

Realization of Low Power High-Speed Channel Filters with Stringent Adjacent Channel Attenuation Specifications for Wireless Communication Receivers

Jimson Mathew^{*}, R. Mahesh⁺, A. P. Vinod⁺ and Edmund M-K. Lai[#]

Department of Computer Science, University of Bristol, United Kingdom^{*}

School of Computer Engineering, Nanyang Technological University, Singapore⁺

Institute of Information Sciences & Technology, Massey University, Wellington, New Zealand[#]

Abstract

Finite impulse response (FIR) filtering is the most computationally intensive operation in the channelizer of a wireless communication receiver. Higher order FIR channel filters are needed in the channelizer to meet the stringent adjacent channel attenuation specifications of wireless communications standards. The computational cost of FIR filters is dominated by the complexity of the coefficient multipliers. Even though many methods for reducing the complexity of filter multipliers have been proposed in literature, these works focused on lower order filters. This paper presents a coefficient-partitioning-based binary subexpression elimination method for realizing low power FIR filters. We show that the FIR filters implemented using proposed method consume less power and achieve speed improvement compared to existing filter implementations. Design examples of the channel filters employed in the Digital Advanced Mobile Phone System (D-AMPS) and Personal Digital Cellular (PDC) receivers show that the proposed method achieved 23% average reductions of full adder and power consumption and 23.3% reduction of delay over the best existing method. Synthesis results show that the proposed method offers average area reduction of 8% and power reduction of 22% over the best known method in literature.

Keywords—Wireless communication receiver, channelizer, FIR filter, binary subexpression elimination, low power

I. INTRODUCTION

The most computationally intensive part of the digital front-end of a wireless communication receiver is the channelizer since it operates at the highest sampling rate [1]. Channelization in receivers involves the extraction of multiple narrowband channels from a wideband signal using several bandpass filters called channel filters. The channel filter must have very narrow transition band and considerable stopband attenuation to meet the stringent adjacent channel attenuation requirements of wireless communications standards. Therefore, FIR filters with large number of taps (typically 200 to 1200 taps) are employed in the channelizer. Furthermore, the channel filters must have low power consumption and high-speed.

The complexity of FIR filters is dominated by coefficient multiplication operation. The number of partial product adders needed for coefficient multiplications and the critical path length (number of adder-steps in a maximal path of decomposed multiplications) of the multiplier are the two metrics commonly used in literature to analyze the complexity of FIR filters. Hence, the methods that minimize the complexity of multiplication in FIR filters focus on reducing the number of adders and critical path length (CPL) used to implement the multipliers. Among the low complexity FIR multiplier methods, the common subexpression elimination (CSE) techniques [2-7] produced good tradeoff between reductions of adders and critical path lengths (CPLs). The basic philosophy of CSE approach is to eliminate redundant computations in multiplier blocks by employing the most common subexpressions (CSs) that exist in the canonic signed digit (CSD) representation of coefficients. However the CSE techniques have not addressed the issue of minimizing the complexity of each adder of the multiplier, which is significant in low power implementations. In our recent work [8], we have shown that the number of full adders (FAs) needed to realize the adders used in coefficient multipliers can be considerably reduced by using a novel coefficient-partitioning method (CPM). The basic idea of CPM is that when multiplication is realized using shifts and adds, the adder width (number of FAs in an adder) can be minimized by limiting the shifts of the operands to shorter lengths. The CPM offered an average full adder reduction of 30% for FIR filters over conventional CSE methods. In [9], we have proposed a binary subexpression elimination (BSE) method based on binary representation of coefficients which produced improved adder reductions for coefficient multipliers compared to the CSD-based CSE methods [2-7].

In this paper, we apply the coefficient-partitioning technique in [8] to our BSE method [9]. We refer the CPM in [8] as CSE-CPM and the proposed method is called BSE-CPM in the remainder of this paper. The main difference between the method in [8] and the proposed method is that the

former is based on CSD coefficients whereas the latter is based on binary representation of coefficients. We show that proposed BSE-CPM offers good reductions of full adders, power consumption and delay compared to [8] and other existing CSE methods. The remainder of this paper is organized as follows. Section II reviews the coefficient-partitioning technique. The proposed BSE-CPM is presented in Section III. The design examples are shown in Section IV. In Section V, the synthesis results are presented. Section VI provides our conclusions.

II. REVIEW OF COEFFICIENT-PARTITIONING METHOD

In this section, we briefly review the CPM [8]. The key idea in CPM is to reduce the ranges (bit length) of the operands so that the adder width can be reduced which in turn minimizes the number of FAs. Firstly, the CSD coefficient is expressed using CSs and then coded using a pseudo floating-point (PFP) representation. The resulting expression is then partitioned into two parts for further reduction of range.

