AN EFFICIENT COEFFICIENT CODING SCHEME FOR LOW COMPLEXITY IMPLEMENTATION OF PULSE SHAPING FILTERS IN GSM RECEIVER OF A SOFTWARE RADIO

A.P.Vinod[1], **E.M-K.Lai**[2], **A.B.Premkumar**[3]

School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798
asvinod@ntu.edu.sg,[1] asmklai@ntu.edu.sg,[2] asannamalai@ntu.edu.sg[3]

Abstract - The IF processing block is by far the most computationally intensive part of wide-band receivers in software radios. From a signal processing point of view, digital filtering is the main task in IF processing. Infinite precision filters require complicated digital circuits due to coefficient multiplication. This paper presents an efficient method to implement pulse shaping filters for GSM receivers. We use an arithmetic scheme, known as pseudo floating-point (PFP) representation to encode the filter coefficients. By employing a span reduction technique, we show that the filters can be coded using 8-bit PFP with hardware complexity that is considerably lower than conventional 24-bit and 16-bit fixed-point filters. Simulation results show that the magnitude response of the filter coded in PFP meets the attenuation requirements of GSM specifications.

Keywords – Software radio, IF processing, coefficient quantization, pseudo floating-point representation, raised cosine filter.

I. INTRODUCTION

Software radio is fast becoming a crucial element of wireless technology. The analog front-end of a software radio receiver contains a wide-band analog-to-digital converter (ADC) for digitizing the entire frequency band allocated to the cellular radio base station. The extraction of individual radio channels from the output of ADC is performed in the IF processing block which contains a bank of digital bandpass filters known as the channelizer. Practical cellular radio base stations require that the wide-band receivers be efficient in their use of resources such as power, hardware cost, and computational resources. A first order estimate of the resources required to implement a wideband receiver shows that the IF processing block is the most computationally intensive part since it operates at the highest sampling rate [1]. Computationally efficient architectures for the implementation of the IF processing block are presented in [2]-[4]. Using multi-rate signal processing concepts, these approaches employ the classical filterbank channelizer for extracting individual channels. Digital filtering is the core function in the IF processing block of a software radio receiver. The linear phase characteristic of FIR filters makes it attractive for software radio channelizers [5]. A low-complexity, high-speed implementation of FIR filters in a cost-effective manner is highly significant due to its ubiquitous use. Hardware implementation of FIR filters generally requires large multipliers if they are realised with continuous-valued coefficients. Approaches to implement filter banks based on the sum-of-powers-of-two (SOPOT) coefficients [6], [7] are attractive because coefficient multiplications can be implemented with simple shifts and additions only. Another approach to reduce the hardware complexity of filters is to encode its coefficients using the canonic signed-digit (CSD) representation [8]. It is known that CSD coding results in nonuniform quantization error due to the uneven distribution of signed powers-of-two terms [9]. In the case of the FIR filter, the most significant error contribution arises from coefficient quantization. The effects of fixed-point arithmetic in the implementation of a digital wideband DCS receiver are discussed in [10]. The authors implement the channel filter with floating-point coefficients and 24 and 16-bit fixed-point coefficients. The magnitude response of the 24-bit filter shows very close resemblance to the floating-point realization. However, the 16-bit implementation indicates a significant degradation in stop-band attenuation preventing the required spectral mask requirements from being met. Simulation results show that the stop-band attenuation of the 16-bit filter is 20dB worse than that of the floating-point implementation.

Efficient implementation methods of fixed-point channel filters meeting the required spectral mask characteristics are hardly discussed in the literature. The general observation in fixed-point implementations [10,11] is that the out-of-band attenuation as well as its droop degrades considerably as fewer bits are used to represent the channel filter coefficients. Moreover, these implementations use relatively long wordlengths such as 24 bits and 16 bits for its coefficients. Hence the hardware complexity of the filter will be high. This paper presents the implementation of pulse shaping filters in software radio receivers using the Pseudo Floating-Point (PFP) representation. An architecture for implementing the quadrature mirror filter (QMF) banks using the PFP representation for the coefficient set has been presented in [12]. The lower bound value of 11 bits is achieved for QMF coefficients realized employing the proposed method. In this paper, we show that the coefficients of the pulse shaping filters can be coded using substantially lower entropy than conventional 24-bit and 16-bit implementations discussed in [10,11]. The magnitude responses of the resulting filters meet the spectral mask characteristic of the relevant standard for mobile communications receivers.

