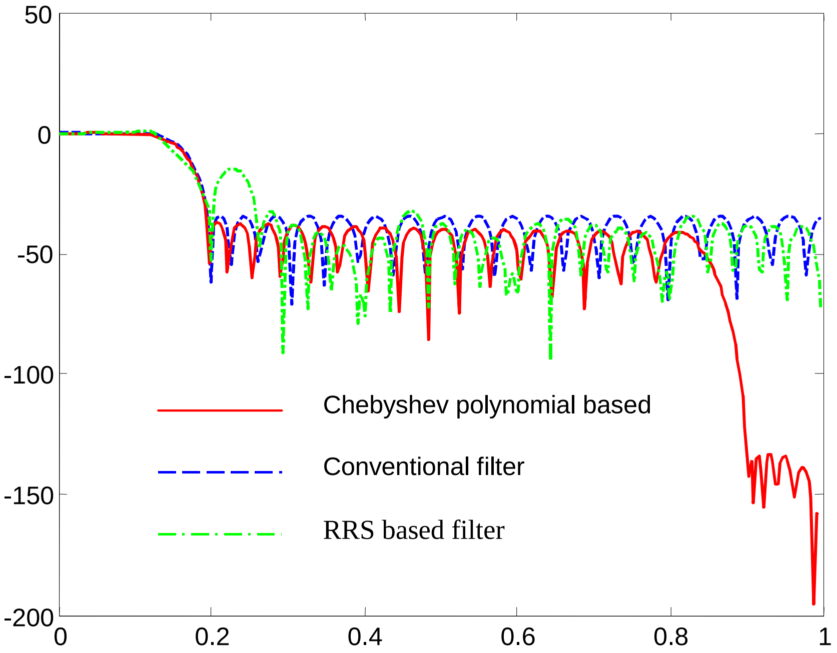
which is related to the order of *N* and the cutoff frequency of ω_s. For a given *N*, the attenuation increases as ω_s increases, and for a given ω_s, the attenuation increases as *N* increases. However, increasing *N* implies that the complexity of the prefilter will be increased, hence a better way to design the prefilter is to increase the attenuation while keeping the order *N* as low as possible. Consider a given filter specification, the choice of

ω_s should be as close as possible to the given stopband, so that the prefilter could generate a narrow transition band filter. Another consideration of choice ω_s is its influence on the implementation of $T_N(x)$. As the variable of the Chebyshev polynomal described in equation (4.11), it appears *N* times in the multiplier coefficient. To make the prefilter as simple as possible, ω_s should be chosen in a way such that x_s is a power of 2, or a combination of few sums of power of two. On the other hand, as discussed previously, the narrower ω_s is, the less the obtainable attenuation from the prefilter. That means, for a narrow band filter, a large *N* is required, which is unanticipated. Thus there exists a trade-off between *N* and ω_s. To avoid a large *N* and deal with the trade-off, an interpolated FIR filter design method [37] can be applied. This will be addressed later. In order to show the applicability of the Chebyshev polynomial to the prefilter design, an example of a lowpass FIR filter will be studied as follows.

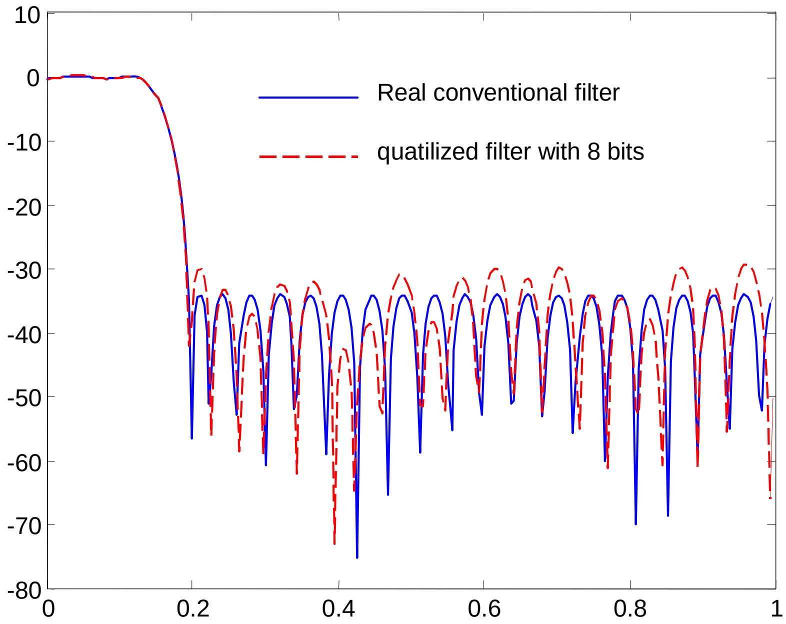
Example 4.2: Design a lowpass filter with specifications given as the passband edge $\omega_p T$ is at 0.125π, the stopband edge $\omega_s T$ at 0.195π, maximum passband ripple is 0.2dB, and the minimum stopband attenuation 35 dB. The frequency response of the proposed cascade filters with prefilter based on Chebyshev polynomial and the equalizer is shown in Figure 4-7 (a). The prefilter is a Chebyshev polynomial of order 12. Also the order of the equalizer filter is reduced to 37, and 19 multipliers are required. The resulting magnitude frequency response of the proposed filter and that of the conventional filter as well as that of the RRS based filter are shown as Figure 4-7 (b). In the same case, the conventional filter requires an order of 45 and 23 multipliers are needed.



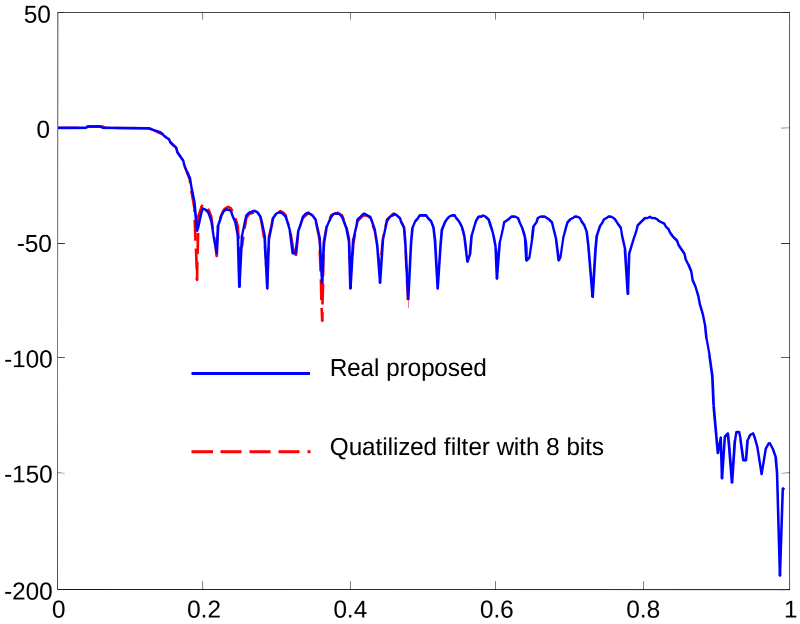
(a)The proposed prefilter and its equalizer



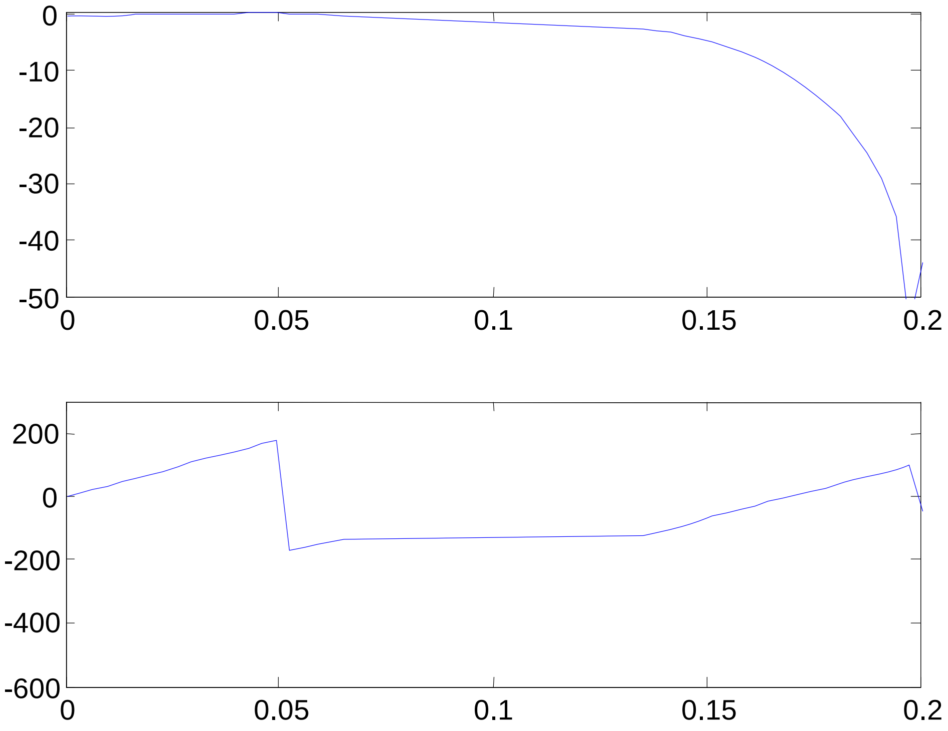
(b) The real proposed cascade filter as well as real conventional filtter and the RRS based filter



(c) The resulting conventional filter and the quantized filter with 8 bits



(d) The resulting proposed filter and the quantized filter with 8 bits



(e) The magnitude and phase responses of the proposed filter with the edge of passband at 0.125π

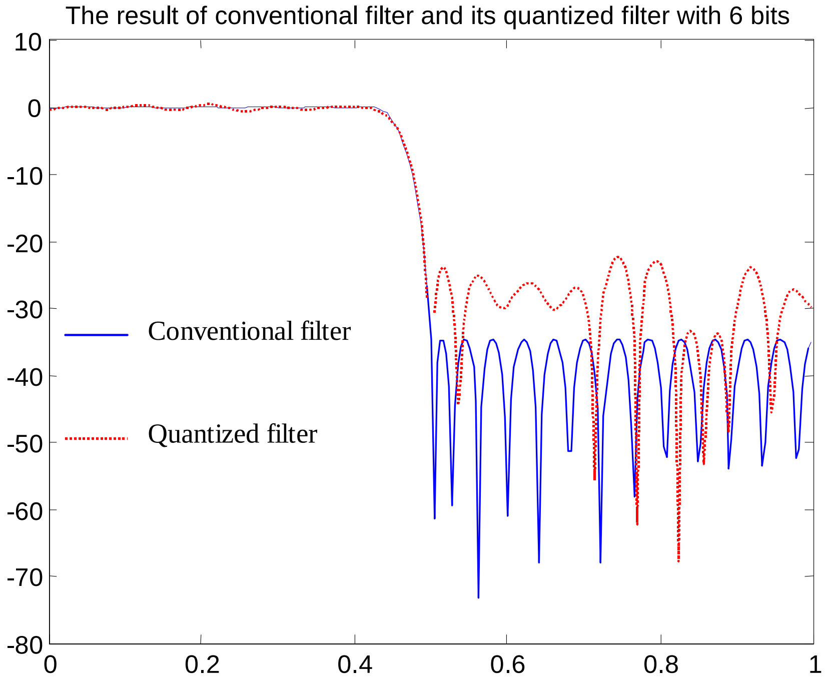
Figure 4-7 Improved FIR Filter Design with Chebyshev Polynomial Prefilter

The performance of the proposed filter is greatly improved compared to the RRS based filter, which has an equalizer of order 47, which is much larger than that of the proposed method. Figure 4-7 (c) and (d) show the effect of the coefficient quantization, on the frequency responses. It is noticeable that the Chebyshev polynomial based filter is much less sensitive to coefficient quantization than the conventional design if the same wordlength of 8 bits is taken. Therefore, when the filters are implemented, the proposed filter can save much more space due to the short wordlength being used.

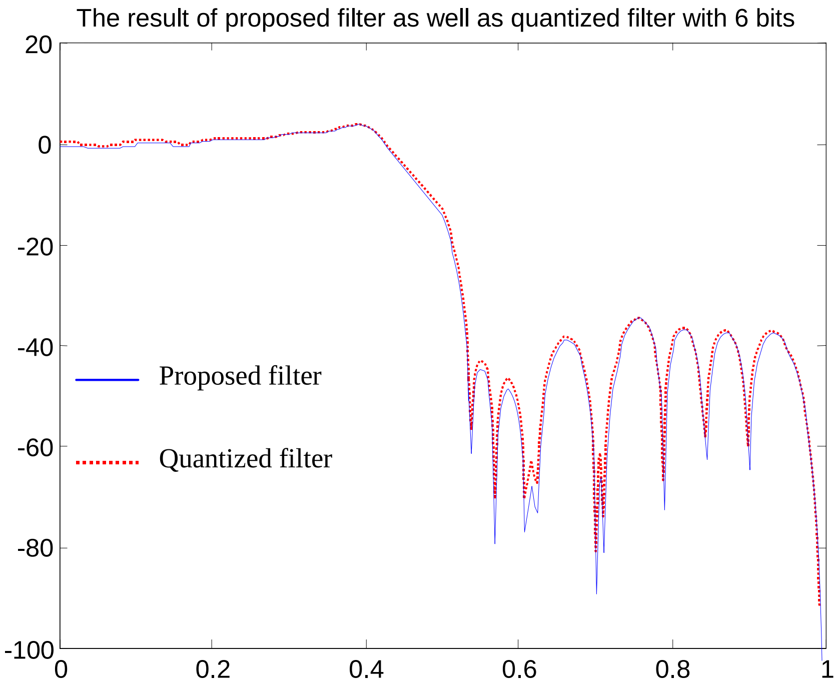
Example 4.3: Consider a lowpass filter specification with passband edge $\omega_p T$ at 0.43π, stopband edge $\omega_s T$ at 0.5π, maximum passband ripple of0.2 dB, and a minimum stopband attenuation of 35 dB.

The frequency response of a conventionally designed filter (using *remez* in Matlab) is shown as Figure 4-8(a), which leads to a filter of order 45 with 23 multipliers required. A prefilter equalizer cascade based on a Chebyshev polynomial is shown in Figure 4-8(b). The prefilter is a Chebyshev polynomial of order 5. Also the order of the equalizer filter is reduced to 33, and 17 multipliers are required. Figure 4-8 (a) and (b) also show the effect of the coefficient quantization. It is noticeable that the Chebyshev polynomial based filter is much less sensitive to coefficient quantization than the conventional design if the same wordlength of 6 bits is taken. Figure 4-8(c) shows the result of the conventional filter with coefficients being quantized to 9 bits. The performance is almost the same as the proposed filter with 6 bits. It indicates that lower precision is needed by the proposed filter design to meet the same filter specifications.

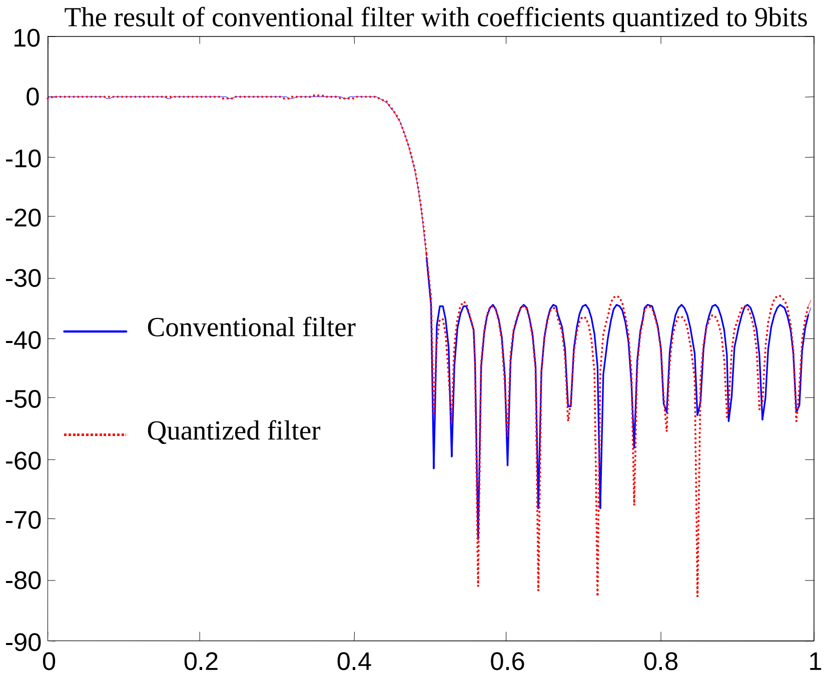
Though the proposed filter based on the Chebyshev function is relatively space efficient, Figure 4-8(d) shows there is still some distortion in the performance in terms of frequency response. What is the reason for this degradation? From Figure 4-8(e), it is not difficult for us to see that the prefilter does provide a large attenuation but the transition band of the prefilter is relatively wide, which causes a decrease in the performance in the passband. The equalizer compensates for some of the distortion. However, the order of the equalizer should be as small as possible to obtain a space efficient filter design. Therefore, this distortion cannot be completely compensated while keeping area requirements low. In response to this situation, a halfband filter that is used as the prefilter will be described in the next section.