The coefficient $h_k=0.0000101001010101$, is used as an example to illustrate the CPM. First, the direct-CSE implementation (CSE without coefficient-partitioning) of h_k is discussed, followed by the CPM and their comparison. The pattern [1 0 1] is present thrice in h_k , which can be expressed as a CS, $x_2=x_1+x_1>>2$ where x_1 is the input and ' $x_1>>k$ ' implies right shifting x_1 by k units. Using direct-CSE, the filter tap output y_k can be expressed as

$$y_k = x_2 \gg 5 + x_2 \gg 10 + x_2 \gg 14 \quad (1)$$

Fig. 1 shows the multiplier using direct-CSE. An adder that adds two n -bit numbers requires at the most $(n+1)$ FAs to compute the sum. The adder assumed is a ripple carry adder and the expression for computing the numbers of FAs are given by [8]:

Case I: Odd number of operands: The number of FAs, N_o , required to compute the output corresponding to a coefficient with n operands can be determined using the expression,

$$N_o = (r_2 + 1) + a_1(r_3 + 1) + (2r_4 + 3) + a_3(r_5 + 1) + (r_6 + 1) + a_5(2r_7 + 3) + (3r_8 + 6) + a_7(r_9 + 1) + (r_{10} + 1) + a_9(2r_{11} + 3) \quad (2)$$

where r_n is the range (number of bits) of the n th operand and a_i s are equal to zero except a_{n-2} , which is 1.

Case II: Even number of operands: The number of FAs, (N_e) , required to compute the output corresponding to a coefficient with n operands is given by [7]:

$$N_e = (r_2 + 1) + (2r_4 + 3) + c_0(r_6 + c_0') + (3r_8 + 6) +$$

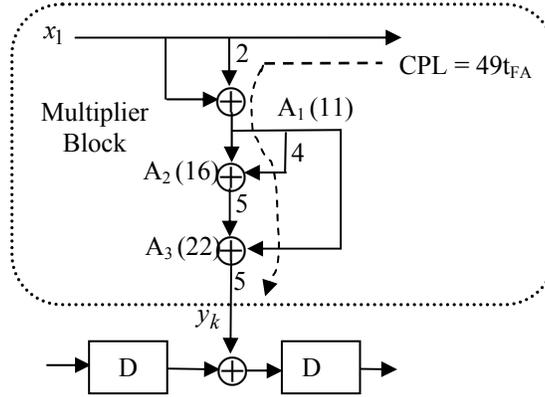


Figure 2. Coefficient multiplier implementation using CPM.

In next section, we apply the coefficient-partitioning technique originally proposed for CSD-based CSE methods [2-7] to our recently proposed BSE and show that improved FA reductions can be achieved using proposed BSE-CPM.

III. PROPOSED BSE-CPM

The BSE method [9] deals with the elimination of redundant Binary Common Subexpressions (BCSs) that occur within the binary representation of coefficients. An n -bit binary number can form $2^n - (n+1)$ BCSs among themselves. For example, a 3-bit binary representation can form 4 BCSs, which are [0 1 1], [1 0 1], [1 1 0] and [1 1 1]. These BCSs can be expressed as [0 1 1] = $x_3 = 2^{-1}x_1 + 2^{-2}x_1$, [1 0 1] = $x_4 = x_1 + 2^{-2}x_1$, [1 1 0] = $x_5 = x_1 + 2^{-1}x_1$, and [1 1 1] = $x_6 = x_1 + 2^{-1}x_1 + 2^{-2}x_1$, where x_1 is the input signal. Note that other BCSs such as [0 0 1], [0 1 0] and [1 0 0] do not require any adder for implementation as they have only one nonzero bit. A straightforward realization of above BCSs would require 5 adders. However x_3 can be obtained from x_5 by a right shift operation (without using any extra adders): $x_3 = 2^{-1}x_1 + 2^{-2}x_1 = 2^{-1}(x_1 + 2^{-1}x_1) = 2^{-1}x_5$. Also, x_6 can be obtained from x_5 using an adder: $x_6 = x_1 + 2^{-1}x_1 + 2^{-2}x_1 = x_5 + 2^{-2}x_1$. Thus only 3 adders are needed to realize the BCSs x_3 to x_6 . The number of adders required for all the possible n -bit binary subexpressions is $2^{n-1} - 1$. The number of adders needed to implement the coefficient multipliers using the BSE [9] is considerably less than the CSD-based CSE methods in [1]-[6].

The issue of minimizing the number of FAs needed to realize each adder was not addressed in [9]. Here, we apply the coefficient-partitioning technique to BSE-coded filter coefficients in order to reduce the number of FAs needed to realize each adder in BSE.

A. BSE-CPM Procedure

The procedure of the proposed BSE-CPM is as follows.

Step 1) Design the filter of length N according to the desired specification.

Step 2) Obtain the binary representation of the coefficients for a desired word length. Set $k = 0$.

Step 3) Identify the BCSs [0 1 1], [1 0 1], [1 1 0], [1 1 1] and [1 0 0 1] in $h(k)$. Express the filter output corresponding to the coefficient $h(k)$ using BCS.

Step 4) Express the filter output corresponding to $h(k)$ in PFP. Set $M = \text{span}$.

