LOW POWER AND HIGH-SPEED IMPLEMENTATION OF FIR FILTERS FOR SOFTWARE DEFINED RADIO RECEIVERS

A. P. Vinod and E. M-K. Lai
School of Computer Engineering, Nanyang Technological University
Singapore 639798
Tel: (+65) 67906258 Fax: (+65) 67926559
Email: {asvinod, asmklai}@ntu.edu.sg

Abstract

The most computationally intensive part of the wideband receiver of a software defined radio (SDR) is the intermediate frequency (IF) processing block. Digital filtering is the main task in IF processing. The computational complexity of finite impulse response (FIR) filters used in the IF processing block is dominated by the number of adders (subtractors) employed in the multipliers. This paper presents a method to implement FIR filters for SDR receivers using minimum number of adders. 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 filter coefficients can be coded using considerably fewer bits than conventional 24-bit and 16-bit fixed-point filters. Simulation results show that the magnitude responses of the filters coded in PFP meet the attenuation requirements of wireless communication standard specifications. The proposed method offers average reductions of 40% in the number of adders and 80% in the number of full adders needed for the coefficient multipliers over conventional FIR filter implementation methods.

Index Terms – IF processing, Pseudo floating-point representation, software defined radio, FIR filter.

I. Introduction

SDR is fast becoming a crucial element of wireless technology. The use of SDR technology is predicted to replace many of the traditional methods of implementing transmitters and receivers while offering a wide range of advantages including adaptability, reconfigurability, and multifunctionality encompassing modes of operation, radio frequency bands, air interfaces, and waveforms [1]. Research in this field is mainly directed towards improving the architecture and the computational efficiency of SDR systems.

The most computationally intensive part of an SDR receiver is the channelizer since it operates at the highest sampling rate [2]. Channelization in SDR receivers involves the extraction of multiple narrowband channels from a wideband signal using several bandpass filters called channel filters [3]. Low power and high-speed FIR filters are required in the channelizer [4]. The key functional units in a digital filter are delay, adder, and multiplier – out of which multiplier dominates the hardware complexity. It is well known that by representing filter coefficients as sum-of-powers-of-two (SOPOT), each multiplication in filtering can be replaced with simple shift-and-add operations [5]-[8]. The complexity of the FIR multiplier is dominated by the number of adders (subtractors) employed in the coefficient multipliers. The number of adders needed in the multipliers is proportional to the coefficient wordlength [9]. The fixed-point arithmetic implementation of channel filters in digital wideband receivers requires 24-bit wordlength to meet the channel specifications [10, 11]. It has been reported in [10] that the 16-bit filter implementation results in a significant degradation in stop-band attenuation, failing to meet the spectral mask requirements. The filters implemented using 24-bit and 16-bit SOPOT coefficients require considerably larger number of adders and hence further hardware optimization is required to meet the constraints of power consumption and speed in SDR receivers [11]. In this paper, we present the implementation of FIR filters using an arithmetic scheme called Pseudo Floating-Point (PFP) representation [12]. We show that the coefficients can be coded using considerably fewer bits than the conventional 24-bit and 16-bit implementations. The magnitude responses of the resultant filters meet the spectral mask characteristic of the relevant standard for mobile communications receivers. The contributions of this paper can be summarized as follows:

1. An efficient coefficient coding scheme using PFP representation for implementing FIR filters in SDR receivers is proposed. By employing a span reduction technique, it is shown that the wordlength of the filters can be minimized to 10 bits or fewer.
2. All previous work on filter implementation [5]-[9] discussed hardware reduction in terms of the number of adders and has not addressed the complexity of adders. The complexity of each adder employed in multiplication is significant for low power and high-speed implementations. We analyze the complexity of implementation in terms of full adders required for each multiplier of the filter. A low power, high-speed implementation of PFP coded filters with a minimum number of full adders is proposed.

This paper is organized as follows: In section II, a brief review of the PFP representation is provided. The PFP coding scheme for implementation of FIR filters is discussed in section III. In section IV, a low power implementation of PFP coded filters is presented.

The design examples of FIR filters for the SDR receiver are illustrated in section V. Section VI provides our conclusions.

II. The Pseudo Floating-Point Representation

The general representation of sum-of-powers-of-two (SOPOT) terms [5] for the i^{th} filter coefficient is

$$h_i = \sum_{j=0}^{B-1} 2^{a_{ij}} \ . \tag{1}$$

where B is the number of digits in the power-of-two representation. The expression for h_i can be rewritten as

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

where $c_{ij} = a_{ij} - a_{i0}$. The term a_{i0} is known as the *shift* and the upper limit value, $(a_{i(B-1)} - a_{i0})$, is known as the *span* [12]. 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* [12]. For a given coefficient set of an N-tap filter, let L and M be the number of bits needed to encode the shift and n-value respectively. Then,

$$L = \max_{0 \le i \le N-1} shift\,(h_i) \tag{3}$$

$$M = \max_{0 \le i \le N-1} span\,(h_i)\,. \tag{4}$$

The following example illustrates this concept. Consider the coefficient $h(n)$, whose 16-

bit SOPOT representation is given by $h(n) = 2^{-6} + 2^{-8} + 2^{-9} + 2^{-14}$. This can be written as $2^{-6}(2^{0} + 2^{-2} + 2^{-3} + 2^{-8})$. 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 less complex since they can be hardwired. Therefore, only 3 bits are needed for storing the shift value (SOPOT representation of 6 is 110) 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, whereas its SOPOT representation requires 16 bits. In the case of the practical 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 based on the examples in [10, 11]. 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. The span contributes significantly more to the wordlength requirement than the shift. The shift values depend on the coefficient wordlength and its maximum value is fixed based on the worst-case coefficient (coefficient that has the largest power of two term) 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.