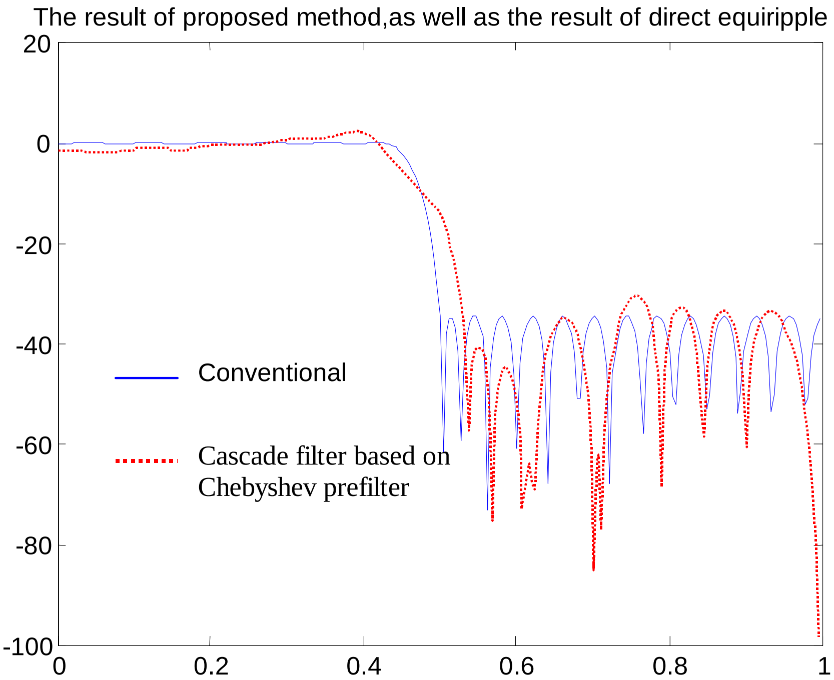
(a)


The result of proposed filter as well as quantized filter with 6 bits
20
0
-20
-40
-60
-80
-100
0
0.2
0.4
0.6
0.8
1
Proposed filter
Quantized filter

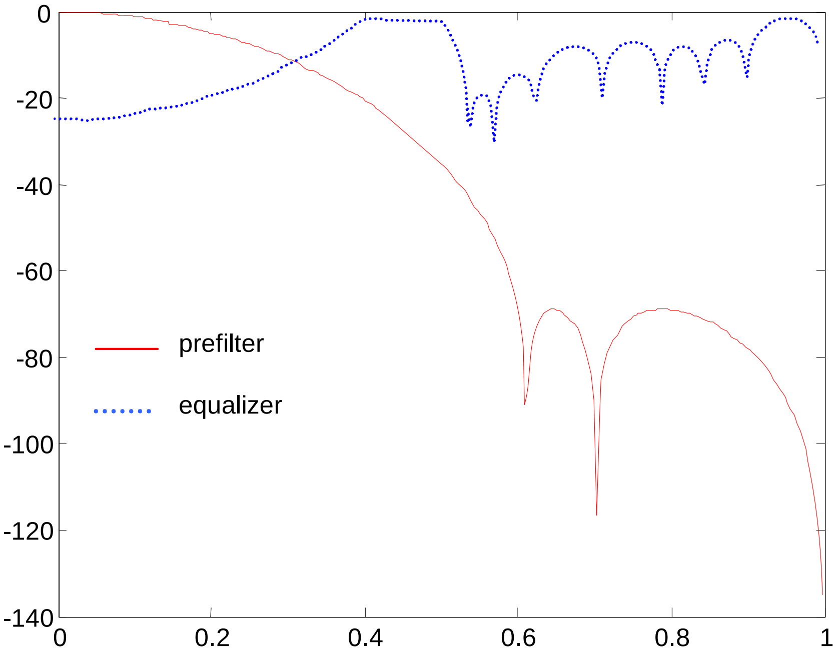
(b)


The result of conventional filter with coefficients quantized to 9bits
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
0
0.2
0.4
0.6
0.8
1
Conventional filter
Quantized filter

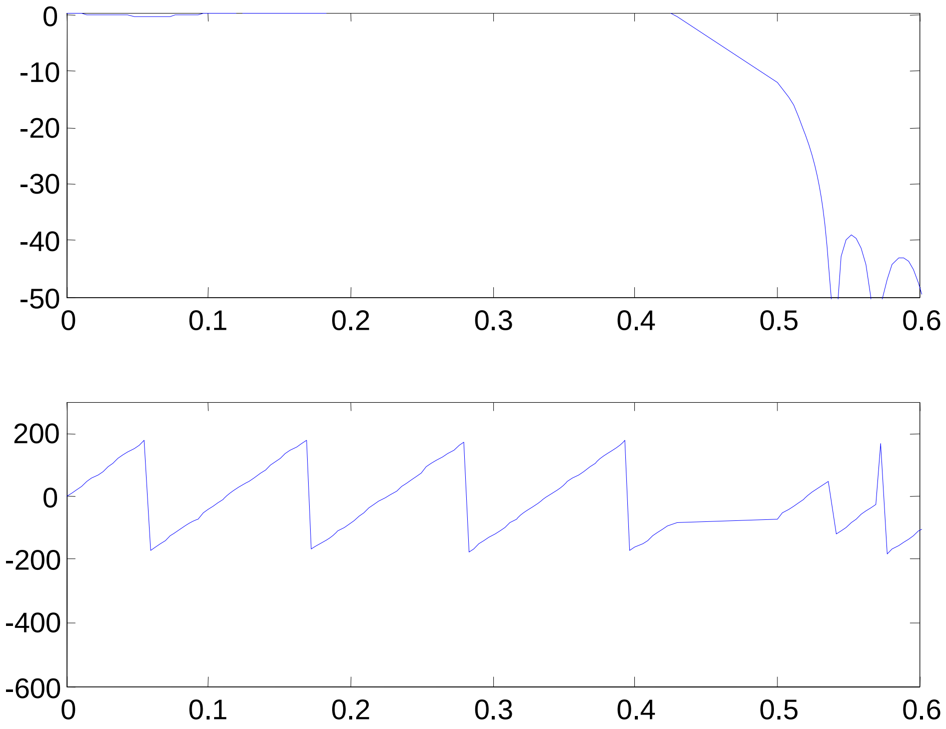
(c)


The result of proposed method,as well as the result of direct equiripple
20
0
-20
-40
-60
-80
-100
0
0.2
0.4
0.6
0.8
1
Conventional
Cascade filter based on Chebyshev prefilter

(d)


0
-20
-40
-60
-80
-100
-120
-140
0
0.2
0.4
0.6
0.8
1
prefilter
equalizer

(e)



(f) The magnitude and phase responses of the proposed filter with passband edge at 0.43π

Figure 4-8 Filter Response with Stopband Edge at 0.5π by Using Proposed Chebyshev Polynomial Prefilter

4.3.3 Halfband Prefilter

It is known that a half-band filter is relatively simple to implement because half of its coefficients are identically zero. By nature, such a filter is computationally more efficient than other filters of the same order [48][49]. In addition, when a half-band filter is designed with a cutoff frequency near $\pi / 2$, it has a sharp transition band, which is an attractive feature in a design of an appropriate prefilter.

A typical half-band filter is an *L*th band filter with $L = 2$, its transfer function is given by

$$H(z) = a + z^{-1}E_1(z^2)\,, \tag{4.18}$$

where $E_1(z^2)$ is the other polyphase component of $H(z)$.

As a result, the input samples appear at the output without any distortion for values of *n*, whereas the in-between samples are determined by interpolation, i.e.

$$y[2n] = \alpha x[n]\,, \tag{4.19}$$

If we assume that $\alpha = 1/2$, then the half band filter satisfies the following condition:

$$h(2n) = \begin{cases} \dfrac{1}{2} & n = 0, \\ 0 & n \neq 0, \end{cases} \, . \tag{4.20}$$

Equation (4.20) indicates that about 50% of the coefficients of *h*(*n*) are zeros. This reduces the number of multiplications required in the filter implementation, which makes the filter computationally quite attractive.

Additionally, if take $\alpha = 1/2$ we have

$$H(z) + H(-z) = 1, \tag{4.21}$$

If $H(z)$ has real coefficients, then $H(-e^{j\omega}) = H(e^{j(\pi-\omega)})$, which leads to the following equation:

$$H(e^{j\omega}) + H(e^{j(\pi-\omega)}) = 1. \tag{4.22}$$

The above equation exhibits symmetry with respect to the frequency $\pi/2$, and the band edges are symmetric with respect to $\pi/2$, i.e. $\omega_p + \omega_s = \pi$. This attractive property provides us a design with sharp transition band filter near $\pi/2$.

By combining this idea with our proposed variable precision method, a half-band filter further reduces the space required for implementation. Since the coefficients of the half-band filter can be reduced to a few powers of two combinations, the half-band filter can thus be implemented efficiently with only a few adders and shifts.

Example 4.4: Consider a lowpass filter with the same specifications as the Chebyshev method: the pass-band edge located is at 0.43π and the stop-band edge located at 0.5π. The maximum pass-band ripple is $DB_p = 0.2db$ while the allowable stop-band ripple is set as $DB_s = -35db$.

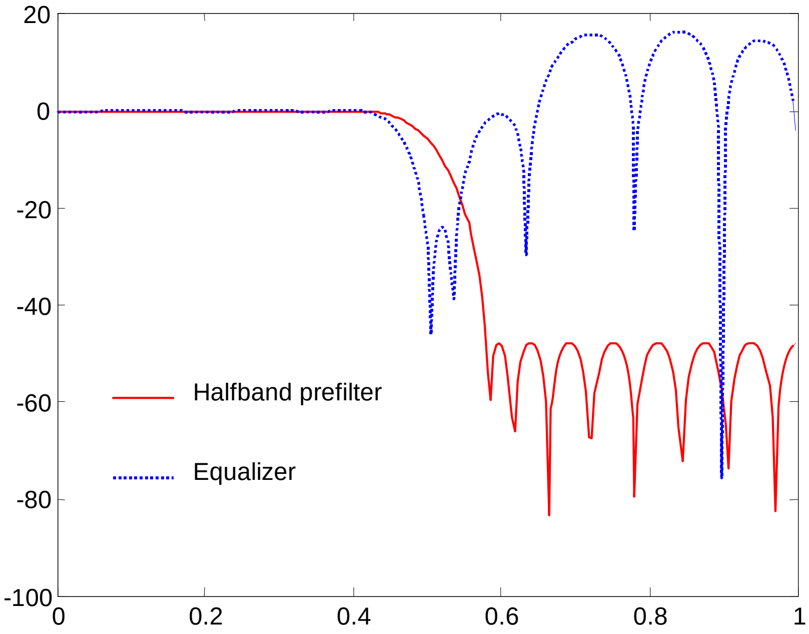
For this wideband filter design, the required order to meet the specifications by using the conventional direct (Parks-McClellan) method is 45. However, the total order of the filter designed using our proposed filter based on half-band prefilter is 42, which includes the order of the prefilter, 19, and the order of the equalizer, 23. The details of the filter design implementation are shown in Table 4-2.

In this design, each coefficient can also be quantized into the CSD representation with one or two nonzero digits. Therefore, the required hardware can be reduced furthur.

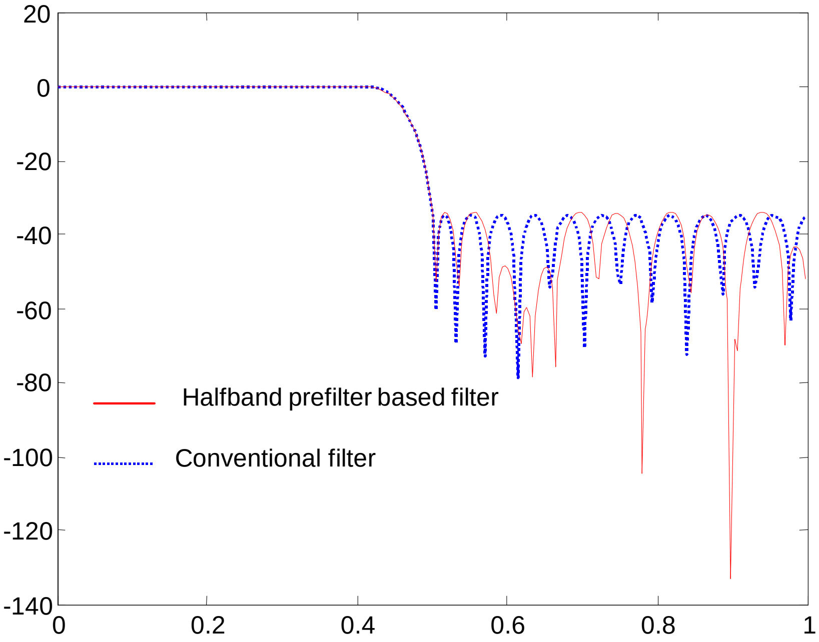
Table 4-2 Complexity Comparisons

	Delays	Adds	Multipliers
Parks-McClellan	45	45	23
Chebyshev-based prefilter	33	36	17
Halfband-based prefilter	42	31	17

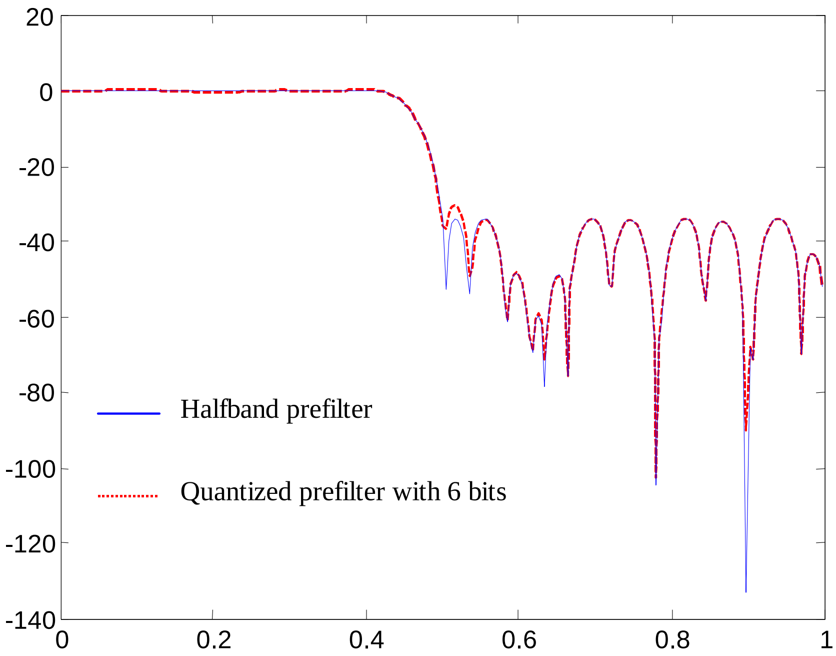
The frequency responses of the filter produced by the cascade filter based on a halfband prefilter, as well as the frequency response of the filter with conventional direct equiripple (Parks-McClellan) design are shown in Figure 4-9. From Figure 4-9 (a), we can see that the halfband prefilter provides a nice frequency response with a sharper transition band compared to the Chebyshev prefilter design, as shown in Figure 4-8(e). This quality relieves the burden on the equalizer design. Hence, the required order of the equalizer is 23, much lower than that of the Chebyshev method, which is 33. The performance of the resulting filter is improved as shown in Figure 4-9 (b). The effect of the coefficients quantization is illustrated inFigure 4-9 (c) and (d). This design also maintains the advantage of low sensitivity, which makes it possible to create a low sensitivity structure with a reduced number of taps. Also, it reduces the wordlength requirement of the coefficients, so that the specification is satisfied while simultaneously reducing the resulting hardware area required by the implementation.