Step 5) Partition the span part into two sub-coefficients of length $M/2$ (or lengths $\lfloor M/2 \rfloor$ and $\lceil M/2 \rceil$ if M is odd). Scale the latter sub-coefficients by its *order*.

Step 6) Increment k . If $k \neq N$, go to Step 3. Otherwise, terminate the program.

B. Power Consumption

Let the average power consumed in a single bit full addition (subtraction) is denoted by P_{add} . If N FAs are required for the implementation of symmetric half coefficients of an FIR filter, the net average power dissipated is:

$$P = P_{add} \times N \quad (4)$$

C. Delay

Let τ^n_{add} denotes n bit full adder delay. For comparison, the adders will be modeled as ripple-carry adders consisting of cascaded one-bit full adders, τ^n_{add} can therefore be approximated as:

$$\tau^n_{add} = (N - 1)\tau_{cout} + \tau_{sum} \quad (5)$$

where τ_{cout} is the greatest of the delays from any input of the FA to its *Carry-out* output, τ_{sum} is the delay from the *Carry-in* input of the FA to its *Sum* output and N is the total number of FAs required for that computation. To compute the delay of FIR filter coefficient multiplier, τ^n_{add} is determined for each coefficient and the worst-case value (largest) of τ^n_{add} is obtained. The values of P_{add} , τ_{cout} and τ_{sum} for TSMC 0.18 μm CMOS technology are given in Table I. The values of the parameters in Table I are obtained using HSIM simulation.

TABLE I. CHARACTERIZED CMOS LIBRARY DATA

Simulation Environment				
1000 random data	C_load = 170 fF	Temp. = 25 C	Supply Volt. = 1.8	Data rate = 10ns
Data				
$P_{add} = 40.43$ μW	$\tau_{cout} =$ 0.565ns	$\tau_{sum} = 0.959ns$		

IV. DESIGN EXAMPLES

In this section, we present examples of implementing channel filters for the D-AMPS and the PDC receivers using our BSE-CPM. We provide comparisons with the best known CSE method [6] (which resulted in least number of adders compared to other CSE methods) and the best known algorithm which resulted in least number of full adders [8].

Example 1: The FIR filters employed in the channelizer of the D-AMPS in [10] are considered in this example. The sampling rate of the wideband signal chosen is 34.02 MHz as in [10]. The channel filters extract 30 kHz D-AMPS channels from the wideband signal after downsampling by a factor of 350. The pass-band and stop-band edges are 30 kHz and 30.5 kHz respectively. The peak pass-band ripple specification is 0.1 dB. The peak stop-band ripple (PSR) specifications at different frequencies and respective filter lengths (N) are chosen to be as in the D-AMPS standard. Channel filters with 260, 610, 940 and 1180 taps are chosen to meet the PSR specifications of -48 dB, -65 dB, -85 dB and -96 dB respectively.

The reduction of FAs over the direct implementation in designing the channel filters with 16-bit coefficient wordlength, for different filter lengths (260, 610, 940 and 1180 taps) are shown in Fig. 3. For the filter with 1180 taps (corresponding to the blocking specification of -96 dB), the FA reduction produced by the CSE method [6] and the CSE-CPM [8] are 28.94% and 64.73% respectively whereas proposed BSE-CPM offers a reduction of 75.02%. The average reductions of FAs for different filter lengths achieved using the CSE method [6], CSE-CPM [8] and proposed BSE-CPM are 29.5%, 64.64% and 73.75% respectively. The reduction of FAs over the direct implementation in designing the channel filters with 24-bit coefficient wordlength, for different filter lengths are shown in Fig. 4. For the filter with 1180 taps, the FA reduction produced by the CSE method [6] and the CSE-CPM [8] are 34.32% and 67.2% respectively whereas proposed BSE-CPM offers a reduction of 79.86%. The average reductions of FAs for

different filter lengths achieved using the CSE method [6], CSE-CPM [8] and proposed BSE-CPM are 29.72%, 70.2% and 75.7% respectively.

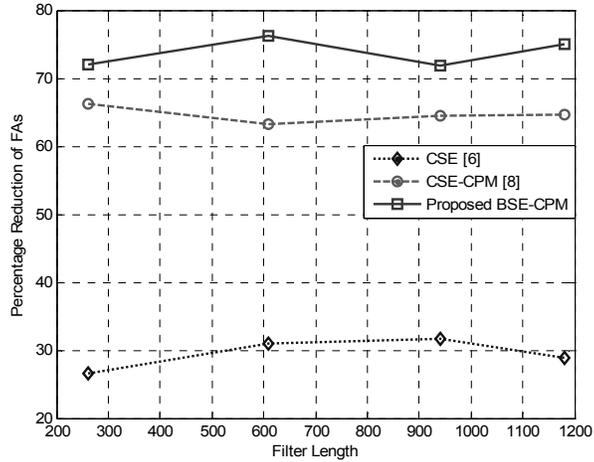


Figure 3. Reduction of FAs in designing the D-AMPS channel filter with 16-bit coefficient word length.