III. PFP Coding Scheme for FIR Filters

In this section, we show that by employing a span reduction technique, the wordlength requirement of the FIR filters for an SDR receiver can be significantly reduced.

A. Span Reduction Technique

In our attempt to achieve a minimum wordlength for any coefficient set, we fix the *shift* to the maximum value, *l,* corresponding to the worst-case coefficient set using equation (3). The span value is progressively reduced by discarding the power-of-two terms and checking whether the resulting filter response meets the filter specifications at each stage. We can expect distortion in the frequency response characteristics, when such a span reduction technique is employed to all the offending coefficients (offending power-of-two terms here being defined as the power-of-two terms that exceed the lower bound span). Our observation in employing the span reduction technique is that the pass-band response of the resulting filter does not change drastically. It has also been noted that the effect of span reduction on stop-band attenuation and peak stop-band ripple is minimal in the case of filters having relatively few taps (filter length, $N < 40$). The reason for this behaviour can be explained as follows. Let the span value after performing the reduction be s_m.

1. In the case of lower order filters, the spans are closely distributed around s_m, whereas for higher order filters the spans are widely distributed. Hence, the magnitudes of those terms whose span exceed s_m, which are discarded are considerably smaller for lower order filters when compared to that of higher order filters. As a result, the sensitivity of the PFP coefficients to span reduction is very low. Sensitivity is a measure of the degree of influence on the frequency response of a filter when any one of the coefficients is quantized. The sensitivity $s(n)$ can be computed by setting each coefficient, in turn, to its nearest power of two, yielding in each case a response $H_q(\omega_i)$, which gives:

$$s(n) = \frac{1}{M} \sum_{i=1}^{M} [H_q(\omega_i) - H(\omega_i)]^2 \ . \tag{5}$$

where $H(\omega_i)$ and $H_q(\omega_i)$ are the frequency responses of the infinite-precision and quantized coefficients respectively at M finite number of frequencies ω_i [13]. The equivalent time-domain expression for sensitivity of an N-tap filter is given by

$$s(n) = \frac{1}{M} \sum_{n=0}^{N-1} [h_q(n) - h(n)]^2 \ . \tag{6}$$

where $h_q(n)$ and $h(n)$ represent the impulse responses of the quantized and infinite-precision coefficients respectively. In the case of lower order filters, $[h_q(n) - h(n)]$ is small due to the distribution of spans close to s_m. Hence, the sensitivity, which is a square function is minimal and therefore the frequency response of the filters are almost unaltered by the proposed span reduction technique. This will be illustrated in the design examples provided in section V.

2. The span deviation from s_m is relatively uniform across the different coefficients in the case of filters with fewer taps when compared to that with larger taps. We have investigated several examples of raised cosine filters with up to 40 taps corresponding to different stop-band attenuation specifications. The floating-point filter coefficients were generated using the “firrcos” function in MATLAB. Filter coefficients represented in 16-bit SOPOT and 8-bit PFP forms were examined. Fig. 1 shows the average span distribution across the coefficients of fifteen filters that we designed. These filters have identical number of taps (i.e., 40) but different cutoff frequencies. Note that one half of the symmetric filter coefficients are shown. The lower bound span value obtained using

the span reduction algorithm indicated by the horizontal dotted line in Fig. 1 is 5 bits. It can be noted that the deviation of the spans of the coefficients $h(0)$ to $h(18)$ from the lower bound value is uniform (within the rage of 5 to 7 bits) across the coefficient grid. As a result, applying span reduction is similar to scaling the coefficients by the power-of-two terms exceeding the lower bound span value. 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 filters, the change in gain is minimal since the span deviation from s_m is minimal. The worst-case span deviation occurs for the larger valued coefficient $h(19)$, whose span is 9 bits. However, as it will be illustrated in the design examples in section V, the magnitude of the offending power-of-two terms of the larger valued coefficient is extremely small (2^{-14} to 2^{-16}). Hence the response deterioration meets the stop-band attenuation specification when the PFP span reduction technique is employed. It is worth to note that the tolerance scheme of Nyquist filters does not have a constant maximum filter transfer function deviation with respect to the perfect Nyquist characteristic. Instead, the tolerance scheme becomes wider towards the pass-band edge [14]. Therefore, a minimal deviation of frequency response characteristics from the ideal characteristic is acceptable in Nyquist filters.

B. PFP Coefficient Coding Algorithm

The steps for PFP coding of filters using the span-reduction approach are given below.

Step 1: Design the FIR filter, $h(n)$, as in the specification. Determine the frequency response of the floating-point coefficients, $H_d(\omega) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n}$.

Step 2: Set the coefficients to their closest SOPOT form using specified wordlength and represent them as a (*shift, span)* pair in PFP. Fix the shift to the maximum value *l,* corresponding to the worst-case coefficient set. Find the maximum span value, *M*. Set iteration index to $k = 0$.

Step 3: Decrease the span to *M-1* by discarding the power-of-two terms of offending coefficients and obtain the new set of coefficients, $h_q(n)$. Determine $h_q(n) - h(n)$.

Step 4: The frequency response of the quantized filter whose span is reduced to *M-1* can be obtained using:

$$H_q(\omega_i) = H_d(\omega_i) + H_e(\omega)\,. \tag{7}$$

In the time-domain, (7) can be expressed as

$$\sum_{n=0}^{N-1}[h(n) + \sum_{n=0}^{N-1} h_q(n) - h(n)]\, e^{-j\omega_i n}\,. \tag{8}$$

where ω_i represents frequency samples in the stop-band. Obtain the frequency response using equation (8).