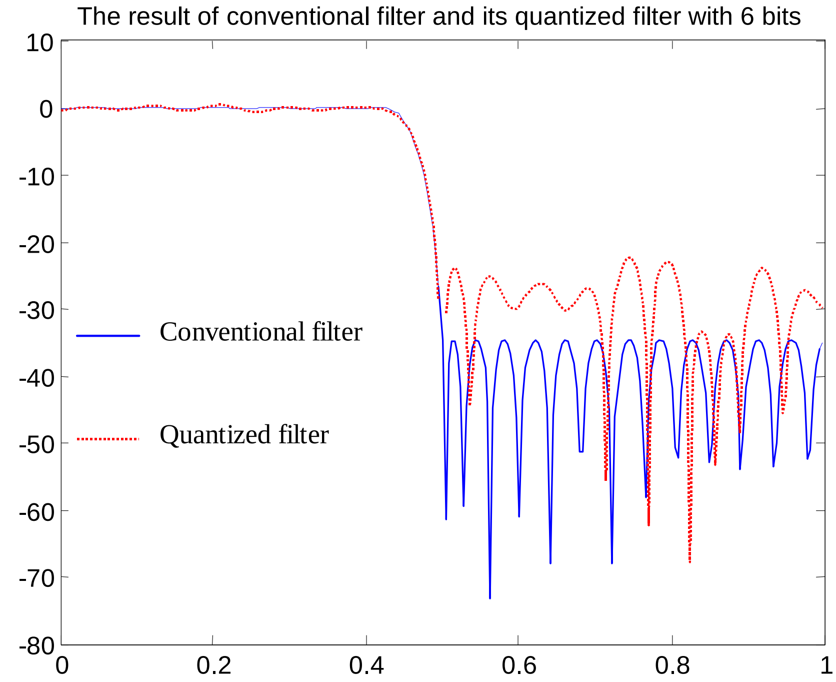
(a) Frequency Response of the Halfband Prefilter and that of the Equalizer



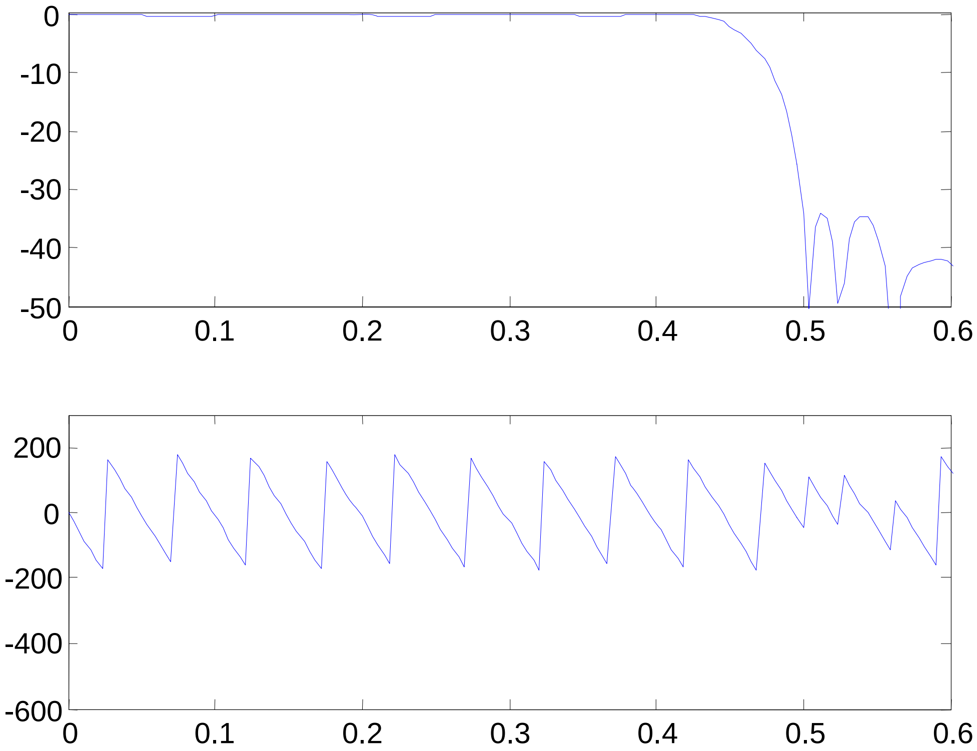
(b) Frequency Response of the Cascade Filter Based on Halfband Prefilter as well as that of the Conventional Filter



(c) The Result of Proposed Cascade Filter and its Quantized Filter with 6 Bits



(d) The result of Conventional Filter and its Quantized Filter with 6 Bits



(e) The magnitude and phase responses of the halfband based filter with passband at 0.43π

Figure 4-9 Filter Design Based on Halfband Prefilter as well as the Conventional Filter

4.4 Concluding Remarks

For the sake of space efficiency, the prefilter and equalizer cascade structure filter is introduced in this chapter. Our interest focuses on such a structure in which the prefilters provide a sharp transition-band and increase the stopband attenuation. Both of these efforts relieve the design burden on the equalizer, and hence lead to a good design with low coefficient sensitivity and small order, while the filter specification is still satisfied. For a narrow band filter design, an RRS prefilter presents superior performance in terms of the hardware implementation. In order to further extend the application to an arbitrary wide band filter, two other kinds of prefilter, specifically the Chebyshev polynomial prefilter and halfband prefilter, are investigated. The attractive feature of the Chebyshev polynomial prefilter is that it can supply great attenuation with very small order. However, for a single Chebyshev prefilter, if great attenuation is required, the transition band is wide, and some perturbation exists in the passband. Therefore, to some extent, the distortion of the performance of the designed filter is not easy to compensate. To solve the weakness and get a relatively sharp transition band prefilter, a halfband prefilter is developed. The examples given in this chapter have demonstrated the effectiveness of our design methods and improve the performance of a desired filter step by step. Moreover, each coefficient can also be quantized using reduced wordlength, or the CSD representation with only one or two nonzero digits can be used, since the low coefficient sensitivity is obtainable. In this way, we can greatly reduce the resulting hardware.

Chapter 5 Improved Masking Filter Development

In Chapter 4, we have described several different methods to develop an efficient cascade structure for simplifying hardware implementation. These methods to some extent improved the prefilter and could extend the applications of the prefilter equalizer structure to wider bandwidth filter design. Another motivation of our design is to get a high-performance effect such as combining a narrow transition band with a very large stop-band attenuation. In practice, if the transition band of the filter is sharp and the attenuation is large, the order of the filter usually is very high. Such a filter with high order is usually difficult to implement with a limited number of arithmetic operations. The purpose of this chapter is to develop techniques suitable for designing these kinds of filters by improving the developed prefilter and equalizer to reduce the overall complexity calculations as much as possible.

In this chapter, our proposed structure is combined with the interpolated FIR filter (IFIR) [37] method and the frequency masking filter technique [38]. How to improve the filter performance by using the interpolated FIR filter is introduced in Section 5.1. In Section 5.2, the masking filter is integrated into the FIR filter design, which can provide a sharp transition band and lead to a superior realization compared to the proposed filter described in Chapter 4. Section 5.3 describes the details of the systematic procedure, and examples are presented to demonstrate the effectiveness of our design. Concluding remarks are also contained in Section 5.4.

5.1 Improved Prefilter Using Interpolated FIR filter Method

To obtain a narrow transition band filter and simplify the filter structure, an interpolated FIR filter can be used [50]. We will take advantage of the characteristic of the IFIR filter in our proposed prefilter design to enhance the filter performance. The basic idea of an IFIR prefilter is to implement the improved prefilter as a cascade of two FIR sections, where one section generates the sparse set of impulse response values with every *M*th sample being nonzero, while the other section, the interpolator, performs the interpolation. The interpolator is often implemented with only a few simple arithmetic operations.

5.1.1 IFIR Filter Structure

Let us consider a filter $H(z)$ with impulse response $h(n)$. This is a model filter that determines the frequency behavior of the final interpolated FIR filter. If *M*-1 zero valued samples are inserted between the samples of $h(n)$, a new sequence, $h_M(n)$, can be obtained as

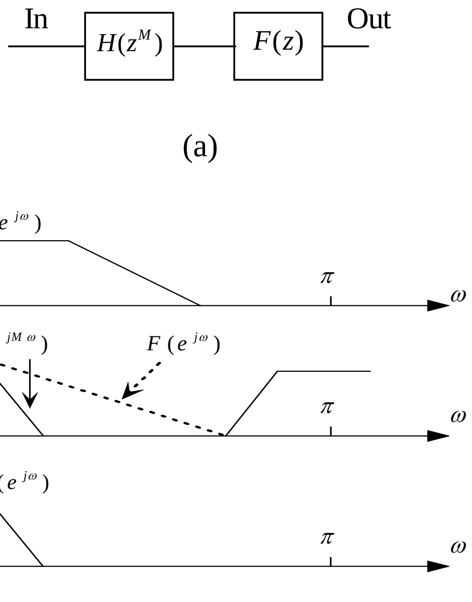
$$h_M(n) = \begin{cases} h(n/M) & n = iM,\ i = 0, \pm 1, \pm 2, \ldots \\ 0 & otherwise \end{cases} \tag{5.1}$$

The corresponding transfer function is defined as $H(z^M)$. Then, the interpolated FIR (IFIR) filter transfer function can be expressed as follows:

$$H_M(z) = H(z^M)F(z), \tag{5.2}$$

where, $H_M(z)$ is the final interpolated FIR filter, $H(z^M)$ is derived by replacing each delay element of the model filter, $H(z)$, with M delay elements, and $F(z)$ is an interpolator.

The IFIR filter is shown as Figure 5-1(a) and the frequency magnitude of the IFIR is shown in Figure 5-1(b). It is noted that the passband and stopband characteristics of the IFIR filter are the same as those in the model filter, but the passband and stopband width are only $1/M$ times those of the model filter. This means that the effect of the interpolation is to shrink the passband and transition bands without any significant increase in the number of the arithmetic operations. In addition, according to the estimation of the filter complexity in equation (1.2), the IFIR filter requires approximately only about $1/M$ times the number of arithmetic operations to meet the same specification of an equivalent conventional filter design since the transition band of the model filter is M times that of the conventional filter.



(a)

(b)

Figure 5-1 Interpolated FIR filter with $M = 2$

5.1.2 Proposed IFIR Filter Structure

The difference between our design method and the general IFIR technique lies in that both the model filter $H(z)$ and the interpolator filter $F(z)$ are restricted to be identical filters except for the periodicity by using the folding technique [1]. Thus it is possible to map all the sub-filters used in the structure to a single hardware unit, which results in an efficient hardware design with fewer number of adders at the cost of more delays. In order to improve the area efficiency, a simple filter with low order and wide transition band can be chosen as the model filter. Several subfilters with different periodicities from the model filter can be cascaded together to remove the undesired band edges and make the transition band of the desired filter sharp to meet the design requirements.

Our other interest is to design wide passband FIR filters. In this section, the complementary filter, $H_c(z) = Z^{-N/2} - H(z)$ [41], is used to synthesize a wide-band low-pass/high-pass by subtracting the output of the narrow-band filter from the delayed version of the input. The proposed filter structure is given as shown in Figure 5-2. Here, *N* is the order of the model filter. M_i is the factor of the up-sample rate of the model filter.

5.1.3 Design Based on Proposed IFIR Approach

Assume the model filter, $H(z)$, is lowpass filter. Then, $H_c(z)$, the complementary filter of the model filter is a highpass filter. The filters $H\left(z^M\right)$ and $H_c\left(z^M\right)$ are derived by replacing each delay element of the model filters, $H\left(z\right)$ and $H_c\left(z\right)$, respectively, with *M* delay elements. When the interpolator shown in Figure 5-1(a) is restricted to

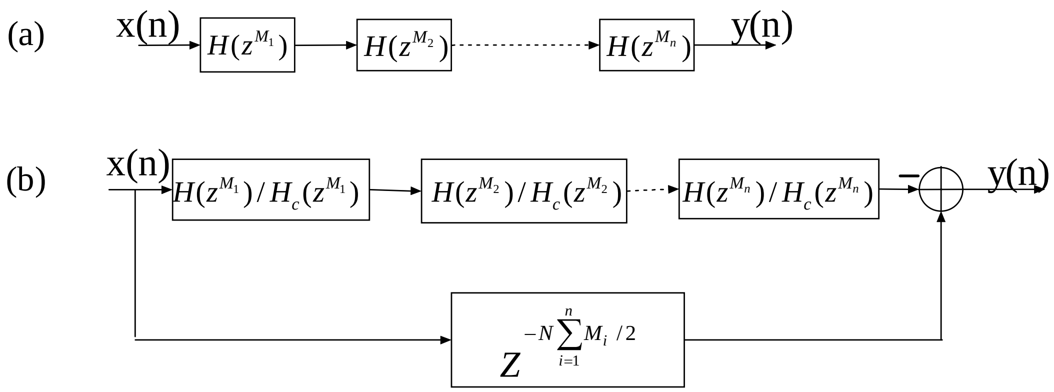


Figure 5-2 Illustration of the Proposed IFIR Filters (a) Proposed Narrow-band IFIR Filter (b) Proposed Wide-band IFIR Filter

be identical to the same model filter except for the periodicity, then the different filters are obtained as follows. A lowpass narrow band filter can be created by

$$H_{NB,LP}(z) = \prod_{n=1}^{M} H(z^n)\,. \tag{5.3}$$

A highpass narrow band filter can be created by

$$H_{NB,HP}(z) = \prod_{n=even}^{M_1} H(z^n) * \prod_{m=odd}^{M_2} H_c(z^m)\,. \tag{5.4}$$

The wide band filter is obtained by taking the complement of a narrow band filter. Then, we have a lowpass wide band filter, formed by

$$H_{WB,LP}(z) = z^{-N(\sum_{n=2}^{M_1} n + \sum_{m=1}^{M_2} m)/2} - H_{NB,HP}(z)\,. \tag{5.5}$$

A highpass wide band filter is formed by

$$H_{WB,HP}(z) = z^{-N(\sum_{n=1}^{M} n)/2} - H_{NB,LP}(z)\,. \tag{5.6}$$

For the FIR filter design, we assume that the frequency response is normalized to π, and the lowpass model filter with passband edge at $\omega_{\mathrm{mod},c}$ and the stopband edge at $\omega_{\mathrm{mod},s}$. Then, the complementary model filter will be a highpass filter with passband edge at $\omega_{\mathrm{mod},s}$ and the stopband edge at $\omega_{\mathrm{mod},c}$. From equations (5.3-5.6) one can derive the relationship of the edges of the desired filters and that of the model filters.