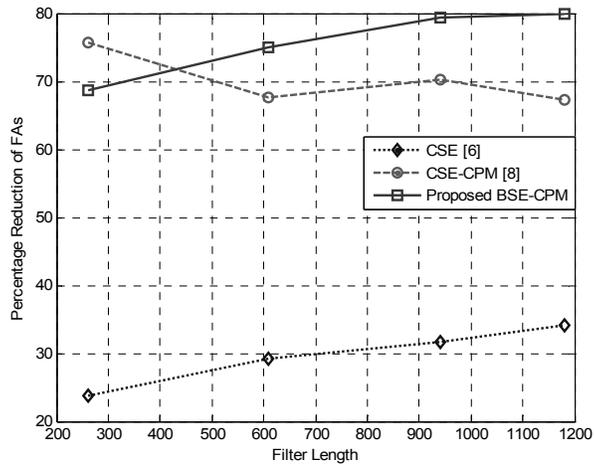


Figure 4. Reduction of FAs in designing the D-AMPS channel filter with 24-bit coefficient word length.

Table II shows the comparison of power consumption and delay of the D-AMPS filters with coefficient wordlength of 16 bits. The average reductions of power consumption achieved using proposed BSE-CPM over the CSE method [6] and the CSE-CPM [8] are 62.7% and 26.3% respectively. The proposed method achieves speed improvements of 62.7% and 26.3% over the CSE method [6] and CSE-CPM [8].

TABLE II. COMPARISON OF POWER CONSUMPTION AND DELAY OF D-AMPS FILTERS FOR 16-BIT COEFFICIENT WORDLENGTH

Taps	CSE [6]		CSE-CPM [8]		Proposed BSE-CPM	
	Power (μ W)	Delay (ns)	Power (μ W)	Delay (ns)	Power (μ W)	Delay (ns)
260	110778	1548.49	51022.7	713.424	42128.1	589.124
610	159618	2231.01	84822.1	1185.76	55065.7	769.924
940	182258	2547.41	94889.2	1326.45	75199.8	1051.29
1180	170089	2377.35	84417.8	1180.11	59796	836.029

Example 2: In this example, the channel filters employed in receivers for the PDC standard are implemented. The sampling rate of the wideband signal is 25.6 MHz, which covers 1024 channels of 25 kHz spacing. The peak pass-band ripple specification is 0.1 dB. Channel filters with 240, 590, 880 and 1000 taps are chosen to meet the PSR specifications of -45 dB, -62 dB, -80 dB and -90 dB at different frequencies as in PDC standard. The reduction of FAs over the direct implementation in designing the channel filters with 16-bit coefficient wordlength, for different filter lengths are shown in Fig. 5. For the filter with 1000 taps (corresponding to the blocking specification of -90 dB), the FA reduction produced by the CSE method [6], CSE-CPM [8] and proposed BSE-CPM are 30.76%, 59.76% and 74.79% respectively. The average reductions of FAs for different filter lengths achieved using the CSE method [6], CSE-CPM [8] and proposed BSE-CPM are 31.25%, 60.7% and 74.54% respectively.

The reduction of FAs over the direct implementation in designing the channel filters with 24-bit coefficient wordlength, for different filter lengths are shown in Fig. 6. For the filter with 1000 taps, the FA reduction produced by the CSE method [6], CSE-CPM [8] and proposed BSE-CPM are 33%, 71.6% and 77.1% respectively. The average reductions of FAs for different filter lengths achieved using the CSE method [6], CSE-CPM [8] and proposed BSE-CPM are 29.6%, 70.03% and 74.7% respectively.

It can be seen from Figures 3 to 6 that, the FA reductions achieved using the proposed BSE-CPM are larger for higher order filters (greater than 450 taps). The reason for greater FA reductions achieved using our BSE-CPM over CSD-CSE can be explained as follows. The cost of any CSE method (including our BSE) mainly depends on three factors – the total number of non-zero bits in the coefficient set N_{nz} , the number of common subexpressions (CSs) that can be formed from the non-zero bits N_{cs} and the number of unpaired bits N_{up} (bits that do not form CSs). Then the number of adders required in CSE technique, N_A , is given by

$$N_A = (\alpha \times N_{nz} - \beta \times N_{cs} + \gamma \times N_{up}) \quad (6)$$

where α , β and γ are the weights of N_{nz} , N_{cs} and N_{up} respectively. In [9], we have done a statistical analysis of coefficients for FIR filters of various passband and stopband edge specifications to determine the impact of above three factors on the number of adders needed to implement the coefficient multipliers. Filters of different lengths (20, 50, 80, 120, 200, 400 and 800 taps) and wordlengths (12, 16, 20 and 24 bits) were analyzed for obtaining the weights α , β and γ in (6). Based on our statistical analysis of above filters, we obtained the average values of α , β and γ as 0.2345, 0.6643 and 4.0487 respectively. It can be noted from (6) that, the weight for N_{up} is substantially larger compared to the weights of N_{nz} and N_{cs} . Therefore the number of adders is largely dependent on N_{up} . In [9], we have also compared the average N_{up} values for the binary and CSD filter coefficient representations. It was found that, in the case of CSD representation, the values of N_{up} are found on the higher side compared to binary representation of filter coefficients especially for higher order filters and the average reduction of N_{up} for binary filter coefficients over CSD values is found to be 67%. This larger reduction of N_{up} for binary representation of coefficients for higher order filters account for the reduction of number of adders (N_A) and consequently the number of FAs in proposed BSE-CPM.