Step 5: If $\left|H_{qs}(\omega_i)\right| \le \left|H_s(\omega_i)\right|$, where $H_{qs}(\omega_i)$ represents the stop-band response of the PFP filter and $H_s(\omega_i)$ as in the stop-band specification of the filter, set $k = k+1$ and go to step 3. Otherwise, terminate the program and choose the PFP coefficients, $h_q(n)$, corresponding to the *'k'*th iteration.

IV. Low Power Implementation

In this section, we present a method to implement the PFP coded filters with low power and high-speed. Note that a reduced numbers of bits are required to encode the coefficients employing the span reduction method and hence the filter can be implemented using a minimum number of adders. Although minimizing the number of adders reduces the complexity, it is also necessary to address the complexity of each adder, which is significant for high-speed, low power implementations. An adder that adds two n-bit numbers requires n full adders (FAs) to compute the sum. The area, power, and speed of an adder depend on the *adder width, n.* Therefore, efforts to optimize these parameters should focus on minimizing the adder width. We shall now obtain the expressions for analyzing the complexity of each adder in the filter structure.

A. Adder Complexity

Definition 1 (Operands): The input signal shifted corresponding to the positional weights of the nonzero digits in the coefficient form the operands of the adders. For example, in the case of the coefficient, $h(n) = 2^{-6} + 2^{-8} + 2^{-9} + 2^{-14}$, if x represents the input signal, the operands are $x >> 6$, $x >> 8$, $x >> 9$, and $x >> 14$, where '>>' represents shift right operation. The number of operands is equal to the number of nonzero digits in the coefficient.

Definition 2 (Range): The range is defined as the number of bits of an operand. If x is an 8-bit signal, the ranges of the above-mentioned operands are 14, 16, 17, and 22 respectively. For an adder whose operands have ranges r_1 and r_2 such that $r_2 > r_1$, the adder width is r_2. Thus the adder would require r_2 FAs to compute the sum.

Definition 3 (adder-step): One adder-step represents an adder in a maximal path of decomposed multiplications. A multiplication can have different adder-steps, depending on its structure. We employ the high-speed tree structure shown in Fig. 2, which uses the minimum number of adder-steps. Coefficients with wordlengths up to 16-bits are considered for analyzing the adder complexity and hence at the most sixteen nonzero operands could occur in a multiplication.

Case I: Odd number of operands

Consider the filter tap shown in Fig. 3 that has an odd number of operands (nine), whose ranges are indicated as r_i. The $r_i s$ shown adjacent to the adders represent the adder widths. The total number of FAs required to implement this filter tap is

$$N_{FA} = r_2 + 2r_4 + r_6 + 3r_8 + r_9 . \tag{9}$$

By extending this minimum adder-step structure to 16-bit coefficients, it can be shown that the number of FAs, (N_o), required to compute the output of a filter tap with *m* (for *m* odd) operands can be determined using the expression:

$$N_o = r_2 + a_1 r_3 + 2r_4 + a_3 r_5 + r_6 + a_5 2r_7 + 3r_8 + a_7 r_9 + r_{10} + a_9 2r_{11} + 2r_{12} + a_{11} 2r_{13} +$$

$$a_9 2r_{11} + 2r_{12} + a_{11} 2r_{13} + r_{14} + a_{13} 3r_{15} . \tag{10}$$

where r_i is the range of the *i*th operand and the a_i's are equal to zero except for a_{m-2}, which is 1. (Note that since we are considering coefficient wordlengths up to 16 bits, the maximum possible value of *m* is 15). This can be illustrated using the example of the filter tap $h_o(n)$, shown in Fig. 4. The coefficient $2^{-6} + 2^{-8} + 2^{-9} + 2^{-14} + 2^{-15}$, is considered where m is odd (five). The numerals adjacent to the data path represent the number of bit-wise right shifts. If x is an 8-bit signal, the ranges, r_1, r_2, r_3, r_4, and r_5, of the operands are 14, 16, 17, 22, and 23 respectively. Note that the adders A_1, A_2, A_3, and A_4 shown in Fig. 4 employed in multiplication have widths 16, 22, 22, and 23 respectively as indicated by the numerals in respective brackets. Using equation (10), the number of FAs for computing $y(n)$ is given by $r_2 + 2r_4 + r_5$, which is 83.

Case II: Even number of operands

Using the approach discussed above, the number of FAs, (N_e), required to compute the output corresponding to a coefficient with *m* operands $(m \le 16)$ is given by:

$$N_e = r_2 + 2r_4 + c_0 r_6 + 3r_8 + c_1 r_{10} + c_2 r_{12} + c_3 r_{14} + 4r_{16}\,. \tag{11}$$

where $c_0 \equiv \begin{cases} 2, \text{for } m = 6 \\ 1, \text{m} \ne 6 \end{cases}$, $c_1 \equiv \begin{cases} 2, \text{for } m = 10 \\ 1, \text{m} \ne 10 \end{cases}$,

$c_2 \equiv \begin{cases} 3, \text{for } m = 12 \\ 2, \text{m} \ne 10 \end{cases}$ and $c_3 \equiv \begin{cases} 3, \text{for } m = 14 \\ 1, \text{m} \ne 10 \end{cases}$.

For example, if six operands are present (i.e., $m = 6$), it would require $(r_2 + 2r_4 + 2r_6)$ FAs.

