Efficient Multiplierless Channel Filters for Multi-Standard SDR

Douglas L. Maskell and A.P. Vinod
School of Computer Engineering
Nanyang Technological University, Singapore
as douglas; asvinod@ntu.edu.sg

Abstract

This paper investigates the design of very low complexity multiplierless linear phase FIR filters for use in the channelizer of multi-standard software defined radios. A technique for reducing the hardware complexity of linear phase FIR digital filters which minimizes the adder depth and the number of adders in the multiplier block is introduced and is used to implement a multistage, multi-standard decimating filter. The design reuses components for different communications standards and is thus ideal for use in systems which support dynamic reconfiguration.

1. Introduction

Software Defined Radio (SDR) is evolving as a promising technology in the area of mobile and personal communications. The basic idea of SDR is to replace most of the analogue signal processing in the transceivers with digital signal processing in order to provide the advantage of flexibility through reconfiguration or reprogramming [1]. This will enable different air-interfaces to be implemented on a single generic hardware platform. A multimode SDR handset with dynamic reconfigurability has the promise of integrated services and global roaming capabilities. In order to become reality, the SDR technology requires more research not only from a wireless systems perspective, but also in the areas of novel computing architectures, embedded systems, and methodologies to realize an all-digital reprogrammable radio.

Most of the research to date [2]-[3] has been focused on SDR base stations, which do not have the same constraints on area and power that handsets have. However the ultimate aim of SDR designs is targeted to migrate from base stations to handsets where their true potential can be realized [4]. Current SDR technology is yet to reach the compact form factors that characterize mobile handsets [5] and thus presents an opportunity for further investigation.

This paper is organized as follows. Section 2 examines the SDR front end before focusing on techniques for implementing high-speed digital filters in hardware. Section 3 introduces the filter design problem and the details of the coefficient optimization, subexpression elimination and adder dept reduction techniques. Section 4 introduces the multi-standard DDC filter design for the GSM and CDMA2000 standards, while section 5 presents the results of the design exercise. Section 6 concludes this paper.

2. Problem scenario

This section describes the problem scenario and provides an analysis of issues relating to the implementation of FIR channelizer filters in hardware.

The basic receiver for SDR uses a digital down converter (DDC) to replace the analogue intermediate frequency (IF) demodulator as shown in the system block diagram of figure 1(a). This eliminates the local oscillator (LO)/IF carrier mixing and achieves perfect gain/phase matching between channels. The IF output from the RF mixer is converted to a digital signal using a high speed analogue to digital converter (ADC). This signal is then converted to baseband by the DDC for processing by the digital signal processor (DSP).

The block diagram for the DDC is shown in figure 1(b). The output from the ADC and the programmable numerically controlled oscillator (NCO) are multiplied to centre the desired signal at the baseband location. Interfering signals in adjacent bands are removed using a narrow band (high selectivity) filter built from a sequence of multi-rate filters consisting of a cascaded integrator-comb (CIC) filter which can decimate by a factor from 4 to 64, a compensating FIR filter (CFIR) which decimates by 2, followed by a programmable FIR filter (PFIR). The output of the PFIR can be programmed to decimate by 2 if desired.
Because of the high input data rates (typically 50-100 Msps), the DDC filter implementation needs to be as fast and as efficient as possible. Generally, multiplierless filter implementations are preferred because of their high speed and hardware efficiency. A number of techniques for the efficient implementation of conventional multiplierless FIR filters have been investigated. Generally, these can be classified into two independent sub-areas, namely: discrete optimization of the quantized filter coefficients [6]-[12]; and multiplier block adder reduction [13]-[21].

Discrete coefficient optimization techniques based on mixed integer linear programming using minimax or minimum normalized peak ripple magnitude have been proposed [8], however these techniques are extremely compute intensive and are restricted to filters with only a small number of taps. Other sub-optimal techniques have been proposed. These include random local search [9], tree search [7], simulated annealing [11] and genetic algorithm [12] methods.