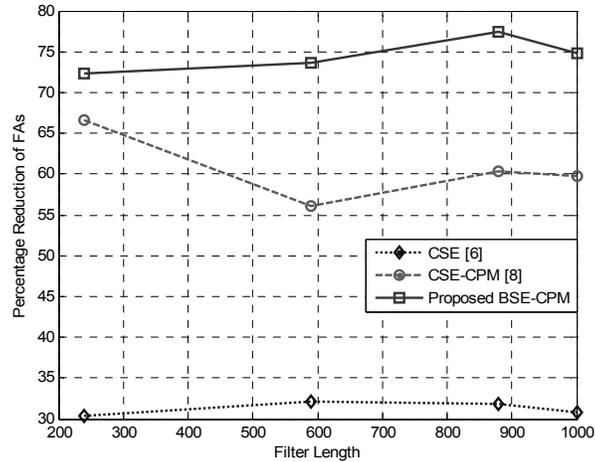


Figure 5. Reduction of FAs in designing the PDC channel filter with 16-bit coefficient word length.

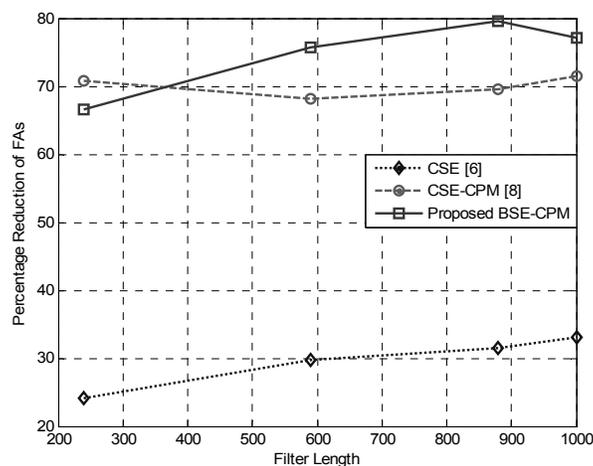


Figure 6. Reduction of FAs in designing the PDC channel filter with 24-bit coefficient word length.

TABLE III. COMPARISON OF POWER CONSUMPTION AND DELAY OF D-AMPS FILTERS

Taps	CSE [6]		CSE-CPM [8]		Proposed BSE-CPM	
	Power (μ W)	Delay (ns)	Power (μ W)	Delay (ns)	Power (μ W)	Delay (ns)
240	96829.9	1353.57	46413.6	649.014	38448.9	537.709
590	142960	1998.23	92463.4	1292.55	55591.3	777.269
880	167016	2334.41	97112.9	1357.52	55227.4	772.184
1000	175102	2447.41	101762	1422.5	63758.1	891.399

Table III shows the comparison of power consumption and delay of the PDC filters. The average reductions of power consumption achieved using proposed BSE-CPM over the CSE method [6] and the CSE-CPM [8] are 63.4% and 36.9% respectively. The proposed method achieves speed improvements of 63.45% and 37% over the CSE method [6] and CSE-CPM [8].

V. SYNTHESIS RESULTS

In this section, the synthesis results of the proposed BSE-CPM method are presented and compared with NR-SCSE [6] and CSD-CPM [8]. We have done the synthesis using Synopsis tool for three different filter specifications of T1, T2 and T3 on 0.18 μ m CMOS technology. T1 represents a 60 tap lowpass filter with passband and stopband specifications of $\omega_p = 0.021\pi$ and $\omega_s = 0.07\pi$ respectively. T2 and T3 represent 90 tap and 120 tap lowpass filters with passband and stopband specifications of $\omega_p = 0.15\pi$ and $\omega_s = 0.25\pi$ respectively. The wordlength of the

filter coefficients is fixed as 14 bits in all the cases. The synthesis results are shown in Table IV. From Table IV, it can be seen that the proposed method shows inferior results compared to NR-SCSE [6] and CSD-CPM [8] for lower order filter, T1. But for the filter specifications, T2 and T3, which is having higher order, the proposed method results in better area reduction and power consumption than NR-SCSE [6] and CSD-CPM [8]. The proposed BSE-CPM method offers area reductions of 8.85% over NR-SCSE [6] and 3.5% over CSD-CPM [8]. Similarly, the proposed BSE-CPM method offers average power reductions of 22.11% over NR-SCSE [6] and 0.95% over CSD-CPM [8]. It can be concluded from Table IV that, for higher order filters, the proposed BSE-CPM method offers much better reduction in area and power than the methods in [6] and [8].