For a lowpass narrow band filter, we find

$$\begin{cases} \omega_{desired,NB,LP,c} = \omega_{\mathrm{mod},c} / M \\ \omega_{desired,NB,LP,s} = \omega_{\mathrm{mod},s} / M \end{cases}, \tag{5.7}$$

For a highpass narrow band filter, we have

$$\begin{cases} \omega_{desired\ NB,HP,c} = \max\{\pi - \omega_{\mathrm{mod},c} / M_1, \pi - (\pi - \omega_{\mathrm{mod},s}) / M_2\} \\ \omega_{desired\ NB,HP,s} = \max\{\pi - \omega_{\mathrm{mod},s} / M_1, \pi - (\pi - \omega_{\mathrm{mod},c}) / M_2\} \end{cases}, \tag{5.8}$$

A wide band filter can be obtained from a narrow band filter created first, for a lowpass wide band filter we have

$$\begin{cases} \omega_{desired,WB,LP,c} = \omega_{desired,NB,HP,s} \\ \omega_{desired,WB,LP,s} = \omega_{desired,NB,HP,c} \end{cases}, \tag{5.9}$$

A highpass wide band filter can be given by

$$\begin{cases} \omega_{desired,WB,HP,c} = \omega_{desired,NB,LP,s} \\ \omega_{desired,WB,HP,s} = \omega_{desired,NB,LP,c} \end{cases}, \tag{5.10}$$

5.1.4 Example Designs

In this section, we compare the filters obtained by our proposed design approach with the conventional method.

Example 5.1: Let us consider a wide band lowpass filter design with the specifications as follows: the passband edge of the wide band filter is at 0.68π, the stopband edge of the filter is at 0.80π, and the ripple of the passband and the stopband of the desired filter are 0.02 and 0.015, respectively.

The conventional implementation of this filter by using the Parks-McClellan FIR filter design requires a filter order of 30, corresponding to 15 multipliers, 30 adders and 30 delay operations. For comparison, when the proposed IFIR filter described as equation (5.5) is used, the implemented structure is shown in Figure 5-3, where M_1 is selected as 2, and M_2 is equal to 1. The model filter is designed to the same specification as the designed filter and the required order is 14. Thus, for the proposed IFIR filter implementation, only 7 multipliers, 16 adders and 28 delay operations are needed. Both the magnitude frequency response of the proposed IFIR prefilter and that of the conventional filter are shown in Figure 5-4. For this example, we can see that the performance of the proposed prefilter has met the overall requirements alone, therefore it is not necessary to cascade an additional equalizer. Occasionally, this happy situation can help us to save even more area. Finally, we can implement the desired filter by reducing the area about 40%.

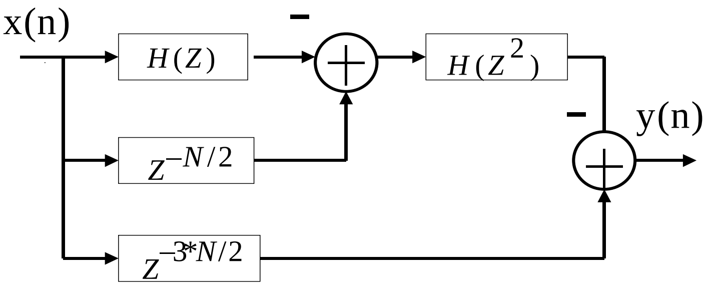
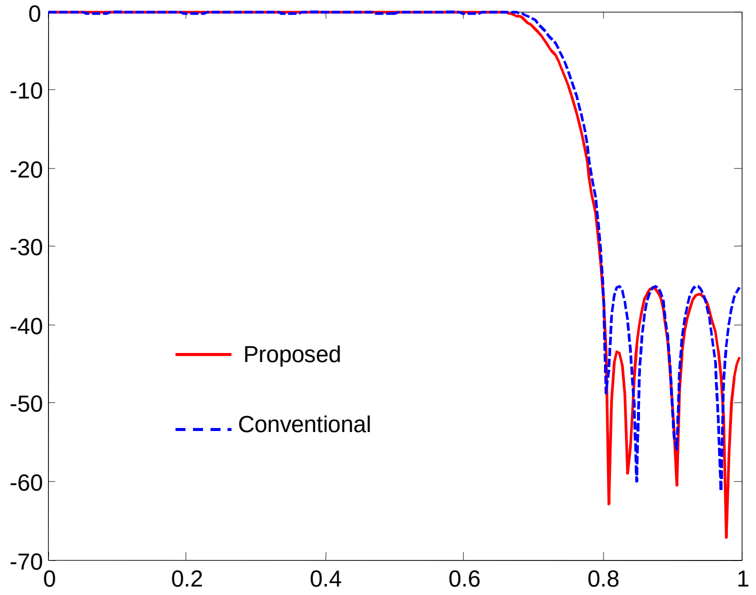
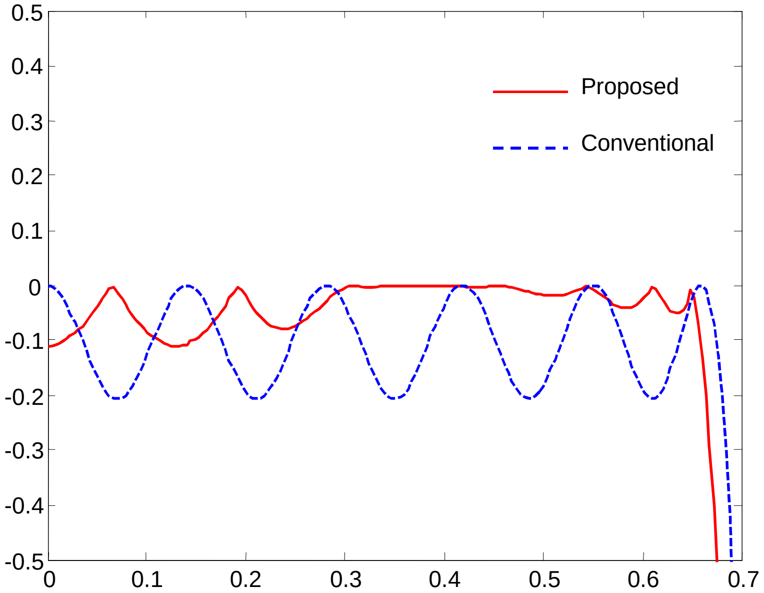


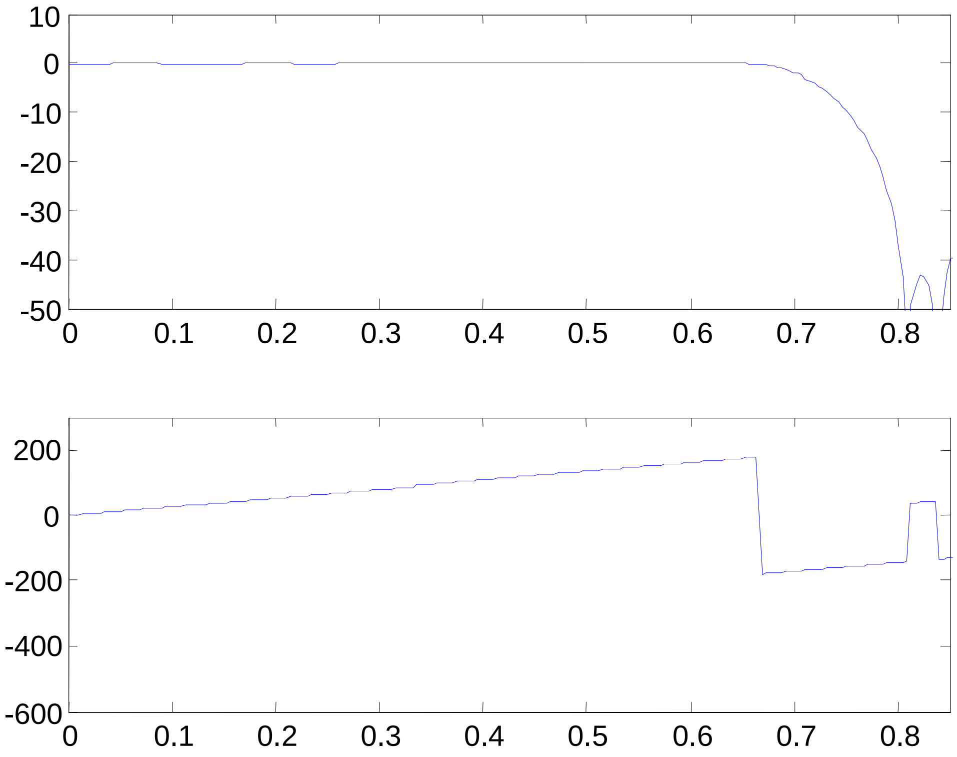
Figure 5-3 Structure of the proposed filter



(a)The Magnitude Frequency Response of the Wide-band FIR Filter



(b)The Ripple on the Passband



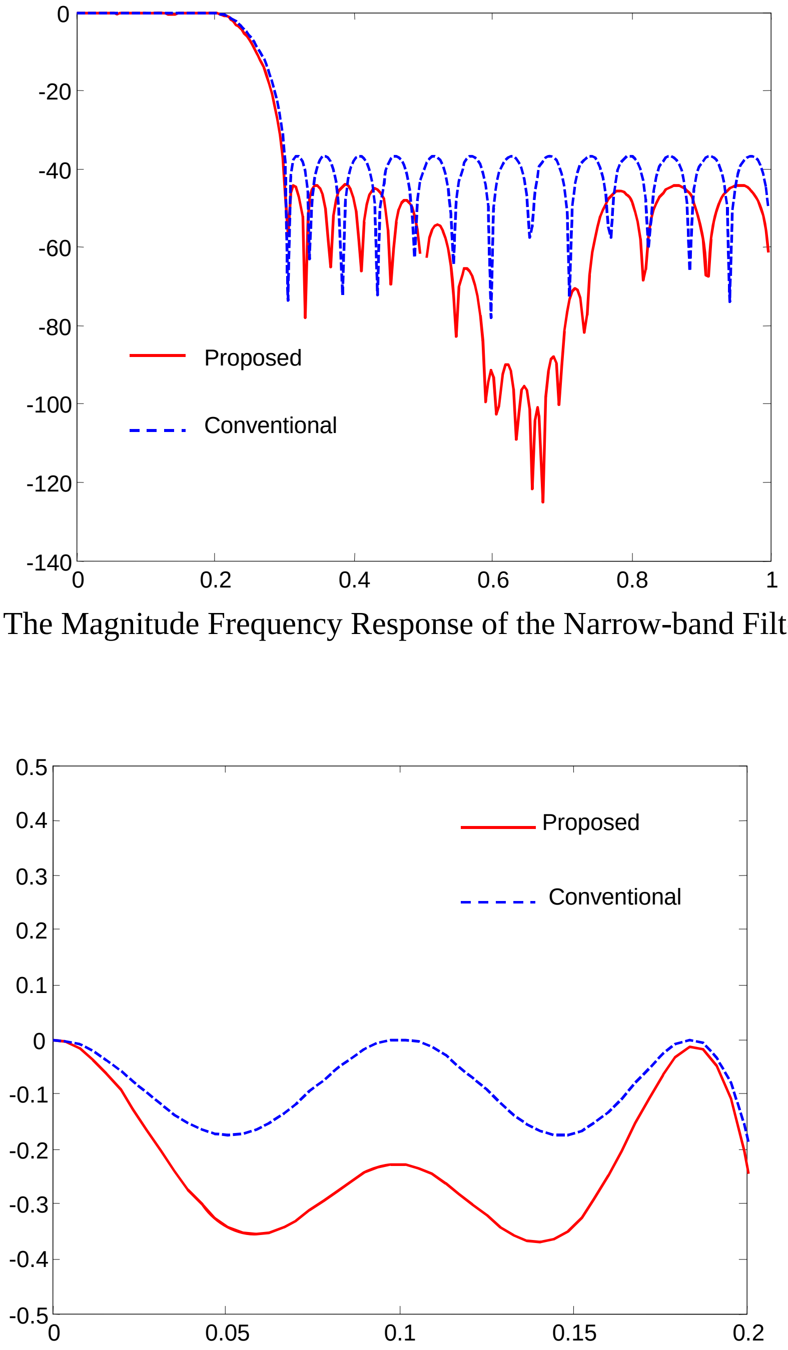
(c) The magnitude and phase responses on the proposed filter passband with edge at $0.68\,\pi$

Figure 5-4 The Frequency Response of the Proposed Wide-band Prefilter as well as the Conventional Filter

Example 5.2: In this example, a narrow-band lowpass filter is designed to satisfy the following specifications: the passband edge of the filter is at 0.2π, the stopband edge of the filter is at 0.3π, the maximum deviation of the passband is 0.2dB and the minimum stopband attenuation is 40dB.

The conventional implementation of such filter requires a filter order of 41, corresponding to 21 multipliers, 41 adders and 41 delay operations. If the proposed IFIR approach is used and $M = 2$ as shown in equation (5.3), the order of the prefilter is expected to be 20. The frequency response of the prefilter is shown in Figure 5-5. The corresponding deviation in the passband of the prefilter is greater than that of the filter specification (0.2dB), which can be seen in Figure 5-5 (b). Therefore, an equalizer of order 7 is employed to compensate the deviation. After cascading the prefilter and the equalizer, the proposed filter performance (as shown in Figure 5-6) is improved, and the total filter order to meet the given filter specification is 27. Only 14 multipliers, 27 adders and 47 delay operations are needed. In contrast to the conventional design method, this design achieves about 34% savings in terms of the number of operations.

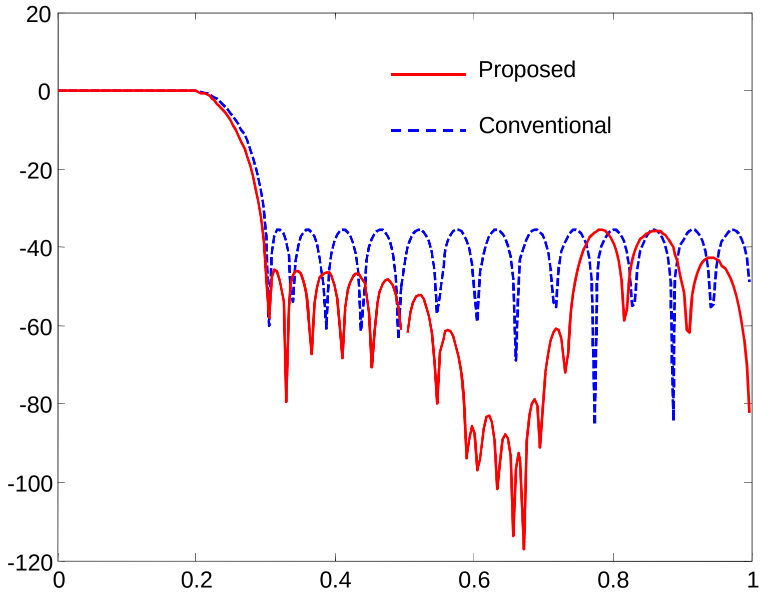
For these examples, we are able to design arbitrary bandwidth FIR filters with significant hardware savings by using our proposed IFIR approach. However, this approach is more suitable for very narrow/wide band filter design. To generalize this method to a wider range of applications, next we will develop a more efficient structure by using the masking technique.