This paper is organized as follows: section II briefly discusses the IF architecture of the GSM receiver. Section III gives a brief review of the PFP representation. We discuss the PFP coding scheme for minimum wordlength implementation of filters in section IV. An algorithm to obtain the PFP coefficients will also be described. In section V, we illustrate the implementation of pulse shaping filters for GSM receivers using several examples. Section VI provides our conclusions.

II. IF ARCHITECTURE FOR GSM RECEIVER

We consider the IF architecture for dual-mode GSM/W-CDMA receiver presented in [4] to show the filter implementation using PFP coefficients. The Global System for Mobile Communication (GSM) is a digital cellular communications system that has already gained acceptance worldwide. The Wideband Code Division Multiple Access (W-CDMA) is a promising candidate for the third generation mobile radio system due to its potential viability for enhancing capacity and high quality data transmission. The architecture of IF processing for GSM is shown in Fig. 1. The IF block performs frequency conversion for W-CDMA and channel extraction from a wide-band received signal for GSM. The input bandwidth of an IF signal covers one channel of 5 MHz in W-CDMA. The purpose of IF processing for GSM is to extract a single channel with a bandwidth of 200 kHz from the 5 MHz received signal. This is achieved by employing a high decimation filter, $H_1(z)$, sample rate converter and finally a pulse shaping filter, $H_2(z)$, to compensate for the frequency response of IF processing. The compensation filter is mainly to attenuate the block signal at 200 kHz as per GSM specifications.

Our focus in this paper is to realize the PFP implementation of the pulse shaping filter, $H_2(z)$, used for IF processing of GSM. No specific attention to the filter implementation issues has been given in the IF architecture designs proposed in literature. While the SOPOT coefficients of the channel filter in [10] offer a magnitude response close to that of continuous-valued coefficients, it suggests the use of 24-bit coefficients. It has been observed that for the shorter wordlength of 16 bits, the stop-band attenuation deteriorates, causing it to fail to meet the filter specifications. Efficient entropy coding schemes of channel filter coefficients are essential for its low-complexity implementation. In this paper, we propose to use a PFP representation for implementing channel filters. The main advantage of using such a representation is, it offers significantly higher performance and consumes less energy. We briefly review the PFP arithmetic scheme that is suitable for programmable Multiply-and-Accumulate (MAC) operations in the following section.

III. THE PSEUDO FLOATING-POINT REPRESENTATION

The general representation of power-of-two terms for the i^{th} filter coefficient, h_i is $h_i = \sum_{j=0}^{B-1} 2^{a_{ij}}$ where B is the number of digits in the power-of-two representation. The expression for h_i can be rewritten as,

$$h_i = 2^{a_{i0}} \cdot \sum_{j=0}^{B-1} 2^{a_{ij}-a_{i0}} = 2^{a_{i0}} \left[\sum_{j=0}^{B-1} 2^{c_{ij}} \right]$$

where $c_{ij} = a_{ij} - a_{i0}$. The term a_{i0} is known as the *shift* and the upper limit value, $\left(a_{i(B-1)} - a_{i0}\right)$, is known as the *span.* The bracketed term is known as the normalised value ($n-$value). The shift and the normalised value are analogous to the exponent and mantissa in true floating-point representations. Instead of expressing the coefficients as a 16-bit integer, it can be expressed as a *(shift, n-value)* pair – this is the definition of the pseudo floating point representation. For a given coefficient set, let L and M be the number of bits needed to encode the shift and *n*-value respectively. Then,

$$L = \max_{0 \le i \le N-1} shift\left(h_i\right)$$
$$M = \max_{0 \le i \le N-1} span\left(h_i\right)$$