TABLE IV. SYNTHESIS RESULTS OF CSE ALGORITHMS FOR DIFFERENT FILTER SPECIFICATIONS

Filter	CSE Algorithms	Area in μm^2	Power in mW	Delay in ns
T1 (60-TAP, 0.021 Π -0.07 Π , WORDLENGTH 14-BITS)	NR-SCSE [6]	121787.41	7.0037	5.6
	CSD-CPM [8]	148450.58	3.9015	5.69
	Proposed BSE-CPM	158366.57	4.1942	6.37
T2 (90-TAP, 0.15 Π -0.25 Π , WORDLENGTH 14-BITS)	NR-SCSE [6]	259875.0	6.9362	5.85
	CSD-CPM [8]	259099.95	6.9398	6.62
	Proposed BSE-CPM	249922.42	6.874	6.73
T3 (120-TAP, 0.15 Π -0.25 Π , WORDLENGTH 14-BITS)	NR-SCSE [6]	278761.65	7.8945	5.67
	CSD-CPM [8]	244972.73	6.6193	5.76
	Proposed BSE-CPM	240142.8	6.0161	6.16

In general, the larger the number of filter taps, the higher will be the power consumption. This is based on the fact that, the number of multiplications in a filter is directly proportional to the number of filter taps (coefficients). When the multiplications are implemented using shifters and adders, the power consumption is dominated and proportional to the number of adders (shifters are relatively less complex than adders). The number of adders needed to implement the multiplier in turn depends on the number of nonzero bits in the binary representation of the coefficients. When common subexpression elimination (CSE) is employed to reduce the number of adders, the actual number of final adders needed (i.e., adders after applying CSE) is inversely proportional to the number of nonzero bits that form the subexpressions and directly proportional to the number of nonzero bits that are unable to form subexpressions. Thus the adder reduction depends not only on the number of filter taps but also on the distribution of nonzero bits in the coefficients (subexpression formation is dependent on the distribution of nonzero bits). We have noticed that, even though the number of taps is more (1180 taps compared to 940 taps in Table II and 120 taps compared to 90 taps in Table IV), these longer filters have following advantages (based on statistical observations):

- (1) Ability to form more binary subexpressions (compared to relatively shorter filters) leaving fewer number of ungrouped nonzero bits.
- (2) The new coefficients are either zero or they possess fewer numbers of nonzero bits with all nonzero bits clustered near the least significant bit locations due to their low magnitude values. This is because the magnitudes of the end-coefficients will decrease with the filter length (number of taps) and only fewer nonzero bits are needed to encode the small magnitude end-coefficients and these nonzero bits would occur as LSB bits (closely clustered). ‘End-coefficients’ are the last few coefficients of a filter. For example, the coefficients $h(45)$ to $h(59)$ of a 120-tap filter are ‘end-coefficients’ (Note that the coefficients $h(0)$ to $h(59)$ are symmetric with $h(60)$ to $h(119)$ for an FIR filter).

Both the above-mentioned advantages account for the reduced power consumption for 1180 taps compared to 940 taps and 120 taps compared to 90 taps.

VI. CONCLUSIONS

We have proposed a coefficient-partitioning based binary subexpression elimination method for implementing low power and high-speed channel filters for wireless communication receivers.

The design examples show that the proposed method achieved 23% average reductions of full adder and power consumption and 23.3% reduction of delay over the best existing method (CSE-CPM [8]). Our method could be employed to realize any FIR filters, and need not be restricted to wireless communications. However, the hardware reductions achieved using BSE method are superior particularly for higher-order FIR filters. Therefore, our method in this paper is best suited for channel filters of wireless communication receivers, which require large number of taps to meet the stringent adjacent channel attenuation specifications. Our future work would focus on incorporating reconfigurability to the proposed method by suitably modifying the constant shifts method (CSM) based reconfigurable filter architecture that we recently proposed in [11] for software radio receiver applications.