B. PFP Filter Implementation

In this section, we discuss the filter implementation using the proposed PFP coding and compare it with the conventional SOPOT implementation. Consider the coefficient $h(n)$, represented in SOPOT form, $2^{-6} + 2^{-8} + 2^{-9} + 2^{-14}$. The output of the filter obtained when $h(n)$ is multiplied by x (8-bit signal) can be represented as

$$x >> 6 + x >> 8 + x >> 9 + x >> 14 . \tag{12}$$

In this case, there is an even number of operands (*m=4*) where r_2 and r_4 are 16 and 22 respectively. Using equation (11), the number of FAs for computing $y(n)$ in SOPOT implementation is 60. We shall now show that the number of FAs can be considerably reduced using the PFP coefficients. The PFP representation of $h(n)$ is $2^{-6}(2^0 + 2^{-2} + 2^{-3} + 2^{-8})$. Fig. 5 shows the PFP implementation of this filter tap. The ranges of the operands inside the bracket of $h(n)$ (corresponding to its span bits) are 8, 10, 11 and 16. Thus, when compared with the conventional implementation, the adders A_1, A_2, and A_3, have smaller widths, since the ranges of their operands are smaller. Using equation (11), the number of FAs for computing $x(2^0 + 2^{-2} + 2^{-3} + 2^{-8})$ is 42. The 'shift right' operation corresponding to the span (2^{-6}) of $h(n)$ is performed after the addition stages, as shown alongside the data paths at the output of adder A_3. Note that the shifts are hardwired and hence they do not have any cost implication other than a minimal increase in chip area. The proposed PFP implementation requires only 42 FAs, which is a reduction of 30% over the SOPOT implementation. The number of adder-steps required to compute the partial products is two, which is the same as that of the conventional method. The use of smaller adder widths in the PFP method reduces the delay of each addition and hence the proposed method offers speed improvement over the conventional method. By employing our span reduction algorithm, it is possible to achieve substantially higher reductions of FAs.

V. Design Examples

Example 1: In this example, we use the SOPOT coefficients of the FIR subfilter in the variable digital filter (VDF) based SRC (sampling rate converter) employed in the SDR receiver presented in [15], to illustrate our method. The filter length chosen is 40 as in [15]. The 20-bit SOPOT coefficients and the 10-bit PFP coefficients obtained after span reduction are listed in Table I. Note that one half of the symmetric coefficient set is considered. The sensitivity values of PFP coefficients obtained using equation (6) are also shown in Table I. The shift value is fixed at $L = 5$ bits based on the worst-case coefficients, $h(1)$ and $h(3)$ (5 bits are required to store the maximum shift value, 16). The lower bound of span (M) obtained by employing the proposed algorithm is 5 bits. Hence the coefficient set is represented using 10 bits in PFP. The magnitude responses of the filters are shown in Fig. 6. The red plot corresponds to the response of the filter whose coefficients are coded in 10-bit PFP. The blue plot represents the response of the conventional 20-bit SOPOT. Response of the 10-bit PFP filter shows close resemblance to that of the 20-bit SOPOT filter. The peak stop-band ripple (PSR) of the PFP filter is –24 dB and that of the SOPOT filter is –24.1 dB. Both filters offer identical peak pass-band ripple (PPR) of 0.1 dB. These comparisons show that there is practically no difference in the response of filters obtained using the proposed representation and the conventional methods. The considerably low sensitivity values of PFP coefficients shown in Table I account for this achievement. Fig. 7 shows the time domain characteristics of the FIR filter implemented using the 20-bit SOPOT and 10-bit PFP versions of the coefficients. The impulse response of the filter (symmetric half set coefficients) realized using SOPOT coefficients (marked with blue colored star symbol) exactly coincides with that of the PFP (marked with red colored square). It is observed from Fig. 7 that the zero crossings of the impulse response remain unaltered when the PFP representation after span reduction is used. Therefore, the FIR filters implemented using our method can perform pulse shaping with minimal inter symbol interference (ISI).

Example 2: The W-CDMA receiver architecture presented in [16] is used to illustrate our design in this example. The IF block of the receiver is shown in Fig. 8. The input bandwidth of the IF signal covers one channel of 5 MHz in W-CDMA. The filter $H_1(z)$, performs pulse shaping to achieve an attenuation of –40 dB at 5 MHz as in the W-CDMA specification [17]. The output signal at 15.36 MHz, which is four times the W-CDMA chip-rate of 3.84 Mc/s, is fed to base-band processing. The roll-off factor is selected as 0.22 for bandwidth efficiency in 3G cellular applications. A raised cosine filter of length 33 is designed. The lower bound PFP obtained is 8 bits. The PFP coefficients obtained after span reduction and their sensitivity values with respect to 16-bit SOPOT coefficients are listed in Table II. The magnitude responses of the filters are shown in Fig. 9. Both the filters meet the desired attenuation of –40 dB at 5 MHz as in W-CDMA specifications. The 8-bit PFP filter response shows close resemblance to that of the 16-bit SOPOT filter. Table III shows the comparison of the numbers of adders required to implement the filters in the design examples. The PFP implementation offers an average reduction of 44% over the conventional implementation in the design examples. Comparison of the numbers FAs required to implement the multipliers for the filters using the SOPOT method and the proposed PFP method are shown in Table IV. The percentage reductions of FAs in the PFP method over the SOPOT method are also shown in Table IV. The number of FAs shown in Table IV is computed as follows. Consider $h(4)$ in Table I. The 16-bit SOPOT form of $h(4)$ is $2^{-10}-2^{-13}+2^{-17}+2^{-19}$. In this case, m is 4 and considering an 8-bit signal, the values of r_2 and r_4 are 21 and 27 respectively. Using equation (11), the number of FAs required to obtain the output of this filter tap is 75. The PFP coding of $h(4)$ is $2^{-10}(2^0-2^{-3}+2^{-7}+2^{-9})$. The values of r_2 and r_4 are 11 and 17 respectively and only 45 FAs are required to obtain the output. (Note that the shift operation 2^{-10}, can be hardwired). Thus, PFP coding before span reduction (BSR) results in 40% reduction over 20-bit SOPOT coding. The 10-bit PFP form of $h(4)$ obtained after span reduction (ASR) is $2^{-10}(2^0-2^{-3})$. In this case, m is 2 and r_2 is 11. Therefore, only 11 FAs are required, which is a reduction of 85% over SOPOT coding. It must be noted that this substantial reduction is achieved without any performance deterioration on the frequency response of the filter.