This concept can be easily illustrated with an example. Consider the coefficient $h(n)$, whose 16-bit SOPOT representation is given by $h\left(n\right) = 2^{-6} + 2^{-8} + 2^{-9} + 2^{-14}$. This can be written as $2^{-6}\left(2^{0} + 2^{-2} + 2^{-3} + 2^{-8}\right)$. In this expression, the term 2^{-6} is the *shift* part (implying 'right shift by 6'), and the bracketed term is the *span* part. The shifts are hardwired and hence are essentially free. Therefore, only 3 bits are needed for storing the shift value and correspondingly, $L = 3$. The span value, $M = 8$, is obtained from the bracketed term. Hence the coefficient can be represented in PFP using $L + M = 11$ bits. In the case of filter implementation in [10, 11], the L and M values of 24-bit fixed-point coefficients are 5 and 23 respectively. Hence, 28 bits are needed by the PFP for general coefficient sets. For the 16-bit coefficients, L and M are 4 and 15 respectively and thus require a total of 19 bits in PFP representation. It would seem that the PFP representation might not be an optimal representation. However, it would be interesting to investigate if the actual coefficient sets would require less than the 28 bits and 19 bits in these cases. It should be noted that the span contributes significantly more to the wordlength requirement than the shift. The shifts are well distributed across coefficients and so is not a parameter that could be optimized further. Therefore, it is beneficial to explore some efficient means of reducing the span without considerable implication on the magnitude response of the filter. In the following section, we show that by employing a minimum entropy coding scheme, the wordlength requirement of the pulse shaping filter for GSM receiver can be reduced to 8 bits.

IV. AN OPTIMAL WORDLENGTH CODING SCHEME FOR GSM CHANNEL FILTER

A. The Span Reduction Apporach

In our attempt to achieve a minimum wordlength filter of 8 bits for any coefficient set, we fix the *shift* to the maximum value, *l,* corresponding to the worst-case coefficient set. Thus, if we could manage to restrict the *span* value to *m* such that $l + m \leq 8$, the PFP bit requirement would reduce the total number of bits to 8, which is the desired wordlength.

A raised cosine filter is designed meeting the design specifications of the pulse shaping filter employed in the IF architecture. The 16-bit SOPOT coefficients are obtained from continuous coefficients by applying a direct discretization. The shift value, *l,* is fixed from the powers-of-two terms of the worst-case coefficient. The coefficient set is examined to determine the number of coefficients whose span exceeds $(8-l)$. If we could somehow manage the spans of those coefficients that exceed $(8-l)$, we would have achieved our goal of setting 8 bits as the lower bound. This reduction in span could be achieved by modifying the coefficients that exceed the span of $(8-l)$ bits. This is possible by simply throwing away the power-of-two terms that exceed the span of $(8-l)$ as illustrated below:

Consider, for example, a typical coefficient with a span of 10 where

$$h(1) = 2^{-6} + 2^{-7} + 2^{-12} + 2^{-16}$$

Assume that the shift value has been fixed as $l = 3$ bits from the worst-case coefficient. Thus, for achieving the desired wordlength of 8 bits, the maximum permissible span is $8 - l = 5$ bits. Hence, $h(1)$ could be modified as:

$$h(1) = 2^{-6} + 2^{-7}$$

We can expect distortion in the frequency response characteristics when such a span reduction technique is employed to all the "offending coefficients". However, the performance degradation has been observed to be minimal and resulting filters satisfy the attenuation requirements. Our observations employing the span reduction technique can be summarized as follows:

a. The pass-band response of the resulting filter does not change.

b. The effect of span reduction on stop-band attenuation and peak stop-band ripple is minimal in the case of filters having relatively fewer numbers of taps (filter length, $N < 40$).