REFERENCES

- [1] J. Mitola, *Software Radio Architecture*. New York: Wiley, 2000.
- [2] M. Mehendale, S. D. Sherlekar, and G. Venkatesh, "Synthesis of multiplierless FIR filters with minimum number of additions," in *Proceedings of the 1995 IEEE/ACM International Conference on Computer-Aided Design*, Los Alamitos, CA: IEEE Computer Society Press, pp. 668-671, 1995.
- [3] M. Yagyu, A. Nishihara, and N. Fujii, "Fast FIR digital filter structures using minimal number of adders and its application to filter design," *ICICE Trans. Fundam. Electron. Commun. Comput. Sci.*, no. 8, pp. 1120-1129, E79-A, 1996.
- [4] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. on Circuits and Systems- II*, vol. 43, pp. 677-688, Oct. 1996.
- [5] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Syst.*, vol. 18, no. 1, pp. 58-68, Jan. 1999.
- [6] M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," *IEEE Trans. on Circuits and Systems- II*, vol. 49, no. 3, pp. 196-203, Mar. 2002.
- [7] C. Y. Yao, H. H. Chen, C. J. Chien and C. T. Hsu, "A Novel Common-Subexpression-Elimination Method for Synthesizing Fixed-Point FIR Filters," *IEEE Trans. on Circuits and Systems-I*, vol. 51, No.11, pp. 2215-2221, Nov. 2004.
- [8] A.P.Vinod and E.M-K.Lai, "An efficient coefficient-partitioning algorithm for realizing low complexity digital filters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1936-1946, Dec 2005.
- [9] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low complexity higher order digital filters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, vol. 27, no. 2, pp. 217-229, Feb. 2008.
- [10] K. C. Zangi, R. D. Koilpillai, "Software radio issues in cellular base stations," *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 4, pp. 561-573, April 1999.
- [11] R. Mahesh and A. P. Vinod, "An architecture for integrating low complexity and reconfigurability for channel filters in software defined radio receivers," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 2514-2517, New Orleans, USA, May 2007.

Reviewer's Comments (RC) and Author Reply (AR)

The authors would like to thank the anonymous reviewers for their valuable comments which helped to improve the quality of the paper. Author reply (RC) to Reviewers' comments (RC) is given below.

Reviewer No. 1

RC: It may be difficult for readers to confirm that the proposed method can expand the flexibility of SDR receivers. Authors should clarify the benefits of the proposed method for SDR. Authors claimed that channelization filters for SDR receivers must achieve low power consumption and high speed. In addition, filters for SDR receivers must have flexibility for adapting receivers to various radio communication systems. But according to the reason as follows, it seems difficult to change coefficients of the filters designed by proposed method.

- The procedure for changing filter coefficients is not shown in the paper.

- It seems that the assumption of synthesis is based on non-reconfigurable devices such as ASIC.

Please explain detail of the flexibility of the filters utilizing proposed method.

AR: We agree that the reconfigurability issues are not covered in the paper. Reviewer-2 also pointed out this aspect and suggested to generalize the paper title and material as a low power technique for implementing FIR channel filters (by removing the term 'SDR'). The proposed method can be used to realize channel filters in wireless communication receivers where the filter specifications are known beforehand. Accordingly, the term 'SDR' has been replaced by the term 'wireless communication receiver' in the title and the text. Also, we have revised the conclusions section (Section VI) with a remark on the possibility of incorporating reconfigurability to the proposed method using our recently proposed reconfigurable architecture in [11].

The last part of Section VI (page 14) has been revised by adding "Our future work would focus on incorporating reconfigurability to the proposed method by suitably modifying the constant shifts method (CSM) based reconfigurable filter architecture that we recently proposed in [11] for software radio receiver applications".

RC: In Fig. 4, the number of full adder of BSE-CPM is larger than that of CSE-CPM when the filter length is shorter than 400. However, in Table II, power consumption and delay property of BSE-CPM is better than that of CSE-CPM. I think that the number of FA and power and delay property must be proportional. Eqs. 4 and 5 of the paper also support my understandings. Please explain the reason of this inverse proportion.

AR: Fig. 4 shows the percentage reduction of FAs in designing the D-AMPS channel filter with 24-bit coefficient word length. But Table II shows the power consumption for D-AMPS channel filter with 16-bit coefficient wordlength. Note that the power consumption and delay results shown in Table II correspond to the FA reduction of the 16-bit filter coefficients shown in Fig. 3 (not that of the 24-bit filter coefficients in Fig. 4). The power reductions of 16-bit filter coefficients shown in Table II are proportional to the FA reduction of 16-bit filter coefficients shown in Fig. 3 (Fig. 3 clearly shows the number of FAs is less for the proposed BSE-CPM compared to CSE-CPM). Therefore, the power consumption results are in line with equations 4 and 5.

In order to have better clarity, we have amended the caption of Table II as "Comparison of Power Consumption and Delay of D-AMPS filters for 16-bit coefficient wordlength".

The caption of Figure 4 is amended as “Reduction of FAs in designing the D-AMPS channel filter with 24-bit coefficient word length”.

RC: *Table II shows the result that 1180-tap filter consumes less power than 940-tap filter. But, in general, longer-length filter consume more power than shorter-length filter. Please explain the reason that the longer-length filter can achieve better power consumption and delay characteristic than the shorter-length filter. Authors also should explain about results shown by Table IV, where T3 (120-tap) achieve better power consumption and delay characteristics than T2 (90-tap).*