Multiplier block reduction has attempted to minimise various cost functions such as adder depth, the number of adders and more recently multiplier block area. Most multiplier block minimisation techniques can be categorized as either graphical [13]-[14] or subexpression elimination [15]-[21]. Common sub-expression (CSE) elimination techniques attempt to minimise the number of additions in the multiplier block by combining terms. Only a few attempts have been made to combine both research areas: that is to minimise the number of additions in the multiplier block (MBA) are the adders used to calculate the multiplierless products of the input signal (x) and the filter coefficient (h_j) in the transposed direct form FIR filter structure. For symmetric FIR filters, only half the equivalent multipliers need be implemented.

3) Structural adders: The structural adders are the adders between the delay stages of the FIR filter structure. The number of structural adders is one less than the number of filter taps (N).

4) Adder depth: The adder depth is the critical path (in terms of addition operations) of the product implementation for any filter coefficient. A multiplication can have different adder depths depending on the implementation structure (eg. a linear or tree implementation). We define the minimum adder depth (D_min) as that which is able to implement all of a particular filter’s coefficients. That is, D_min is dependant on the filter coefficient with the greatest number of nonzero bits, and is defined as:

\[ D_{\text{min}} = \max \left( \lceil \log_2 l_{\text{max}} \rceil \right) \]  

where \( l_{\text{max}} \) is the maximum number of nonzero bits in any of the filter coefficients.

One method of reducing the adder depth is to implement each coefficient (h_j) as a separate binary tree of adders. However, as seen in [24], [25], this results in a greater number of full adders and hence a larger area. Other approaches [13], [21] result in a slightly smaller number of adders but at the expense of adder depth and an increased design time. Adder depth has a significant effect on the latency, and hence the filter throughput, which could be overcome to some extent by adding pipeline stages to the multiplier block adders. However, while this would address the latency issues relating to an increased adder depth, the addition of pipeline latches would result in a significant increase in the filter area counteracting the initial reduction in the number of adders.

3. Definition of terms

As in [19], [20], [24] we introduce some of the terms used through this paper.

1) CSD format: A number \( b_0 b_1 b_2 ... b_{M-1} \) is in CSD format if each \( b_i \) is of the set \{0, 1, -1\} and there are no two consecutive nonzero digits.

2) Multiplier block adders: The multiplier block adders (MBA) are the adders used to calculate the multiplierless products of the input signal (x) and the filter coefficient (h_j) in the transposed direct form FIR filter structure. For symmetric FIR filters, only half the equivalent multipliers need be implemented.

3) Structural adders: The structural adders are the adders between the delay stages of the FIR filter structure. The number of structural adders is one less than the number of filter taps (N).

4) Adder depth: The adder depth is the critical path (in terms of addition operations) of the product implementation for any filter coefficient. A multiplication can have different adder depths depending on the implementation structure (eg. a linear or tree implementation). We define the minimum adder depth (D_min) as that which is able to implement all of a particular filter’s coefficients. That is, D_min is dependant on the filter coefficient with the greatest number of nonzero bits, and is defined as:

\[ D_{\text{min}} = \max \left( \lceil \log_2 l_{\text{max}} \rceil \right) \]  

where \( l_{\text{max}} \) is the maximum number of nonzero bits in any of the filter coefficients.
Our proposed algorithm is presented in figure 2, and is described below. It implements a number of techniques for reducing the adder depth and hence the filter latency and filter area.

3.1. Coefficient Optimization

As mentioned earlier, discrete coefficient optimization techniques based on mixed integer linear programming are extremely compute intensive and are thus restricted to filters with a small to moderate number of taps. Thus, instead of attempting to find a global optimal solution, we find a much faster local solution which satisfies the original frequency specifications, while aggressively minimising both the number of nonzero bits and the coefficient wordlength. The rationale for this is that adder cost estimates [26] show that increasing the filter order in the range around 4% to 10% has little effect on the complexity. The observation was also made that this large range significantly increases the design effort needed to find the optimal solution, and that the designer may choose instead to select a single design from the region knowing it to be near-optimal.