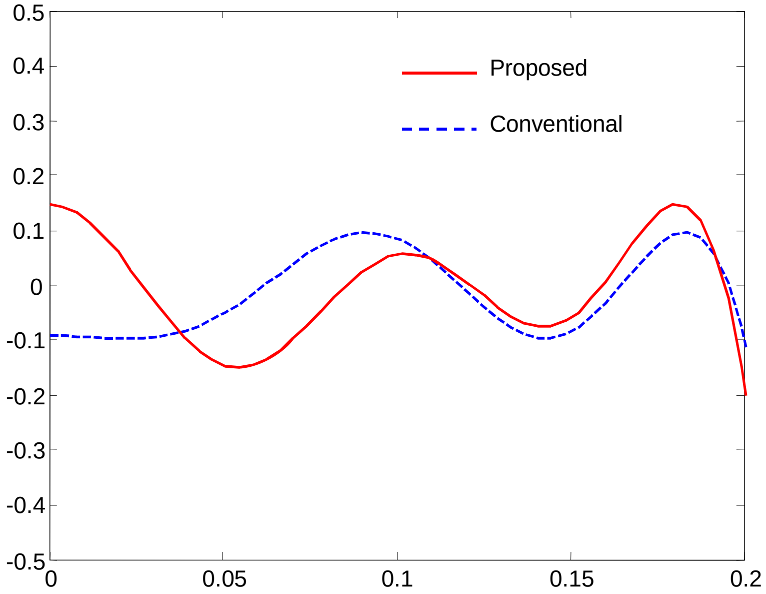
(a) The Magnitude Frequency Response of the Narrow-band Filter

(b) The Ripple on the Passband

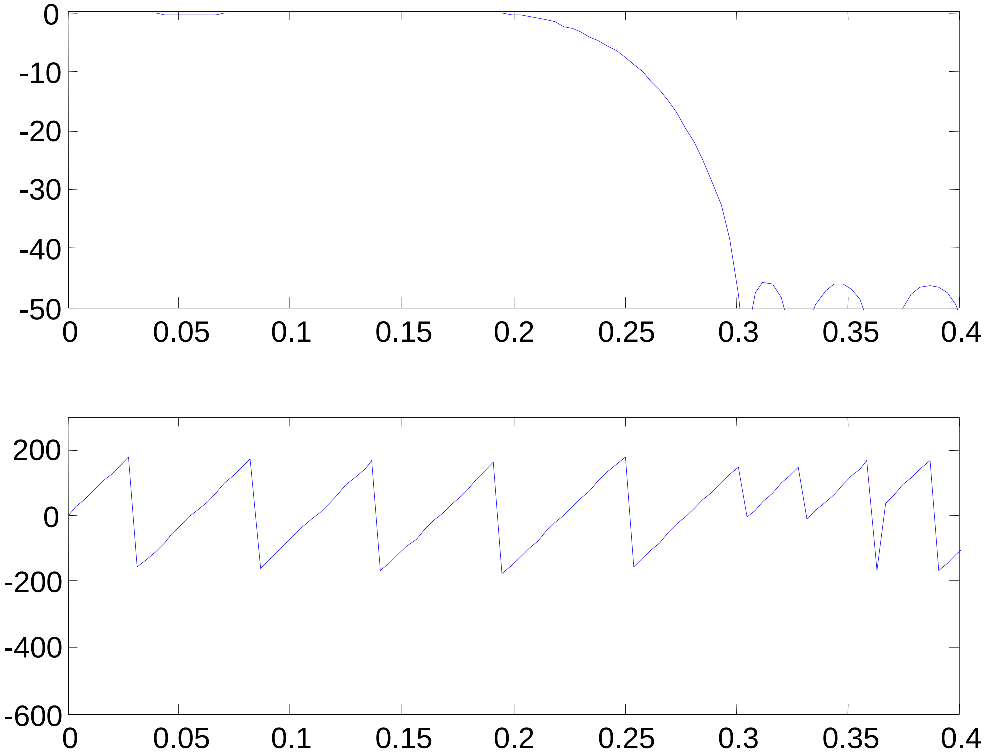
Figure 5-5 The Frequency Response of the Proposed Narrow-band Prefilter as well as the Conventional Filter



(a) The Magnitude Frequency Response of the Desired Filters



(b)The Ripple on the Passband



(c) The magnitude and phase responses on passband of the proposed filter with the edge 0.2π

Figure 5-6 The Frequency Response of the Proposed Prefilter and Equalizer Cascaded Filter as well as the Conventional Filter

5.2 Improved Prefilter with Masking Method

An alternative method to reduce the arithmetic complexity is to use frequency-response masking [38]. Our purpose here is to explore the practicality of the wideband prefilter created by using the frequency-response masking method. We can construct an arbitrary wideband filter with a model filter, its complementary filter and two masking filters. The structure of such a prefilter is shown in Figure 5-7, whose transfer function is given by:

$$P(z) = H\left(z^M\right)F\left(z\right) + H_c(z^M)F_c(z)\,, \tag{5.11}$$

where $F\left(z\right)$ and $F_c\left(z\right)$ are the masking filters and $H\left(z^M\right)$ is obtained from the model filter $H\left(z\right)$ by replacing z^{-1} with z^{-M}. $H_c\left(z^M\right)$ denotes the complementary filter of the filter $H\left(z^M\right)$ and is given by:

$$H_c(z^M) = z^{-MN/2} - H(z^M)\,, \tag{5.12}$$

Then, we have the desired filter implementation as shown in Figure 5-8.

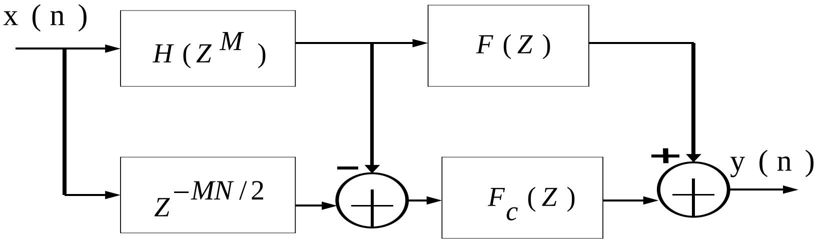


Figure 5-7 Masking Prefilter Structure

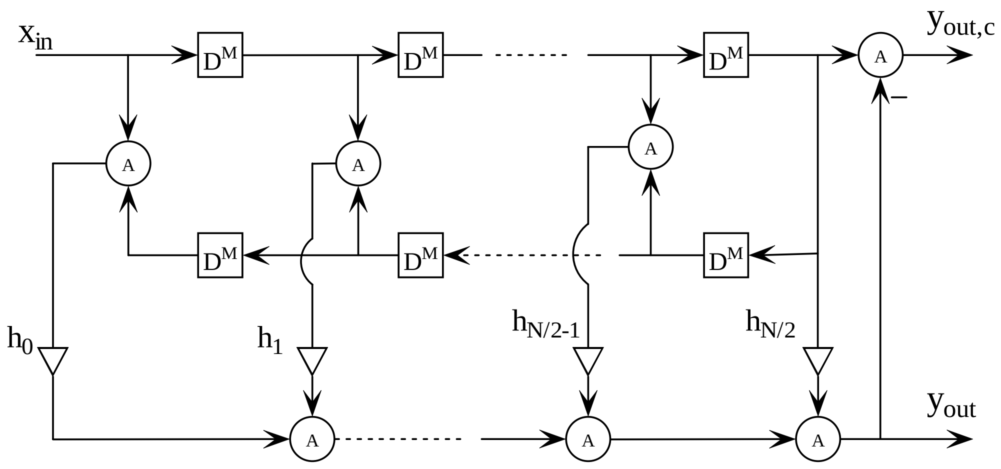


Figure 5-8 The M^{th} Interpolated Filter and Its Complementary Filter

5.2.1 Frequency Response of Masking Filter Approach

The benefit of $P(z)$ as given by equation (5.11) lies in the fact that the pair of complementary filters $H\left(z^M\right)$ and $H_c\left(z^M\right)$ are acquired from the model pair of complementary filters $H(z)$ and $H_c(z)$ by replacing z^{-1} with z^{-M} as shown in Figure 5-9 (a)-(c). As we described previously, the above substitutions make $H\left(z^M\right)$ and $H_c\left(z^M\right)$ periodic with a periodicity of $2\pi/M$ without increasing the number of multipliers, resulting in FIR filters with a sparse impulse response with only every *M*th impulse value being nonzero. The most important benefit of this approach is focused on the fact that $H\left(z^M\right)$ and $H_c\left(z^M\right)$ provide several transition bands of width $1/M$ times the transition bandwidth of the original filters $H(z)$ and $H_c(z)$. By using the masking filter approach, one of these sharper transition bands can be used as the transition band of

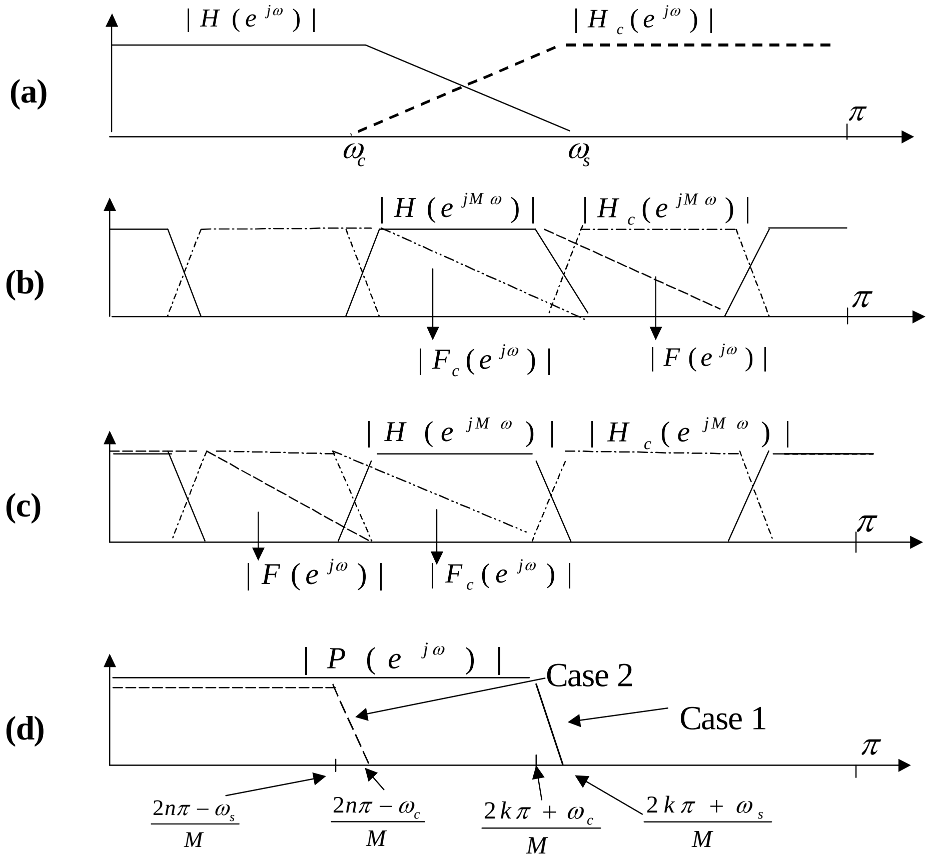


Figure 5-9 The Magnitude Response of the Frequency Response Masking Technique

the overall filter as shown in Figure 5-9 (d). Since the order of a linear phase FIR filter is inversely proportional to the transition bandwidth in equation (1.2), we can generate an FIR filter that saves many more multipliers when compared to the conventional FIR filter with the same transition bandwidth.

Due to the periodic responses of $H\left(z^M\right)$ and $H_c\left(z^M\right)$, they cannot be used alone. To generate the desired filer response, the masking filters $F\left(z\right)$ and $F_c\left(z\right)$ must be combined together. There are two cases to be considered as shown in Figure 5-9 (b) and (c). The band edges of the desired filter are determined by either of the cases as shown in

Figure 5-9 (d). Hence, through the proper selection of the number of the duplication band, i.e. the index *k* or *n* in the Figure 5-9 (d), we an generate our desired arbitrary bandwidth filters.

5.2.2 Proposed Filter Implementation

In general, the masking filters $F(z)$ and $F_c(z)$ are designed in such a manner that in the passband these filters approximate unity and the edges approximate the passband of the selected periods of the model filters $H\left(z^M\right)$and $H_c\left(z^M\right)$. Consequently the overall passband approximates the specification as desired. In the stopband, the masking filters are designed to attenuate the unwanted passband and transition bands.

For the masking filter design, our main interest concentrates on those cases in which the magnitude response of the masking filter, $F(z)$, should provide a narrow transition band and great attenuation of the unwanted replicas of the desired passband. From Figure 5-9(b), it is noted that the transition bandwidths of the masking filters are governed by the bandwidth and the transition bandwidth of the interpolated model filter. When the desired transition band is very narrow, the interpolator factor M is required to be large. In this case, the transition band for the two masking filters is also obliged to be very sharp, leading to a high order for the masking filters. To overcome this complication, the IFIR filter introduced in the previous section is considered as the masking filter, which allows us to design a single low order filter instead of a large order filter and offers large attenuation as well. In addition, it is allowable that the magnitude of $F(z)$ has a slight ripple in the passband, since the equalizer can compensate for some of this deviation.

Therefore, to some extent, a little ripple can be ignored during the design of the model filter and the masking filter design. This advantage makes the design of the proposed prefilter much easier.

5.3 Systematic Design Procedure

As discussed previously, the masking filter will provide a large attenuation, and the equalizer will help to correct the passband ripple. Here, we will describe how to obtain a specified narrow transition band filter by properly determining the parameters as described above with low complexity. The design procedure outlined here is for a lowpass filter.

5.3.1 Selection of the Model Filter

Assume a desired narrow transition band filter has the passband edge at $\omega_{d,c}$ and the stopband edge at $\omega_{d,s}$. To choose the model filter, we must determine the factor M of the interpolated filter $H\left(z^M\right)$, the index number of the passbands *k* or *n* as depicted as shown in Figure 5-9(d), and the passband and stopband edge frequencies, ω_c and ω_s. For a given *M*, there are two possible cases. The band edges of the model filter can be determined by either of the following:

Case 1 (as shown in Figure 5-9 (b)): If the passband width of $F\left(z\right)$ is wider than that of $F_c\left(z\right)$, the transition band of the desired filter is given as:

$$\frac{2k\pi + \omega_c}{M} \le \omega \le \frac{2k\pi + \omega_s}{M}, \tag{5.13}$$

which indicates that $\frac{2k\pi}{M} < \omega_{d,c}$, where, the integer k refers to the k^{th} duplicated period of $H\left(z^M\right)$, so

$$k = \left\lfloor M\omega_{d,c} / 2\pi \right\rfloor. \tag{5.14}$$

In other words, we can select the passband edge and stopband edge of the model filter $H(z)$ as follows:

$$\begin{matrix} \omega_c = M\omega_{d,c} - 2k\pi \\ \omega_s = M\omega_{d,s} - 2k\pi \end{matrix}. \tag{5.15}$$

Case 2 (shown as Figure 5-9 (c)): If the passband width of $F_c(z)$ is wider than that of $F(z)$, the desired filter transition band is governed by:

$$\frac{2n\pi - \omega_s}{M} \le \omega \le \frac{2n\pi - \omega_c}{M}; \tag{5.16}$$

which indicates that $\frac{2n\pi}{M} > \omega_{d,s}$, where, the integer n means the n^{th} duplicated period of $H\left(z^M\right)$, and

$$n = \left\lceil M\omega_{d,s} / 2\pi \right\rceil. \tag{5.17}$$

That is, the passband edge and the stopband edge of the model filter $H(z)$ can be set as the following frequency:

$$\begin{matrix} \omega_c = 2n\pi - M\omega_{d,s} \\ \omega_s = 2n\pi - M\omega_{d,c} \end{matrix}. \tag{5.18}$$

In fact, the solution of these parameters is not unique. In general, *M* is not given. We can select a proper value of *M* within the constraint by:

$$M \le \mathrm{int}\{\pi/(\omega_{d,s} - \omega_{d,c})\}, \tag{5.19}$$

where

$$M \ge \left(\omega_s - \omega_c\right)/\left(\omega_{d,s} - \omega_{d,c}\right). \tag{5.20}$$

In short, the selection of the parameters must obey the overall criterion that minimizes the number of the operations for the whole realization.

5.3.2 Masking Filter Design

Case 1: As shown in

Figure 5-9 (b), set the passband edge of the masking filter, $\omega_{m,c}$ at the desired passband edges, $\omega_{d,c}$. The stopband edge of the masking filter, $\omega_{m,s}$, can be set at

$$\omega_{m,s} = \frac{2(k+1)\pi - \omega_s}{M}, \tag{5.21}$$

and the passband edge of the complementary masking filter, $\omega_{c,c}$, should be chosen at

$$\omega_{c,c} = \frac{2k\pi - \omega_c}{M}, \tag{5.22}$$

while the stopband edge of the complementary masking filter, $\omega_{c,s}$, is set at

$$\omega_{c,s} = \frac{2k\pi + \omega_s}{M}. \tag{5.23}$$

Case 2: As shown in Figure 5-9 (c), set the passband edge of the complementary masking filter, $\omega_{c,c}$ at the desired passband edges, $\omega_{d,c}$. The stopband edge of the complementary masking filter, $\omega_{c,s}$, is set at

$$\omega_{c,s} = \frac{2n\pi + \omega_c}{M} , \tag{5.24}$$

and the passband edge of the masking filter, $\omega_{m,c}$, should be chosen at

$$\omega_{m,c} = \frac{(2n-1)\pi + \omega_s}{M} , \tag{5.25}$$

while the stopband edge of the complementary masking filter, $\omega_{m,s}$, is set at

$$\omega_{m,s} = \frac{2n\pi - \omega_c}{M} . \tag{5.26}$$

The order of the corresponding masking frequency filter N is determined so that it meets the attenuation specification.