The reason for this behaviour can be explained as follows.

1. In the case of short length filters, the spans are more closely distributed around the lower bound *m*-bit whereas for long filters spans are sparsely distributed. Hence, the magnitudes of those terms whose span exceeds *m,* which are thrown away are considerably smaller for short length filters when compared to that of filters having large number of taps. As a result, the sensitivity of PFP coefficients to span reduction is very low.

2. The span deviation from the lower bound *m*-bit is relatively uniform across the coefficient grid in the case of filters with fewer taps when compared to that with larger taps. As a result, applying span reduction to the filter is similar to scaling the entire coefficient. Scaling the coefficient set will not affect the frequency response shape; instead it only changes the filter gain. In the case of PFP representation for short length filters, the deviation in gain is minimal since the span deviation from *m* is minimal.

The raised cosine filters employed as pulse shaping filters in IF processing are short length filters whose taps is normally is less than 40. For instance, consider the filter $H_2(z)$ employed in the IF architecture of the GSM receiver shown in Fig. 1. According to the specification of GSM, the attenuation requirement at 200 kHz is –20 dB. As shown in section V, a raised cosine filter with 19 taps meets this specification. Hence the proposed span reduction technique can be employed to encode its coefficients using fewer number of bits.

B. Minimum Wordlength PFP Coding Algorithm

The steps to encode the filter using 8-bit PFP are summarized below:

Step 1: Design the infinite-precision raised cosine filter according to the GSM channel filter specifications.

Step 2: Set all the coefficients to their closest sum of power-of-two (SOPOT) coefficients using 16-bits and represent them as a *(shift, span)* pair in PFP.

Step 3: Fix the *shift* to the maximum value, *l,* corresponding to the worst-case coefficient set.

Step 4: Employ the span reduction technique to all the offending coefficients. This is achieved by throwing away the powers-of-terms, whose span exceeds $(8-l)$ bits.

V. DESIGN EXAMPLE

The pulse shaping filter $H_2(z)$ in Fig. 1 is now implemented using PFP coding. As in the specification of GSM, the roll-off rate of the filter is to be 0.22, and –20dB attenuation is required at 200 kHz [4]. A raised cosine filter of length 19 is employed as the pulse shaping filter. The filter coefficients are shown in Table 1. Note that only half the coefficient set are listed since the filter is a linear phase FIR filter. The infinite precision filter, with coefficients $h(n)$ is generated by the raised cosine FIR filter design program provided by the MATLAB® "firrcos" function. The coefficients of this filter are listed in the second column of Table 1. The third column of Table 1 shows the 16-bit SOPOT coefficients obtained by discretizing the continuous coefficients. The *shift* value is fixed at $l=4$ bits based on the worst-case coefficients, $h(1)$ and $h(3)$. Therefore, for 8-bit PFP coefficients, the maximum permissible *span* value is $m=8-l=4$ bits. Out of 19 coefficients in the original 16-bit SOPOT set, span of eleven coefficients exceed 4 and these are replaced with coefficients with span less than or equal to 4. The fourth column of Table 1 gives the new set of coefficients obtained in the 8-bit PFP format, after employing span reduction. The magnitude response of the filter is shown in Figure 2. The solid line corresponds to the response of the filter whose coefficients are coded in 8-bit PFP. The dotted line represents the response of the conventional 16-bit SOPOT coefficients and the dashed line represents the response of the 8-bit SOPOT coefficients. It may be noted that the magnitude response of the filter coded using 8-bit SOPOT coefficients suffers substantial deterioration in the pass-band as well as in the stop-band. Response of the 8-bit PFP filter shows close resemblance to that of the 16-bit SOPOT filter. The peak pass-band ripple (PPR) value for of the 8-bit PFP filter is 0.09 dB, which is better than 0.14 dB of the 16-bit SOPOT filter. Though the attenuation of the PFP filter at 200 kHz (-37.7 dB) is slightly lower than the attenuation of the 16-bit SOPOT filter (-41.7 dB), it is still higher than the GSM specification of –20 dB. Both filters offer identical peak stop-band ripple (PSR) of –35.80 dB. These comparisons show that there is practically no difference in the response of filters obtained using the proposed representation and the normal methods.