VI. Conclusions

We have presented an efficient coefficient coding scheme using pseudo floating-point representation and a span reduction algorithm for implementation of FIR filters in SDR receivers. The computational complexity of the algorithm is relatively less since it is applied only to the filter stop-band response samples. Our method can be used to implement any FIR filters provided the number of taps is less than 40. A low power and high-speed implementation using a minimum number of full adders is also proposed. Our method offers average reductions of 40% in the number of adders and 80% in the number of full adders over conventional FIR filter implementation methods.

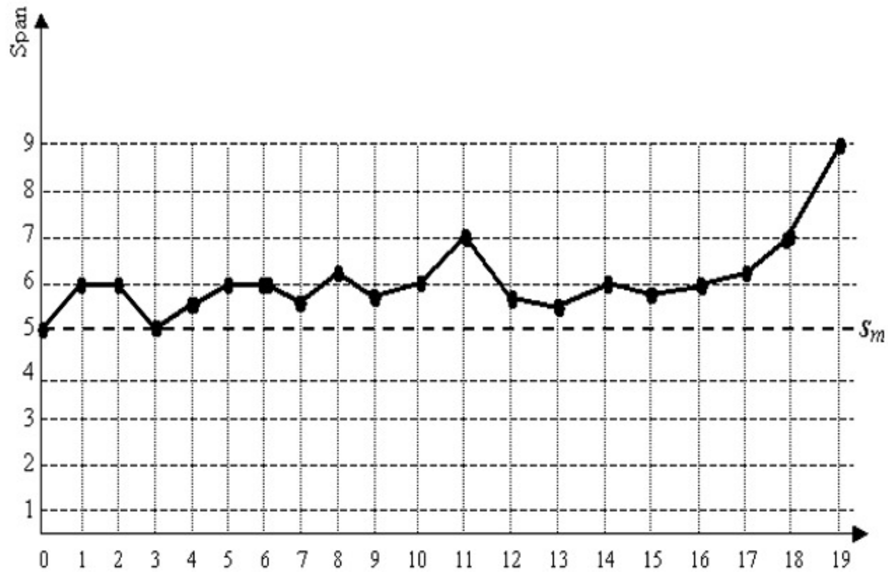

Fig. 1. Average distribution of spans across the coefficient sets of 40-tap raised cosine filters.

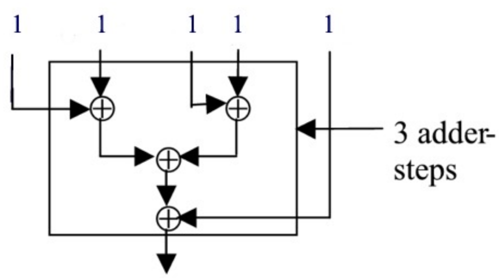

Fig. 2. Tree structure employed for multiplication.

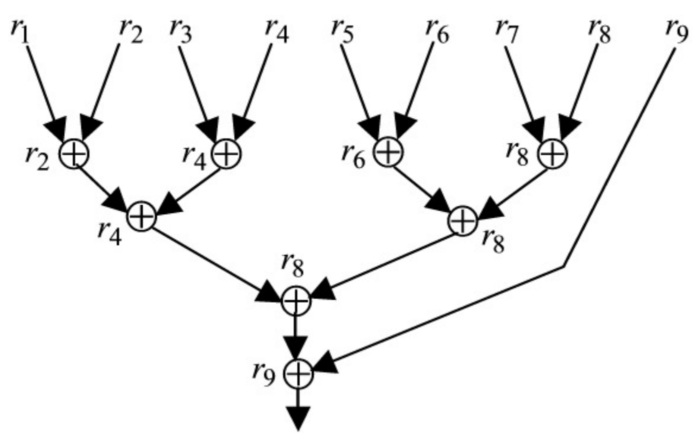

Fig. 3. Implementation of filter tap for odd number of operands.

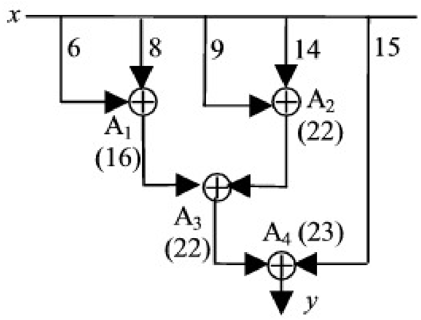

Fig. 4. Implementation of the filter tap, $h_0(n)$.

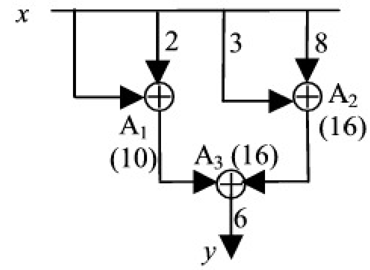

Fig. 5. PFP Implementation of the filter tap.

TABLE I
Coefficients of the FIR filter in Example 1.