To achieve this we use a random search algorithm which we have implemented in MATLAB. This modified algorithm looks for a solution based upon an initial coefficient word size and an initial number of nonzero CSD bits. If a solution is not found, we incrementally increases the number of nonzero CSD bits followed by an increase in the coefficient wordlength until an acceptable solution is found. Aggressively minimising the number of nonzero bits results in a significant reduction in the adder depth as adder depth is directly related to the number of bits in the coefficient.

3.2. Horizontal CSE

Common subexpression elimination is performed on the filter coefficients [16], based on the four most common 2-bit subexpressions. These subexpressions are [101], [101], [1001] and [10001], and are allocated LSB first. The rationale for using the 4 most common subexpressions rather than the two most common subexpressions as in [16] or higher order combinations, such as 3-bit or 4-bit horizontal SSEs [19], [16], [17], is based on the following observations.

The choice of signals to route within the multiplier block is a trade off between a reduced multiplier block area and an increased routing area. In addition, the available silicon area has increased significantly since Hartley [16] made the observation that the use of subexpression sharing will lead to an increase in the signal routing cost, which could be alleviated by choosing only the two most common subexpressions. SSEs are not considered, as aggressively minimising the number of nonzero digits results in very few 3-bit or 4-bit common subexpressions occurring in the filter coefficients, making it uneconomical to route these rarely used signals through the multiplier block.

3.3. Increasing the Adder Depth up to $D_{\text{min}}$

The minimum adder depth ($D_{\text{min}}$) is calculated. It should be observed that only a small number of the filter’s coefficients are constrained by $D_{\text{min}}$, because $D_{\text{min}}$ is calculated based upon the coefficient value with the most nonzero bits. We now take the remaining bits in the coefficients after common subexpression elimination and implement them (where possible) as a linear array of adders with a depth up to $D_{\text{min}}$. A linear array of adders requires a smaller amount of hardware than a tree implementation [24], [25].
4. Multi-standard DDC filter design

The DDC filter chain consists of a 6 stage CIC filter followed by an inverse-sinc CIC Compensation filter. Two different standards (GSM and CDMA2000) are implemented. The specifications for these two standards are given in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>CDMA2000</th>
<th>GSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol/chip rate</td>
<td>1.2288Mmps</td>
<td>270.833ksps</td>
</tr>
<tr>
<td>Channel spacing</td>
<td>1.25 MHz</td>
<td>200 kHz</td>
</tr>
<tr>
<td>Passband frequency</td>
<td>0-590kHz</td>
<td>0-80kHz</td>
</tr>
<tr>
<td>Stopband frequency</td>
<td>750kHz</td>
<td>100kHz</td>
</tr>
<tr>
<td>Stopband ripple</td>
<td>-50dB</td>
<td>-18dB</td>
</tr>
<tr>
<td>Stopband frequency</td>
<td>900kHz</td>
<td>300kHz</td>
</tr>
<tr>
<td>Stopband ripple</td>
<td>-87dB</td>
<td>-50dB</td>
</tr>
<tr>
<td>Stopband frequency</td>
<td>--</td>
<td>500kHz</td>
</tr>
<tr>
<td>Stopband ripple</td>
<td>--</td>
<td>-85dB</td>
</tr>
<tr>
<td>Stopband frequency</td>
<td>--</td>
<td>700kHz</td>
</tr>
<tr>
<td>Stopband ripple</td>
<td>--</td>
<td>-95dB</td>
</tr>
</tbody>
</table>

For GSM, the CIC filter uses a decimation by 64 with an input sample rate of 69.333Msps, while for the CDMA filter, we use a decimation factor of 16 with an input sample rate of 78.643Msps. The 20 tap inverse-sinc compensation filter has a decimation factor of 2 and is designed to have a cutoff frequency which eliminates the high frequency image of the CDMA2000 3rd stage (PFIR) filter as shown in figure 3 (note that a normalized frequency of 1 corresponds to a frequency of 4.9152MHz). The design uses a 58 tap PFIR which provides 90db of attenuation at 760kHz. The total response (CIC+CFIR+PFIR) satisfies the requirements for the CDMA2000 spectral mask (see figure 5).