VI. CONCLUSIONS

We have presented an efficient coefficient coding scheme using pseudo floating-point representation for pulse shaping filters in software radio receivers. This representation uses a span reduction technique that results in minimizing the wordlength to 8 bits, which is considerably less than the fixed-point realizations discussed in the literature. Simulation results clearly show that the proposed method results in filters with good frequency response characteristics. It is also worth noting that the proposed PFP filter is hardware efficient due to its minimum wordlength requirement.

REFERENCES

[1] J. Mitola, *Software Radio Architecture.* New York: Wiley, 2000.

[2] K.C. Zangi and R.D. Koilpillai, "Software radio issues in cellular base stations," *IEEE Journal on Selected Areas in Communications,* vol. 17, no.4, pp. 561-573, April 1999.

[3] W.H. Yung, M. Jian, and Y.W. Ho, "Polyphase decomposition channelizers for software radios," in *Proc. of IEEE International Symposium on Circuits and Systems,* vol. 2, pp. 3533-356, May 2000.

[4] M. Jian, W.H. Yung, and B. Songrong, "An efficient IF architecture for dual-mode GSM/W-CDMA receiver of a software radio," in *IEEE International Workshop on Mobile Multimedia Communications,* pp. 21-24, November 1999.

[5] David B. Chester, "Digital IF Filter Technology for 3G Systems: An Introduction," *IEEE Commun. Mag.,* pp. 102-107, February 1999.

[6] Y.C. Lim and S.R. Parker, "FIR filter design over a discrete Powers-of-Two Coefficient space", *IEEE Transactions on Acoustic, Speech and Signal Processing,* vol. ASSP-31, no. 3, pp. 583-591, June 1983.

[7] S.C. Chan, W. Liu, and K.L. Ho, "Multiplierless perfect reconstruction modulated filter banks with sum-of-powers-of-two coefficients," *IEEE Signal Processing Letters,* vol. 8, no. 6, pp. 163-166, June 2001.

[8] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems,* vol. 36, pp. 1044-1047, July 1989.

[9] Y.C. Lim, R. Yang, D. Li, and J. Song, "Signed-power-of-two term allocation scheme for the design of digital filters," *IEEE Trans. Circuits Syst. II,* vol. 46, pp. 577-584, May 1999.

[10] H.R. Karimi, N.W. Anderson, and P. McAndrew, "Digital signal processing aspects of software definable radios," *IEE Colloquium, Adaptable and Multistandard Mobile Radio terminals,* March 1998.

[11] K. Kalbasi, "Performance tradeoffs in a dual-mode, W-CDMA/EDGE digital IF receiver," Tech. Rep., *Agilent EEs of Electronic Design Automation*, 2001.

[12] A.B. Premkumar, C.T. Lau, and A.P. Vinod, "High performance architectures for QMF banks," *NASA* 9^{th} *Symposium on VLSI Architecture,* November 2000.