For narrowband masking filter design, the proposed structure is acceptable, but a little modification is desirable. Many methods, such as an RRS structure, can be used to produce an efficient prefilter. For generality, we apply the single model IFIR filter introduced in the last section to create the masking filter. By cascading the several instances of the same model filter, but with different interpolated factors described in equation (5.3), the undesired band edges can be removed. Thus a narrow transition band and large attenuation for the masking filter can be generated.

According to the above analysis, a modified prefilter-equalizer structure is provided to synthesize an area-efficient filter with arbitrary bandwidth. The design procedure is summarized as follows:

Step 1: Determine the factor of the interpolated filter, M , by equation (5.19-5.20) such that the interpolated filter $H\left(z^M\right)$ and $H_c\left(z^M\right)$ can be generated. Then select the correct k^{th} or n^{th} passband based on equations (5.14 or 5.17) to make sure that the designed filter meets the required specifications.

Step 2: Design a model lowpass FIR filter, $H(z)$, in a simple form with a low order and a wide transition band. The band edges can be chosen by equation (5.15) and (5.18).

Step 3: Design the masking filters, $F(z)$ and $F_c(z)$, from the recommended IFIR filters to meet the requirements as shown in equations (5.21) to (5.26).

Step 4: Configure the desired prefilter in the structure of Figure 5-7.

Step 5: Obtain the equalizer by using the modified Parks-McClellan method and determine the variable precision coefficients by our proposed algorithm in chapter 2.

5.3.3 Design of Examples

Example 5.3: In this example, we will illustrate the efficiency of the filter obtained by applying the proposed method compared to those acquired using the other methods. Consider the same specifications of a linear phase FIR lowpass filter as shown in [51]:

passband edge: $\omega_p T = 0.2\pi$, stopband edge: $\omega_s T = 0.205\pi$

passband ripple: $\delta_p = 0.1\,dB$, stopband ripple: $\delta_s = -40\,dB$.

For the conventional direct FIR filter design, the minimum order to meet the given specifications is 761, and requires 761 adders and 381 multipliers.

For the introduced method of Rong Huan [51], the overall number of multipliers is minimized when the filter is designed with interpolator factor $L = 21$ in case 1. The best solution is obtained by the model filter band edges set at $\omega_p T = 0.4\pi$ and $\omega_s T = 0.442\pi$ with 129 multipliers and 254 adders.

For the proposed approach, the overall number of multipliers is minimized to be 100 and the number of adders is 198 as follows:

Step 1: According to equations (5.19) and (5.20), we choose M to be less than 20 and if M is about 10, we should get a good solution. In this example, $M = 12$ is taken and case 1 is adopted, which leads to $K = 1$ by equation (5.14).

Step 2: The band of edges of the model filter can be designed by equation (5.15), $\omega_c T = 0.4\pi$ and $\omega_s T = 0.46\pi$. To meet the specification, the order of the model filter is chosen to be 76.

Step 3: From equation (5.21)-(5.23), the band edges of the masking filters $F(z)$ and $F_c(z)$ are obtained as shown in Table 5.1. Their orders are designed as 63 and 57, respectively.

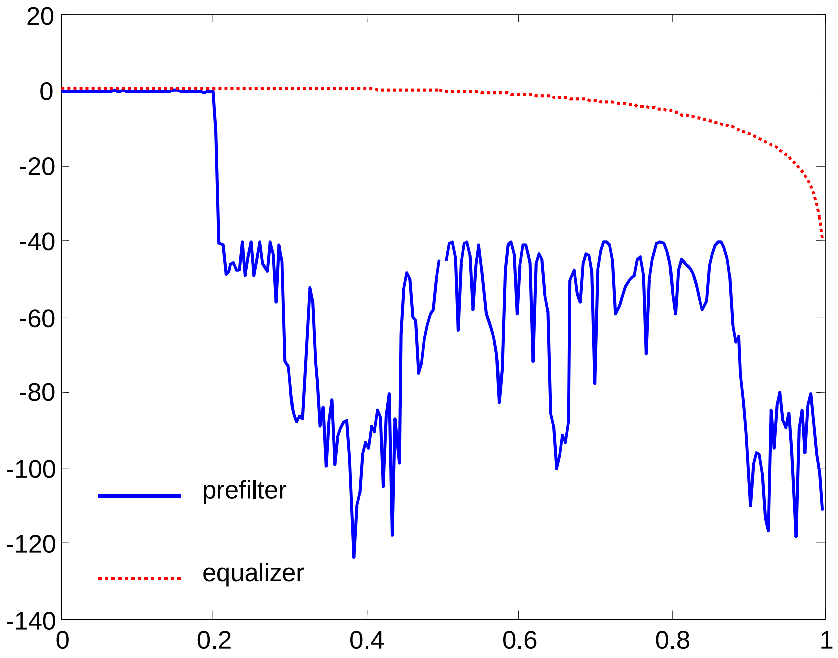
At this point, the overall prefilter is designed. Figure 5-10 (a) is the magnitude frequency response of the prefilter with the ripple details in the passband shown as Figure 5.10 (b). We observe that the ripple of such a prefilter is beyond the specified filter requirements. In order to compensate for the distortion, an equalizer of order 3 is cascaded. Thus the sum of the proposed filter lengths is 199. Figure 5.11 presents the overall filter designed by the proposed approach, as well as the filter obtained by the conventional method.

If all filter coefficients are quantized to 10 bits, the proposed filter has slight deformation shown in Figure 5.12 (a), which can be modified by increasing the filter length to 209. For further area reduction, the coefficients of the proposed filter can be quantized with just a few CSD digits using our proposed algorithm due to its low

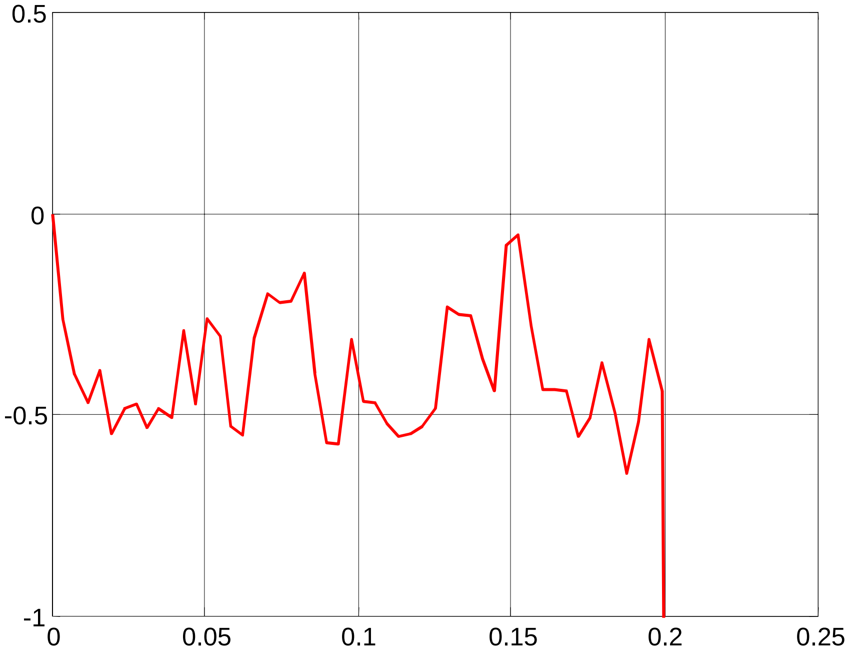
Table 5-1 Description of the proposed filter

Parameters	Passband edge	Stopband edge	Filter length ($M = 12$)
$H(z)$	0.4π	0.46π	76
$F(z)$	0.198π	0.285π	57
$F_c(z)$	0.133π	0.205π	63

sensitivity characteristics. However, if the conventional method is used and the coefficients are quantized to 10 bits, the distortion is considerable, as shown in Figure 5.12 (b). In order to satisfy the given specifications, the order of the conventional method must be increased to at least 856 with 12bits/coefficient or the wordlength of the coefficients must be increased to 14bits/coefficient with an order of 761.

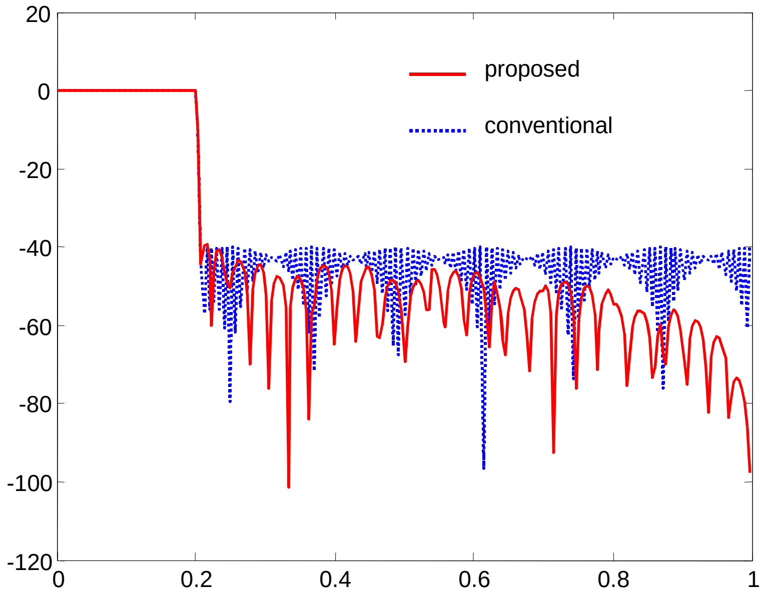


(a) The Magnitude Frequency Response of Prefilter as well as Equalizer

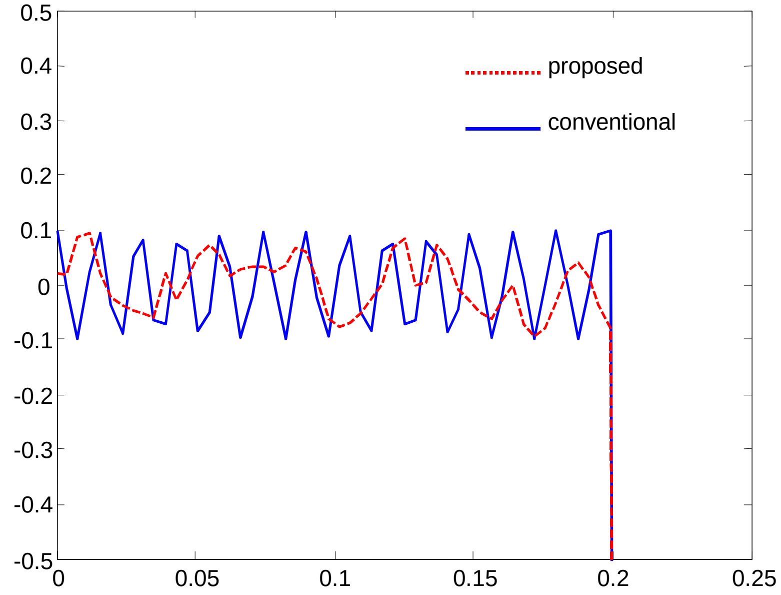


(b) The Detailed Respond of the Passband of the Prefilter

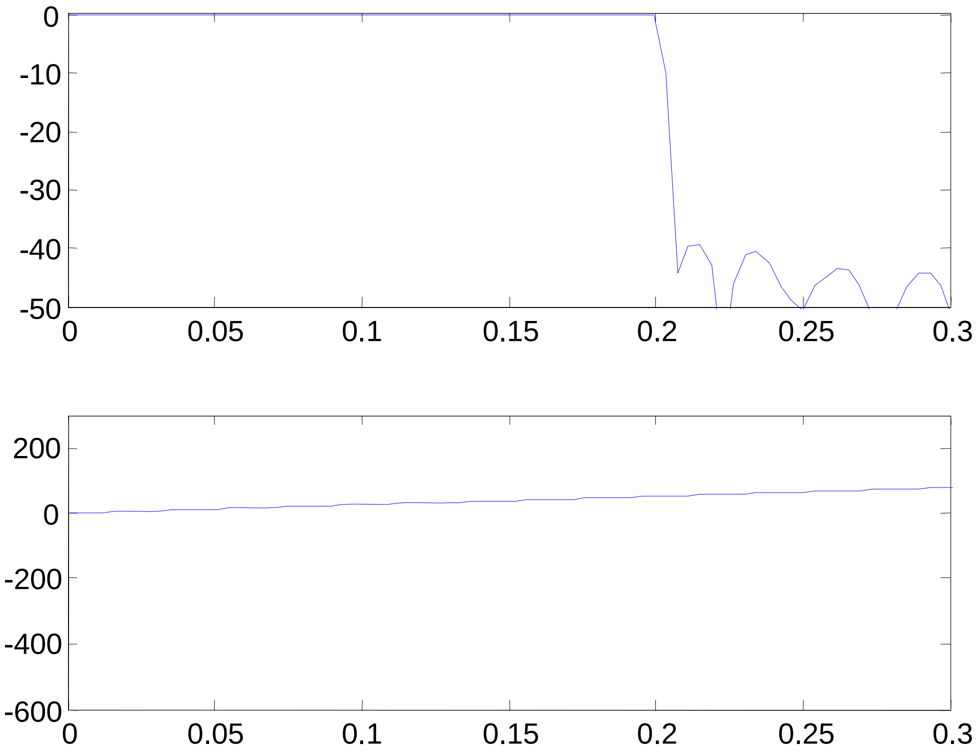
Figure 5-10 The Result of Prefilter with Bandwidth $\omega_p = 0.2\pi$



(a) The Magnitude Frequency Response

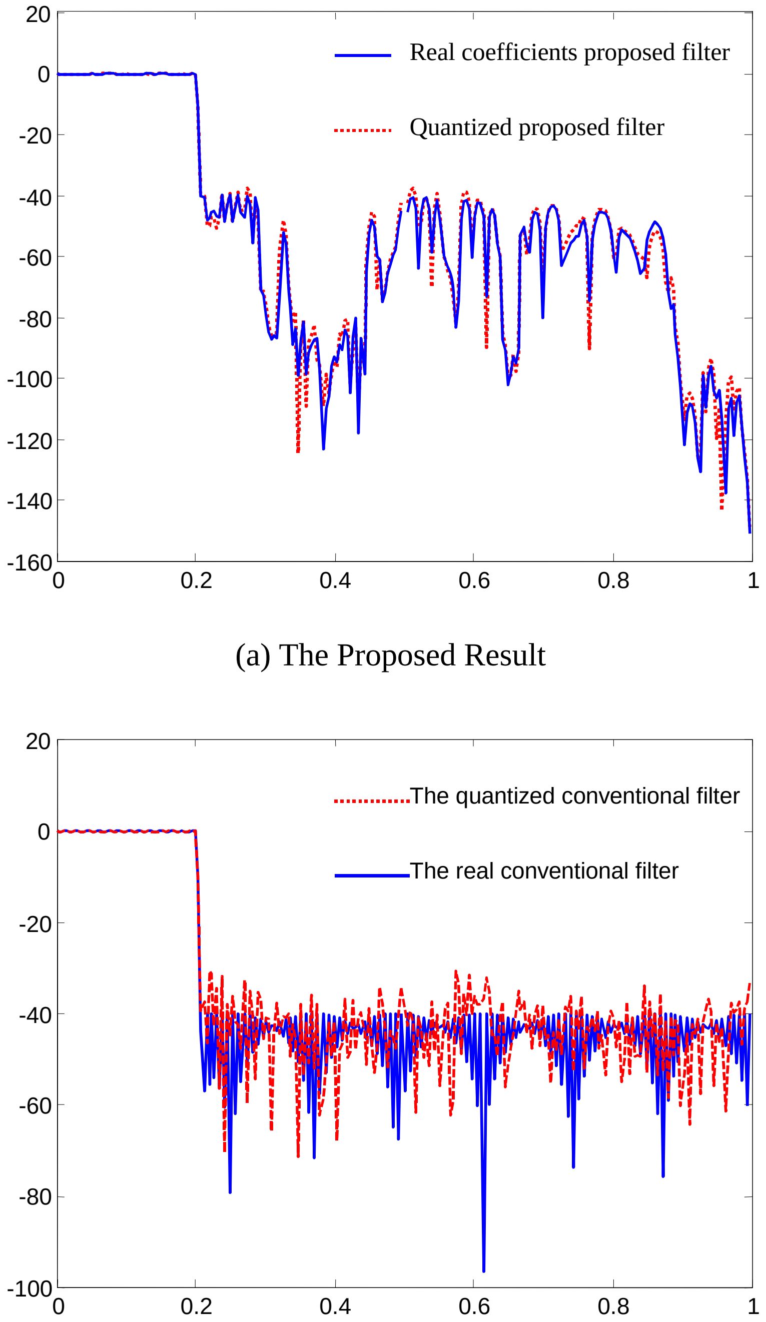


(b) The Details of the Passband



(c) The magnitude and phase responses of on passband of the proposed filter

Figure 5-11 The Result of Proposed Filter with Bandwidth $\omega_p = 0.2\pi$ **,** as well as the Filter from the Conventional Design



(a) The Proposed Result

(b) The Conventional Result

Figure 5-12 The Comparison of the Quantized Result (10Bits/Coefficient)

Example 5.4**:** In order to demonstrate the efficiency of the arbitrary bandwidth filter design, consider a desired wide bandwidth FIR lowpass filter with the following specifications: pass-band edge at 0.6π, stop-band edge at 0.605π, and the ripples at the passband and stopband are $0.1\,dB$ and $-40\,dB$ respectively.