$h(n)$	20-bit SOPOT Coefficients (shift: 5-bit, span: 5-bit)	10-bit PFP Coefficients	Sensitivity ($M = 128$)
$h(0)$	$2^{-15} + 2^{-17}$	$2^{-15}(2^0 + 2^{-2})$	0
$h(1)$	$2^{-16} - 2^{-20}$	$2^{-16}(2^0 - 2^{-4})$	0
$h(2)$	$2^{-12} + 2^{-18}$	2^{-12}	1.1×10^{-13}
$h(3)$	$2^{-16} + 2^{-17} + 2^{-19}$	$2^{-16}(2^0 + 2^{-1} + 2^{-3})$	0
$h(4)$	$2^{-10} - 2^{-13} + 2^{-17} + 2^{-19}$	$2^{-10}(2^0 - 2^{-3})$	7.1×10^{-13}
$h(5)$	$-2^{-11} + 2^{-15} + 2^{-16} + 2^{-18}$	$2^{-11}(-2^0 + 2^{-4} + 2^{-5})$	1.1×10^{-13}
$h(6)$	$-2^{-9} - 2^{-12} - 2^{-17} - 2^{-18}$	$2^{-9}(-2^0 - 2^{-3})$	1×10^{-12}
$h(7)$	$2^{-9} + 2^{-14} + 2^{-20}$	$2^{-9}(2^0 + 2^{-5})$	7.1×10^{-15}
$h(8)$	$2^{-8} + 2^{-11} - 2^{-14} - 2^{-16}$	$2^{-8}(2^0 + 2^{-3})$	4.6×10^{-11}
$h(9)$	$-2^{-8} - 2^{-9} - 2^{-12} - 2^{-15} - 2^{-19}$	$2^{-8}(-2^0 - 2^{-1} - 2^{-4})$	8.2×10^{-12}
$h(10)$	$-2^{-8} - 2^{-9} - 2^{-11} - 2^{-13} - 2^{-16} - 2^{-19}$	$2^{-8}(-2^0 - 2^{-1} - 2^{-3} - 2^{-5})$	2.3×10^{-12}
$h(11)$	$-2^{-6} - 2^{-10} + 2^{-16}$	$2^{-6}(-2^0 - 2^{-4})$	1.8×10^{-12}
$h(12)$	$2^{-7} - 2^{-10} - 2^{-13} - 2^{-14} + 2^{-18} + 2^{-20}$	$2^{-7}(2^0 - 2^{-3})$	2.5×10^{-10}
$h(13)$	$-2^{-5} + 2^{-10} + 2^{-12} + 2^{-13} - 2^{-16} - 2^{-18} - 2^{-20}$	$2^{-5}(-2^0 + 2^{-5})$	9.4×10^{-10}
$h(14)$	$-2^{-11} - 2^{-13} + 2^{-17} + 2^{-18}$	$2^{-11}(-2^0 - 2^{-2})$	1×10^{-12}
$h(15)$	$2^{-4} - 2^{-7} + 2^{-10} + 2^{-13} + 2^{-14} + 2^{-18} + 2^{-20}$	$2^{-4}(2^0 - 2^{-3})$	1.1×10^{-8}
$h(16)$	$-2^{-6} - 2^{-8} - 2^{-9} - 2^{-13}$	$2^{-6}(-2^0 - 2^{-2} - 2^{-3})$	1.2×10^{-10}
$h(17)$	$-2^{-3} + 2^{-6} + 2^{-10} + 2^{-13} + 2^{-15}$	$2^{-3}(-2^0 + 2^{-3})$	9.9×10^{-9}
$h(18)$	$2^{-3} - 2^{-6} + 2^{-9} - 2^{-13} - 2^{-16} - 2^{-19} - 2^{-20}$	$2^{-3}(2^0 - 2^{-3})$	2.6×10^{-8}
$h(19)$	$2^{-1} - 2^{-6} - 2^{-8} - 2^{-13} - 2^{-15} - 2^{-18} - 2^{-19}$	$2^{-1}(2^0 - 2^{-5})$	1.3×10^{-7}

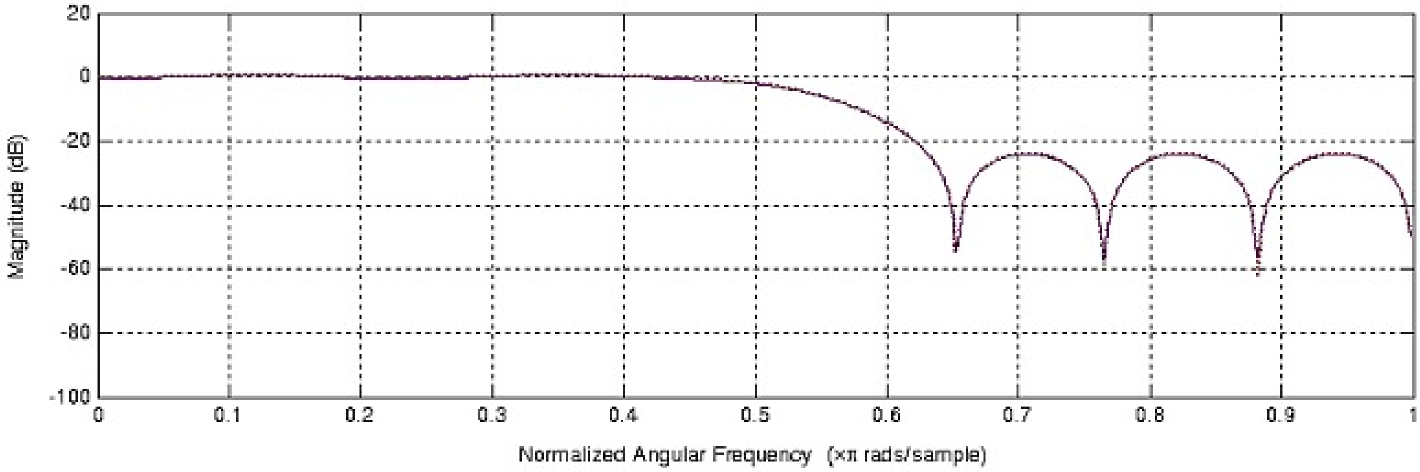

Fig. 6. Filter frequency responses in Example 1.
Solid line: 10-bit PFP, Dotted line: 20-bit SOPOT (both responses coincide).