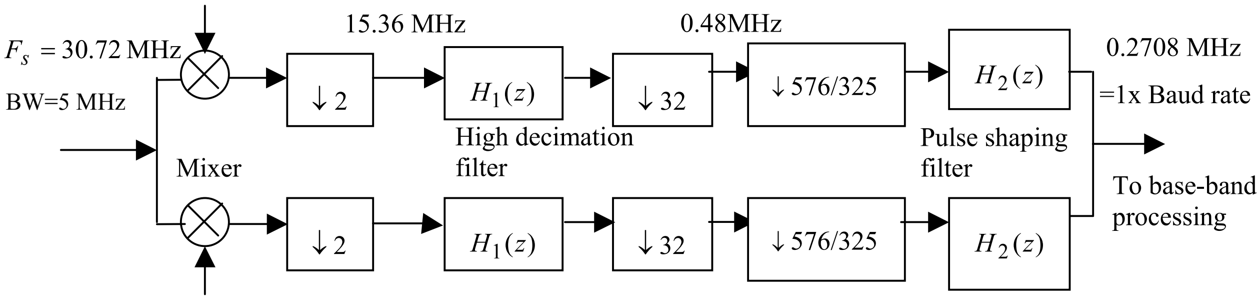

Fig. 1. IF architecture for GSM

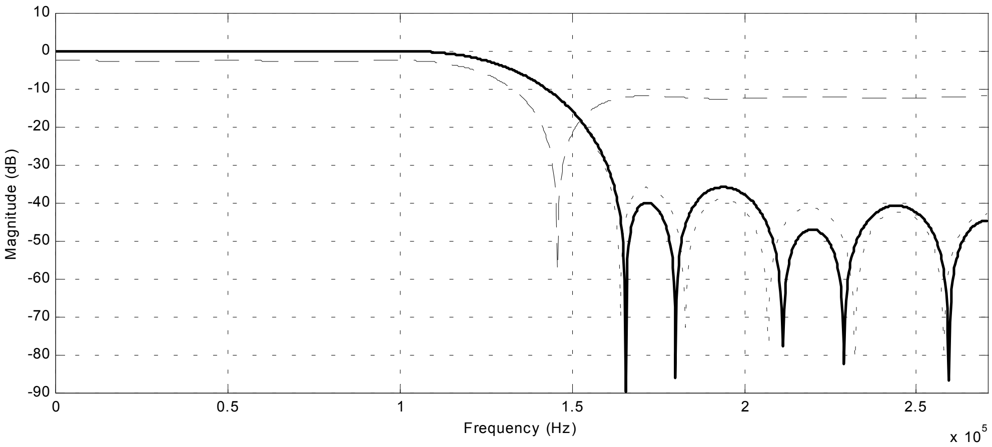

Fig. 2. Magnitude response of pulse shaping filter, $H_2(z)$, for the GSM receiver. Solid: 8 bit PFP coefficients, Dot: 16-bit SOPOT coefficients, Dash: 8-bit SOPOT coefficient (Frequency scale 0 - 270.833 kHz).

$h(n)$	Continuous Coefficients	SOPOT Coefficients (16-bit)	PFP Coefficients (8-bit) Span-4 bits, Shift-4 bits
$h(0)$	0.01209918192395	$2^{-7}+2^{-8}+2^{-12}+2^{-13}$	$2^{-7}+2^{-8}$
$h(1)$	0.00003817608605	2^{-15}	2^{-15}
$h(2)$	-0.02486280958996	$-2^{-6}-2^{-7}-2^{-10}-2^{-12}-2^{-13}-2^{-14}-2^{-16}$	$-2^{-6}-2^{-7}-2^{-10}$
$h(3)$	-0.00005589755747	$-2^{-15}-2^{-16}$	$-2^{-15}-2^{-16}$
$h(4)$	0.04741839914182	$2^{-5}+2^{-6}+2^{-11}+2^{-15}+2^{-16}$	$2^{-5}+2^{-6}$
$h(5)$	0.00007155394225	2^{-14}	2^{-14}
$h(6)$	-0.09569277954707	$-2^{-4}-2^{-5}-2^{-10}-2^{-11}-2^{-12}-2^{-13}-2^{-14}-2^{-15}-2^{-16}$	$-2^{-4}-2^{-5}$
$h(7)$	-0.00008231854614	$-2^{-14}-2^{-16}$	$-2^{-14}-2^{-16}$
$h(8)$	0.31472306476432	$2^{-2}+2^{-4}+2^{-9}+2^{-12}+2^{-16}$	$2^{-2}+2^{-4}$
$h(9)$	0.50008615390770	$2^{-1}+2^{-13}+2^{-16}$	2^{-1}

Table 1. Coefficients of pulse shaping filter for the GSM receiver