In this case, the required order to meet the specification using the conventional direct method is 761. If we use our proposed method, we can achieve a low complexity filter with a total order of 178. According to equations (5.19) and (5.20), we choose M to be about 10 and less than 20. In this example, $M = 11$ is taken and case 1 is adopted, leading to $K = 3$ by equation (5.14).

The band of edges of the model filter can be designed by equation (5.15), $\omega_c T = 0.6\pi$ and $\omega_s T = 0.655\pi$. To meet the attenuation requirement, the order of the model filter is chosen to be 72.

The edges of the masking filters $F(z)$ and $F_c(z)$ are obtained from equation (5.21)-(5.23), and their order are chosen as to be 63 and 43, respectively. The band edges of the filters are given in the Table 5.3.

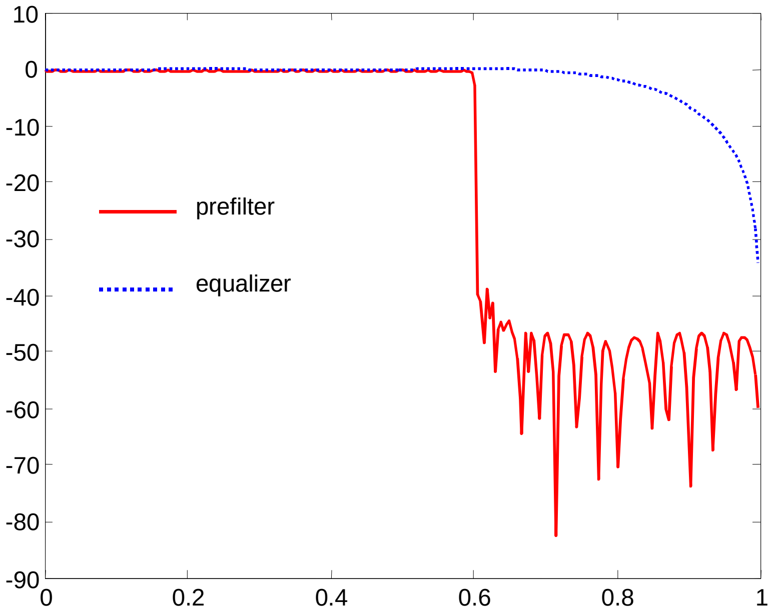
Figure 5-13(a) presents the magnitude frequency response of the overall prefilter and the ripple in detail is shown in Figure 5.13(b). It is noted that the ripple of such a prefilter

Table 5-2 Description of the proposed wideband filter with $\omega_c T = 0.6\pi$

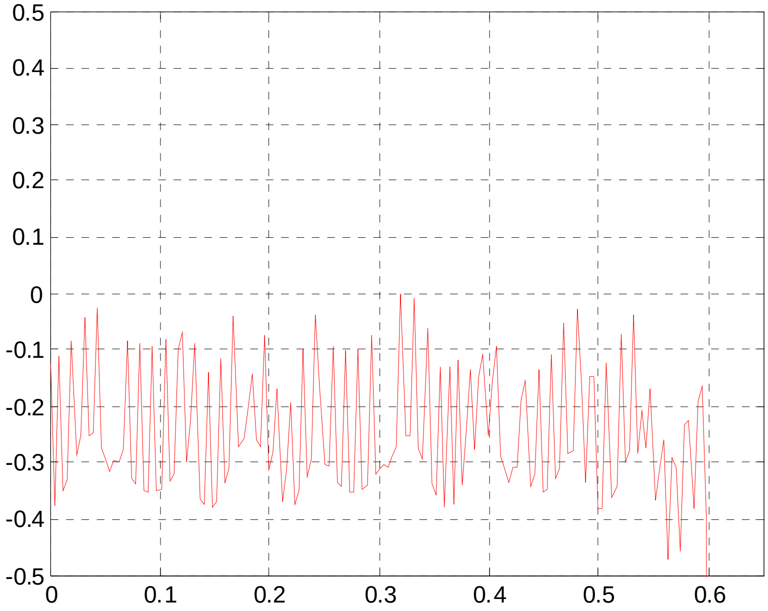
Parameters	Passband edge	Stopband edge	Filter length ($M = 12$)
$H(z)$	0.6π	0.655π	72
$F(z)$	0.595π	0.672π	63
$F_c(z)$	0.49π	0.605π	43

is beyond the specified filter requirement. In order to compensate for the distortion, an equalizer with an order of 7 is cascaded. Figure 5.14 shows the overall filter designed by the proposed approach, as well as the filter obtained by the conventional method.

If all filter coefficients are quantized to 10 bits, the proposed filter has a slight deformation shown as Figure 5.15 (a), while still satisfying the specifications. Thus the filter can be quantized to 10bits/coefficient. However, if the conventional method is used and the coefficients are quantized to 10 bits, the distortion is considerable, as shown in Figure 5.15 (b). In order to satisfy the given specification, the order has to be enlarged to 858, or the wordlength of the coefficients must be increased to 14bits/coefficient.

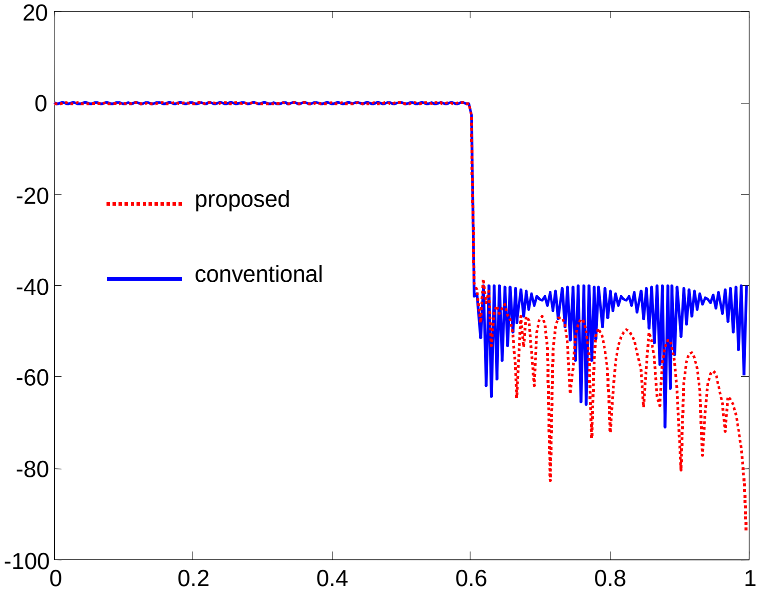


(a) The Magnitude Frequency Response of Prefilter as well as Equalizer

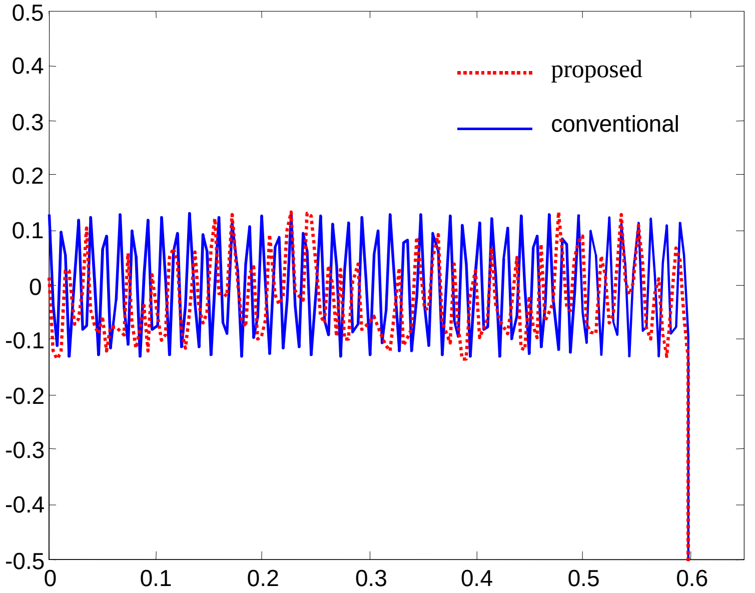


(b) The Details of the Passband of the Prefilter

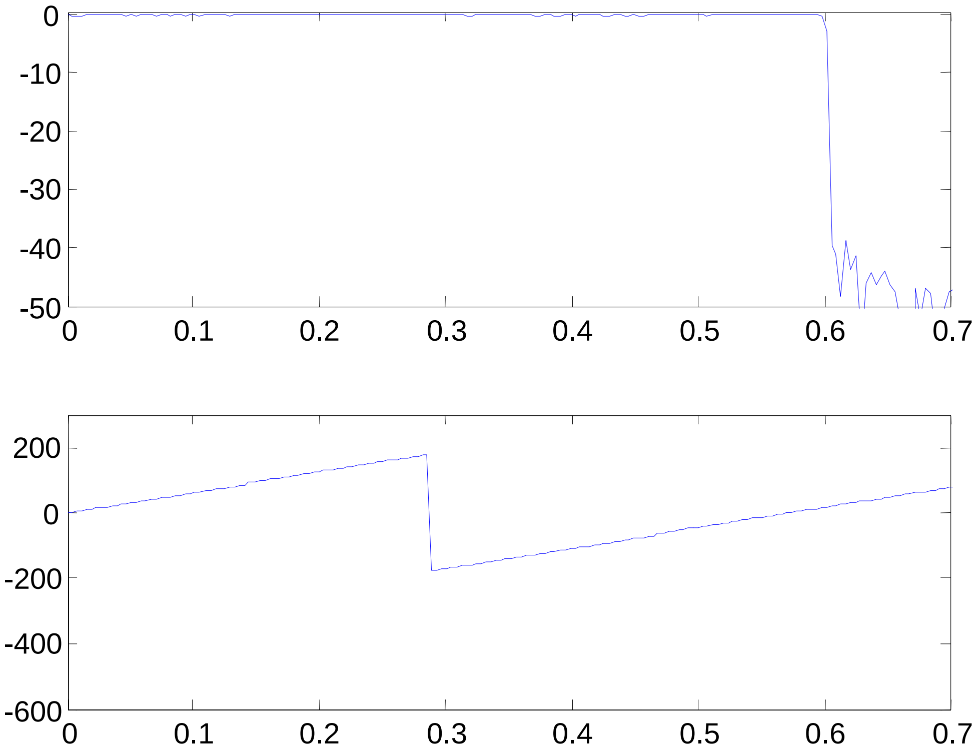
Figure 5-13 The Result of Prefilter with Bandwidth $\omega_p = 0.6\pi$



(a) The Magnitude Frequency Response

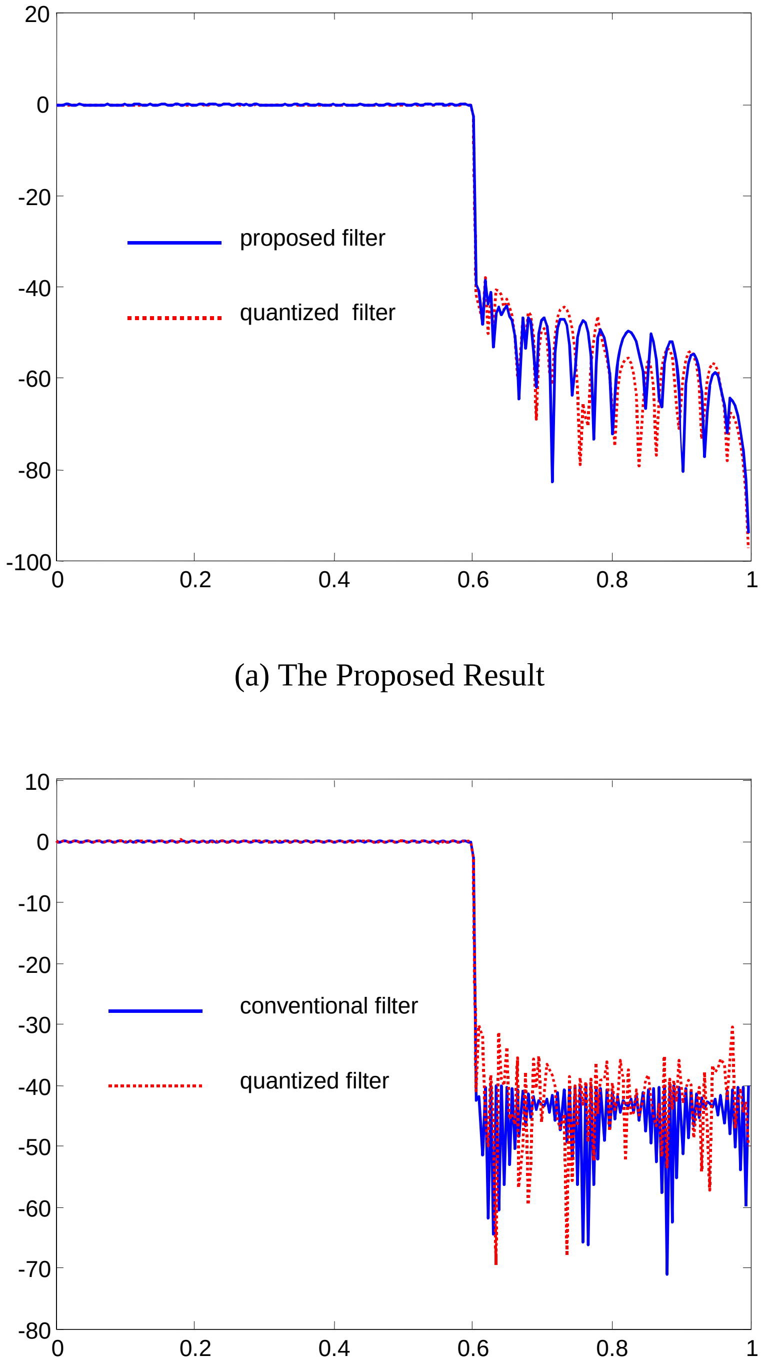


(b) The Details of the Passband



(c) The magnitude and phase responses on the passband of the proposed filter

Figure 5-14 The Result of Proposed Filter with Bandwidth $\omega_p T = 0.6\pi$, as well as the Filter from the Conventional Design



(a) The Proposed Result

(b) The Conventional Result

Figure 5-15 The Comparison of the Quantized Filter (10Bits/Coefficient) with Bandwidth $\omega_p T = 0.6\pi$

5.4 Concluding Remarks

In this chapter, a proposed prefilter structure is used to improve the desired filter performance and reduce the complexity required in implementing an FIR digital filter, which combines the proper interpolator and the masking technique. The attractiveness of the proposed structure lies in that our improved two-level design allows an arbitrary width filter to be realized in a simple way. Our proposed method can be summarized in two steps. The first step is targeted at improving the prefilters based on the frequency masking technique and aims to provide a sharp transition-band as well as increasing the stopband attenuation. The second step is targeted at compensating the ripple in the passband and reducing the complexity of equalizers required. When this method is combined with our proposed variable precision algorithm, we can produce a filter with a few bits per coefficient. The resulting filter is shown to be more efficient. A systematic design procedure for the efficient cascading FIR filters is given, leading to a good design with low coefficient sensitivity and low order, while still satisfying the specifications. This allows the area of the resulting hardware to be reduced.