For GSM, we use the same CIC filter (but at a decimation by 64) and the same 20 tap CFIR. A 41 tap PFIR is designed as shown in figure 4 (note that in this case a normalized frequency of 1 corresponds to a frequency of 1.083328Mhz), satisfying the GSM requirements (see figure 6).

5. Results

The infinite precision filter coefficients for the CFIR and the two PFIR filters are converted to CSD format. We use our optimization technique to simultaneously optimize the CFIR and PFIR coefficients of the CDMA2000 implementation. Once a satisfactory solution is found, we freeze the CFIR coefficients and optimize the GSM PFIR coefficients. The technique generates a CFIR filter with a wordlength of 12 bits and a maximum of 4 non-zero bits. For the more stringent CDMA2000 PFIR we require a wordlength of 15 bits with a maximum of 6 non-zero bits. The GSM PFIR has a wordlength of 10 bits with a maximum of 3 non-zero bits. The magnitude characteristics of the infinite precision filter and the optimized finite precision filter for the two standards are shown a figure 5 and figure 6, respectively. The finite precision filter coefficients for the three filters are given in table 2.

The four most common CSEs are used to realize the various filter stages in the implementation. The hardware requirements in terms of the multiplier block adders and the structural adders for the various filters are presented in Table 3. It should be noted that the hardware requirement for the CFIR filter will be double that presented in table 3 if a decimating filter structure is used instead of a conventional FIR filter with a decimator at the output. While a decimating
filter structure reduces the speed requirement of the CFIR filter, this is not an issue with modern hardware such as field programmable gate array (FPGA).

The results from this work show a significant reduction in the hardware requirements compared to other low delay multiplierless implementations [27], [28], as shown in table 4.

6. Conclusions

We present a simple fast technique for designing channel filters suitable for SDR mobile handsets. The proposed design technique compares favorably to existing design techniques presented in the literature. We are currently examining techniques for relaxing the PFIR constraints to enable better component reuse and thus enable more effective reconfiguration options.

Table 2. The infinite precision filter coefficients for the optimized filters.

<table>
<thead>
<tr>
<th>CFIR (combined)</th>
<th>PFIR (CDMA2000)</th>
<th>PFIR (GSM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{-12}$</td>
<td>$2^{-2}$</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>$2^{-12}+2^{-11}$</td>
<td>$2^{-2}$</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>$2^{-7}+2^{-10}$</td>
<td>$2^{-2}$</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>$2^{-7}+2^{-10}+2^{-12}$</td>
<td>$2^{-2}$</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>$2^{-7}+2^{-10}+2^{-12}$</td>
<td>$2^{-2}$</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>$2^{-7}+2^{-10}+2^{-12}$</td>
<td>$2^{-2}$</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>$2^{-7}+2^{-10}+2^{-12}$</td>
<td>$2^{-2}$</td>
<td>$2^{-2}$</td>
</tr>
</tbody>
</table>

Table 3. The hardware complexity of the multi-standard channelizer (single channel).

<table>
<thead>
<tr>
<th>MB Adders</th>
<th>CIC</th>
<th>CFIR</th>
<th>PFIR</th>
<th>Structural Adders</th>
<th>CIC</th>
<th>CFIR</th>
<th>PFIR</th>
<th>Total Adders</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDMA2000</td>
<td>134</td>
<td>12</td>
<td>34</td>
<td>GSM</td>
<td>94</td>
<td>12</td>
<td>11</td>
<td>134</td>
</tr>
</tbody>
</table>

Table 4. A comparison with other implementations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2 dB -80 dB</td>
<td>0.015 dB -80 dB</td>
<td>0.1 dB -97 dB</td>
<td>0.1 dB -87 dB</td>
<td>129</td>
</tr>
<tr>
<td>129</td>
<td>147</td>
<td>134</td>
<td>134</td>
<td>52</td>
</tr>
</tbody>
</table>
10. References