AR: We agree that, in general, the larger the number of filter taps, the higher will be the power consumption. This is based on the fact that, the number of multiplications in a filter is directly proportional to the number of filter taps (coefficients). When the multiplications are implemented using shifters and adders, the power consumption is dominated and proportional to the number of adders (shifters are relatively less complex than adders). The number of adders needed to implement the multiplier in turn depends on the number of nonzero bits in the binary representation of the coefficients. When common subexpression elimination (CSE) is employed to reduce the number of adders, the actual number of final adders needed (i.e., adders after applying CSE) is inversely proportional to the number of nonzero bits that form the subexpressions and directly proportional to the number of nonzero bits that are unable to form subexpressions. Thus the adder reduction depends not only on the number of filter taps but also on the distribution of nonzero bits in the coefficients (subexpression formation is dependent on the distribution of nonzero bits). We have noticed that, even though the number of taps is more (1180 taps compared to 940 taps in Table II and 120 taps compared to 90 taps in Table IV), these longer filters have following advantages (based on statistical observations):

- (3) Ability to form more binary subexpressions (compared to relatively shorter filters) leaving fewer number of ungrouped nonzero bits.
- (4) The new coefficients are either zero or they possess fewer numbers of nonzero bits with all nonzero bits clustered near the least significant bit locations due to their low magnitude values. This is because the magnitudes of the end-coefficients will decrease with the filter length (number of taps) and only fewer nonzero bits are needed to encode the small magnitude end-coefficients and these nonzero bits would occur as LSB bits (closely clustered). ‘End-coefficients’ are the last few coefficients of a filter. For example, the coefficients $h(45)$ to $h(59)$ of a 120-tap filter are ‘end-coefficients’ (Note that the coefficients $h(0)$ to $h(59)$ are symmetric with $h(60)$ to $h(119)$ for an FIR filter).

Both the above-mentioned advantages account for the reduced power consumption for 1180 taps compared to 940 taps and 120 taps compared to 90 taps.

To clarify above mentioned point, we have added above explanation to the last paragraph of Section V on page 13.

Reviewer No. 2

RC: *The title should be changed so as not to include the software radio (SDR), and also the concept of SDR should be eliminated from all of the text. To reviewer, this paper seems to present the filter coefficient algorithm and no concept of SDR is connected even in Section V. It is misleading for readers.*

AR: The term ‘SDR’ has been replaced by the term ‘wireless communication receiver’ in the title and the text. We agree that the reconfigurability issues are not covered in the paper. The proposed method can be used to realize channel filters in wireless communication receivers where the filter specifications are known beforehand. Also, we have revised the conclusions section (Section VI) with a remark on the possibility of incorporating reconfigurability to the proposed method using our recently proposed reconfigurable architecture in [11].

The last part of Section VI (page 14) has been revised by adding “Our future work would focus on incorporating reconfigurability to the proposed method by suitably modifying the constant shifts method (CSM) based reconfigurable filter architecture that we recently proposed in [11] for software radio receiver applications”.

RC: *In Section IV, only the reduction percentages are given but no consideration is discussed. The authors should give the evaluation and discussion for the results of Figs. 3 to 6, e.g., why is the performances of BSE-CPM improved according to the filter length (except 1000)?*

AR: We revised the paper by including the analysis of the results shown in Figures 3 to 6. The following paragraph has been added for clarification in Section IV, last paragraph of Example 2 on pages 9 and 10:

“It can be seen from Figures 3 to 6 that, the FA reductions achieved using the proposed BSE-CPM are larger for higher order filters (greater than 450 taps). The reason for greater FA reductions achieved using our BSE-CPM over CSD-CSE can be explained as follows. The cost of any CSE method (including our BSE) mainly depends on three factors – the total number of non-zero bits in the coefficient set N_{nz} , the number of common subexpressions (CSs) that can be formed from the non-zero bits N_{cs} and the number of unpaired bits N_{up} (bits that do not form CSs). Then the number of adders required in CSE technique, N_A , is given by

$$N_A = (\alpha \times N_{nz} - \beta \times N_{cs} + \gamma \times N_{up}) \quad (6)$$

where α , β and γ are the weights of N_{nz} , N_{cs} and N_{up} respectively. In [9], we have done a statistical analysis of coefficients for FIR filters of various passband and stopband edge specifications to determine the impact of above three factors on the number of adders needed to implement the coefficient multipliers. Filters of different lengths (20, 50, 80, 120, 200, 400 and 800 taps) and wordlengths (12, 16, 20 and 24 bits) were analyzed for obtaining the weights α , β and γ in (6). Based on our statistical analysis of above filters, we obtained the average values of α , β and γ as 0.2345, 0.6643 and 4.0487 respectively. It can be noted from (6) that, the weight for N_{up} is substantially larger compared to the weights of N_{nz} and N_{cs} . Therefore the number of adders is largely dependent on N_{up} . In [9], we have also compared the average N_{up} values for the binary and CSD filter coefficient representations. It was found that, in the case of CSD representation, the values of N_{up} are found on the higher side compared to binary representation of filter coefficients especially for higher order filters and the average reduction of N_{up} for binary filter coefficients over CSD values is found to be 67%. This larger reduction of N_{up} for binary representation of coefficients for higher order filters account for the reduction of number of adders (N_A) and consequently the number of FAs in proposed BSE-CPM”.