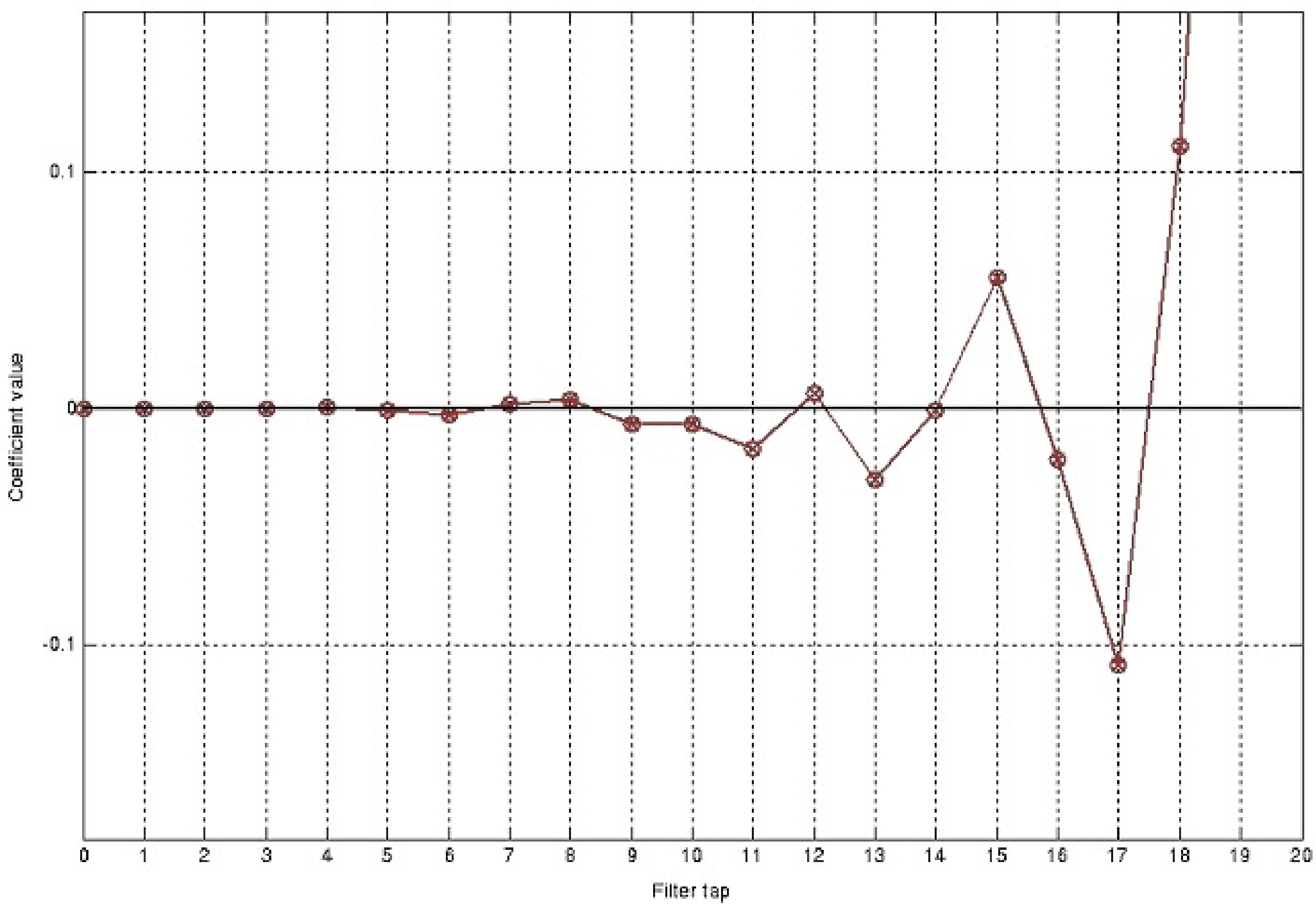

Fig. 7. Impulse response of the FIR subfilter in Example 1 using 20-bit SOPOT coefficients (marked with cross) and 10-bit PFP coefficients (marked with circle) obtained by the proposed algorithm (both characteristics coincide).

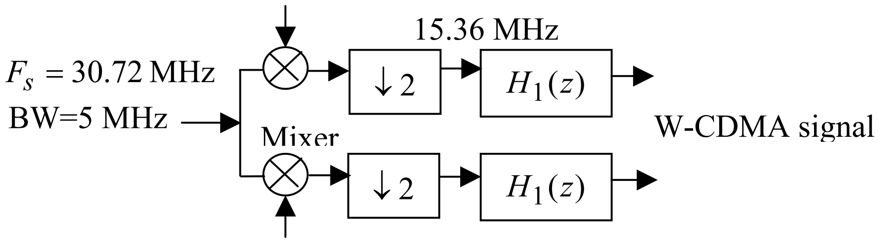

Fig. 8. IF block of the W-CDMA receiver.

TABLE II
Coefficients of the filter in Example 2.

$h(n)$	8-bit PFP Coefficients (shift: 3-bit, span: 5-bit)	Sensitivity ($M = 128$)
$h(0)$	0	0
$h(1)$	2^{-7}	9.6×10^{-8}
$h(2)$	$-2^{-7}-2^{-8}-2^{-11}$	3.6×10^{-10}
$h(3)$	$-2^{-7}-2^{-9}-2^{-11}$	6.5×10^{-11}
$h(4)$	0	0
$h(5)$	$2^{-7}+2^{-8}+2^{-9}+2^{-11}$	8.9×10^{-11}
$h(6)$	$2^{-6}+2^{-7}$	5.3×10^{-11}
$h(7)$	$2^{-6}+2^{-8}$	3.1×10^{-10}
$h(8)$	0	0
$h(9)$	$-2^{-6}-2^{-7}-2^{-8}-2^{-11}$	4.5×10^{-11}
$h(10)$	$-2^{-5}-2^{-6}$	7.2×10^{-9}
$h(11)$	$-2^{-5}-2^{-7}-2^{-9}$	6.1×10^{-9}
$h(12)$	0	0
$h(13)$	$2^{-4}+2^{-7}+2^{-9}$	5.7×10^{-9}
$h(14)$	$2^{-3}+2^{-5}$	9.4×10^{-9}
$h(15)$	$2^{-3}+2^{-4}+2^{-5}+2^{-8}$	2.5×10^{-8}
$h(16)$	2^{-2}	0