Chapter 6 Conclusions and Future Work

Conclusions

In this dissertation, we have conducted a systematic research on the design and implementation of high performance, area efficient FIR filters. In order to reduce the number of arithmetic operations, we study the possibilities from two aspects: developing algorithm approach and improving filter structure to reduce the hardware complexity.

We developed a variable precision algorithm for FIR filters, which is founded on our sensitivity analysis. The algorithm is proposed to reduce the wordlength of the coefficients and/or the number of nonzero bits of the coefficients by eliminating bits that are not necessary. The sensitivity analysis is used to predict the variable precision of the quantized coefficients to meet the filter specifications without any degradation of the filter performance. Furthermore, the proposed algorithm is used with CSD representation to reallocate different nonzero digits. In order to increase the proposed algorithm efficiency, a scaling algorithm and CSD number representation are incorporated. The examples presented in this dissertation show that using variable precision coefficients results in significant reduction in filter complexity compared with the uniform wordlength method. From the viewpoint of implementation, the use of variable precision coefficients can reduce the number of operations and storage blocks in the chip area, which is highly applicable to the design of multiplierless filters in VLSI implementations.

One of the most important parts of our design is to synthesize an arbitrary bandwidth FIR filter with the prefilter and the equalizer cascade structure. The proposed prefilter is designed to provide large stopband attenuation and wide band response with relatively low complexity. Since the prefilter achieves most of the requirements, the equalizer can be realized in much lower order. For a narrow filter design, an RRS prefilter presents superior performance in terms of the hardware implementation. In order to further extend the application to arbitrary wide band filters, two other kinds of prefilters, specifically the Chebyshev polynomial prefilter and halfband prefilter, are investigated. The Chebyshev polynomial prefilter can offer large attenuation with very low order. However, for a single Chebyshev prefilter, the transition band may be too wide and some perturbation of the passband will occur if great attenuation is required. Under such situations, the distortion of the filter performance will be difficult to completely remove. To overcome the weakness, we develop a halfband prefilter. Moreover, the coefficient sensitivity of such a structure is studied, and the numerical instances also show that low coefficient sensitivity is obtainable in this structure. Each coefficient can be quantized with reduced wordlength, or the CSD representation can be implemented using only one or two nonzero digits. Thus, the resulting hardware area can be greatly reduced. The design examples provided in this dissertation demonstrate the effectiveness of our design methods.

In order to improve the desired filter performance and reduce the complexity, we also have combined the proper interpolator and the masking technique and applied them to our proposed structure. The proposed filter based on the frequency masking technique is targeted at improving the prefilters, which provides a sharp transition band and the large stopband attenuation and allows any arbitrary bandwidth filter to be realized in a simple way. When the proposed structure is combined with our proposed variable precision algorithm, the complexity of the equalizers is reduced. A systematic design procedure for the efficient cascading FIR filters is presented, which leads to a good design with low coefficient sensitivity and small order while still satisfying the specification. Thus the resulted hardware area is reduced.

Future Work

We have identified several areas for future work. We want to explore the possibility of using an IIR structure to implement the FIR filter. The idea is to design a low order IIR filter using filter approximation techniques that can ensure the resulting low order IIR filter reflects the features of the FIR filter and satisfies the original design specifications. In general, the IIR filter is constructed by employing the iterative arithmetic structure and the order of the filter is relatively low. Hopefully, the computational complexity of the resulting FIR filter will be low compared with that of the existing FIR direct methods.

For arbitrary bandwidth filters, more studies should be conducted in the future for more efficient methods. For example, tunable FIR filters can be designed with the same hardware requirements as that of their prototypes. Thus, if a narrow prototype filter is implemented with a savings in area hardware, arbitrary bandwidth filters can be realized in the same reduced area by using the tunable filter.

Additionally, we want to develop more computationally efficient layout methods.

The area costs for high performance digital filters depend not only on analyzing an elegant algorithm and optimizing its various parameters, but also considering details related to transistors and wires. In an ideal state, we suppose that a delay operation costs little area in the hardware. However, in VLSI technology, signal propagation delays over long wires usually occupy significant space, which force the designer to alter the way in which an area efficient arithmetic circuit can be desgined. Our future work could consider both the logic gate level and the layout level.

We would like to realize our FIR filter designs in various devices currently available. Commercially, we would analyze the effect of the different proposed methods in terms of the consumed area for different devices. Determining the relationship of the design methods and the various device types in terms of area cost would be highly desirable.

Many digital signal processing applications are arithmetic intensive and cost sensitive, thus requiring innovative solutions for cost effective implementation.

References

[1] Keshab K. Parhi, *VLSI Digital Processing Systems Design and Implementation*, John & Wiley Sons, Inc., 1999.

[2] Douglas L. Jones, “Leaning Characteristic of Transpose-Form LMS Adaptive Filters,” *IEEE Trans. Circuits and Syst.-II: Analog and digital signial processing*, vol. 39, no. 10, Oct. 1992.

[3] Vikram Pasham, Andy Miller and Ken Chapman,“Transposed Form FIR Filters”, *Xilinx Application note: Virtex and Virtex-II series*, Jan. 2001.

[4] S.R. Powell and P.M. Chau, “Efficient narrowband FIR and IFIR filters based on powers-of-two sigma-delta coefficient truncation,” *IEEE Trans. Circuits and Syst.II*, vol. 41, pp. 497-505, Aug. 1994.

[5] Behrooz Parhami, *Computer Arithmetic Algorithm and Hardware Designs*, Oxford University Press, Inc., 2000.

[6] O. Gustafsson, H. Johansson, and L. Wanhammar, “Design and efficient implementation of single filter frequency-response masking FIR filters,” *European Signal Processing Conference*, Tampere, Finland, Sep 2000.

[7] “Estimating FPGA requirements for DSP applications”, *Technical note of Hunt Engineering Ltd*. May 2002.

[8] H. Samueli, “An improved search algorithm for the design of multiplierless FIR filters with power of two coefficients,” *IEEE Trans. Circuits and Syst.*, vol. 36, no. 7, pp.1044-1047, Jul.1989.

[9] Gianfranco Bilardi and Franco P. Preparata, “Memory requirements of first–order digital filters,” *IEEE Trans. Circuits and Syst.*, vol. 36, no. 6, 1992.

[10] Andrew G. Dempster and Malcolm D. Macleod, “Use of Minimum Adder Multiplier Block in FIR Digital Filters,” *IEEE Trans. Circuits and Syst.*, vol. 42, no. 9, 1995.

[11] Joseph B. Evans, “Efficient FIR filter architectures suitable for FPGA implementation,” *IEEE Trans. Circuits and Syst.*, vol. 41, no. 7, pp. 490-493, July 1994.

[12] H. Shaffeu, M. M. Jones, “Improved design procedure for multiplierless FIR digital filters,” *Electron. Lett.*, vol. 27, no. 13, pp.1142-1144, Jun. 1991.

[13] Hwan Rei Lee and Chein Wei Jen, “A new hardware efficient architecture for programmable FIR filters,” *IEEE Trans. Circuits and Syst*., vol. 43, no. 9, pp. 637-644, Sep. 1996.

[14] Arda Yurdakul, “A synthesis tool for the multiplierless realization of FIR based multirate DSP systems,” *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 69-72, May. 2000.

[15] Michael A. Soderstrand and Louis G. Johnson, “Reducing hardware requirement in FIR filter design,” *ICASSP '00. Proceedings*, vol. 6, pp. 3275-3278, 2000.

[16] J. W. Adams and A. N. Wilson, “A new approach to FIR digital filters with fewer multipliers and reduced sensitivity,” *IEEE Trans. Circuits and Syst.*, vol. CAS-30, no. 5, pp. 277-283, May 1983.

[17] Quangfu Zhao and Yoshiaki Tadokoro, “A simple design of FIR filters with powers of two coefficients,” *IEEE Trans. Circuits and Syst.*, vol. 35, no. 5, pp. 566-570, May 1988.

[18] Min Soo Park and Woojin Song, “ A new design method of 2-d linear phase FIR filters with finite precision coefficients,” *IEEE Trans. Circuits and Syst.*, vol. 41, pp. 478-482, July 1994.

[19] Scott R. Powell and Paul M. Chau, “Efficient Narrowband FIR and IFIR filters based on power of two sigma-delta coefficient truncation,” *Analog and digital signal processing*, vol. 41, no. 8, Aug. 1994.

[20] C.L. Chen, K.Y. Khoo and A.N. Willson, Jr, “Improved polynomial-time algorithm for designing digital filter with power-of-two coefficients,” in *Proc. of IEEE International Symposium on Circuits and Systems*, (Seattle,WA), May 1995.

[21] S. Summerfield and C. K. Lu, “Design and VLSI architecture and implementation of wave digital filters using short signed digit coefficients,**”** *IEE Proceedings on Circuits, Devices and Systems*, vol.143, pp. 259-266, Oct. 1996.

[22] Lijljana D.Milic, Miroslav D. Lutovac,“ Design of multiplierless Elliptic IIR filters,” *IEEE international conference on Acoustics, Speech, and Signal Processing*, vol.3, pp. 2201-2204, 1997.

[23] R. E. Crochiere, “A new statistical approach to the coefficient wordlength problem for digital filters,” *IEEE Trans. Circuits and Syst.*, vol. 22, no.3, pp.190-196, 1975.

[24] C. T. Mullis and R. A. Roberts, “Synthesis of minimum round-off noise fixed point digital filters,” *IEEE Trans. Circuits and Syst.*, 23: 5551-162, 1976.

[25] S. Y. Hwang, "Minimum uncorrelated unit noise in state-space digital filtering," *IEEE Trans. Acoustics, Speech, and Signal Processing*, 25:273-281, 1977.

[26] Casper W. Barnes and Shinji Shimmaka, "Finite word effects in block-state realizations of fixed point digital filters," *IEEE Trans. Circuits and Syst.*, vol. CAS-27, no. 5, 1980.

[27] Ljiljana D. Milic and Miroslav D. Lutovac, "Design of multiplierless elliptic IIR filters with a small quantization error," *IEEE Trans. Signal Processing*, vol. 47, pp. 469-479, Feb. 1999.

[28] Linda S. DeBrunner, Victor DeBrunner, Xiaojuan Hu, Olivier Demuynck and Andi Swartztrauber, "A reduced-space half-band filter design on an Actel FPGA," 35th *Asilomar Conference on Signals, Systems and Computers*, vol.2, pp.1237-1240 Nov. 2001.

[29] J. W. Adams and A. N. Wilson, "Some efficient digital prefilter structures," *IEEE Trans. Circuits and Syst.*, vol. CAS-31, no. 3, pp.260-266, Mar 1984.

[30] P.P. Vaidyanathan and G. Beitman, "On prefilters for digital FIR filter design," *IEEE Trans. Circuits and Syst.*, vol. CAS-32, no. 5, pp.494-499, May 1985.

[31] "Designing FIR filters with Actel FPGAs," *Actel application note,* Feb. 2003.

[32] J. J. Nielsen, "Design of linear-phase direct form FIR digital filters with quantized coefficients using error spectrum shaping techniques," *IEEE Trans. Circuits and Syst.*, vol. 37, no. 7, pp. 1020-1026, Jul. 1989.

[33] A. G. Dempster, M. D. Macleod, "Variable statistical word-length in digital filters," *IEE Pro.-Vis. Image Signal Process*, vol.143, no.1, pp.62-65, Feb1996.

[34] Linda S. DeBrunner, Victor DeBrunner and Paul Pinault, “Variable wordlength IIR filter implementations for reduced space designs,” *Workshop on Signal Processing Systems*, Lafayette, Louisiana, pp. 326-335, October 2000.

[35] DeBrunner, V., DeBrunner L.S., and Xiaojuan Hu, “Using variable length coefficients to design low-space FIR filters for FPGAs,” *36th Asilomar Conference*, vol.1, pp. 346-349, Nov. 2002.

[36] Stefan Johansson, Shousheng He, and Peter Nilsson, “Wordlength optimization of a pipelined FFT processor,” *42nd Midwest Symposium on Circuits and Systems*, vol. 1, pp. 501-503, 2000.

[37] Y. Neuvo, “Interpolated finite impulse response filters,” *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-32, pp. 563-570, June 1984.

[38] Y.C. Lim, “Frequency-response masking approach for the synthesis of sharp linear phase digital filters,” *IEEE Trans. Circuits and Syst*., vol. CAS-33, pp. 357-364, Apr1986.

[39] G.Rajan, Y.Neuvo and S.K.Mitra, “On the design of sharp cutoff wide-band FIR filters with reduced arithmetic complexity,” *IEEE Trans. Circuits and Syst*., vol. 35, no. 11, pp. 1447-1454, November 1988.

[40] Xiaojuan Hu, Linda DeBrunner, and Victor DeBrunner, “An Efficient Design for FIR Filters with Variable Precision,” *IEEE International Symposium on Circuits and Systems, ISCAS’02*, vol.4, pp. IV-36 5-IV 368, May 2002.

[41] Sanjit K. Mitra, *Digital signal processing: a computer-based approach*, The McGraw-Hill companies,Inc., 1998.

[42] Kah Howe Tan and Wen Fung Leong, “Public domain MATLAB program to generate highly optimized VHDL for FPGA implementation,” *ISCAS 2001*, vol. 4, pp. 514-517, 2001.

[43] D. Ait-Boudaoud and R. Cemes, “Modified sensitivity criterion for the design of powers-of-two FIR filters, ” *Electron. Lett.*, vol. 29, no. 16, pp. 1467-1469, Aug. 1993.

[44] A. A. Beex and Vitor E. DeBrunner, “Reduced sensitivities of direct form digital filter structures by increasing system order,” *IEEE Trans. Circuits and Syst.*, vol. 36, no. 3, pp. 438-442, Mar. 1989.

[45] P. P. Vaidyanathan and Truong Q. Nguyen, “A ‘TRICK’ for the Design of FIR Half-Band Filters,” *IEEE Trans. Circuits and Syst.*, vol. CAS-34, no. 3, March 1987.

[46] Lars Wanhammar, *DSP Integrated Circuits*, Academic Press, 1999.

[47] Richard I. Hartley and Keshab K. Parhi, *Digit-serial computation*, Kluwer Academic Publishers, 1995.

[48] Alan N. Willson and H. J. Orchard, “A design method for half-band FIR filters,” *IEEE Trans. Circuits and Syst*., vol. 45, no. 1, pp. 95-101, January 1999.

[49] Kaname Jinguji and Manabu Oguma, “Optical Half-band Filters,” *J. Lightwave Technol.*, vol. 18, no. 2, pp. 252-259, Feb. 2000.

[50] Xiaojuan Hu, L.S. DeBrunner, and V. DeBrunner, “Design of space-efficient, wide- and narrow transition-band, FIR filters,” *ICASSP’03*, vol.2, pp.569-572, April 2003.

[51] Rong Huan Yang, Bede Liu, and Yong Ching Lim, “A new structure of sharp transition FIR filters using frequency-response masking,” *IEEE Trans. Circuits and Systs.*, vol. 35, no.8, Aug. 1988.

[52] B. Beliczynski, I. Kale, and G.D.Cain, “Approximation of FIR by IIR digital filters: an algorithm based on balanced model reduction,**”** *IEEE Trans. Signal Processing.*, vol. 40, no.3, pp. 532-542, Mar. 1992.