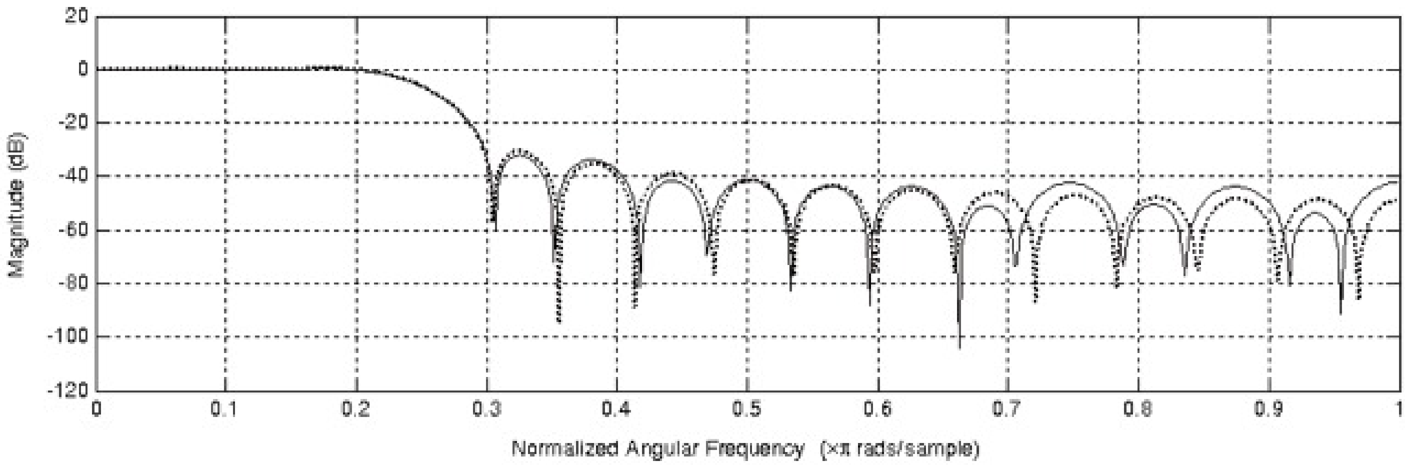

Fig. 9. Filter frequency responses in Example 2.
Solid line: 8-bit PFP, Dotted line: 16-bit SOPOT
(Frequency scale indicated is 0 – 7.68 MHz).

TABLE III
Number of adders required to implement the filters in design examples.

Implementation method	Example 1	Example 2
Conventional SOPOT	107	85
Proposed PFP	63	45
Percentage Reduction	41%	47%

TABLE IV
Number of full adders required to implement the filters in design examples.

Example 1			Example 2		
SOPOT (16-bit)	PFP (BSR)	PFP (ASR)	SOPOT (16-bit)	PFP (BSR)	PFP (ASR)
1599	1132	282	1233	896	230
Reduction	29%	82%		27.3%	81%

REFERENCES

[1] W. Tuttlebee and H. W. Wally, *Software Defined Radio: Enabling Technologies.* New York: Wiley, 2002.

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

[3] T. Hentschel, "Channelization for software defined base-stations," *Annales des Telecommunications,* ISSN 0003-4347, vol. 57, pp. 386-420, no. 5-6, May/June 2002.

[4] D. B. Chester, "Digital IF filter technology for 3G systems: An introduction," *IEEE Commun. Mag.,* vol. 37, no. 2, pp. 102-107, February 1999.

[5] 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.

[6] 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.

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

[8] 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.

[9] Y. C. Lim, Y. Sun, and Y. J. Yu, "Design of discrete-coefficient FIR filters on loosely connected parallel machines," *IEEE Transactions on Signal processing,* vol. 50, no. 6, pp. 1409-1416, June 2002.

[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 (Reg. No. 1998/406),* pp. 3/1-3/8, London, 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*, European Microwave Week, September 2001.

[12] A. B. Premkumar, C. T. Lau, and A. P. Vinod, "High performance architectures for QMF banks," *9th NASA Symposium on VLSI Design,* Albuquerque, New Mexico, November 2000.

[13] X. Hu, L. S. DeBrunner, and V. DeBrunner, "An efficient design for FIR filters with variable precision," *Proc. 2002 IEEE Int. Symp. on Circuits and Systems,* pp. 365-368, vol. 4, May 2002.

[14] T. Henk, "Modelling considerations and mean square error analysis for PSK data transmission," *International Journal of Circuit Theory and Applications*, vol. 8, no. 4, pp. 343-353, October 1980.

[15] K. S. Yeung and S. C. Chan, "The design and multiplier-less realization of software radio receivers with reduced system delay," *IEEE Transactions on Circuits and Systems-I: Regular papers,* vol. 51, no. 12, pp. 2444-2459, December 2004.

[16] 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, San Diego, CA, November 1999.

[17] N. Spencer, "An overview of digital telephony standards," *IEE Colloquium on the Design of Digital Cellular Handsets (Ref. No. 1998/240),* London, pp. 1/1-1/7, March 1998.

Low-power and High-speed Implementation of Pulse Shaping Filters in Software Defined Radio Receivers

Vinod, A. P.

2006-07-01