

Low-Complexity Filter Bank Channelizer for Wideband Receivers Using Minimum Adder Multiplier Blocks

A.P.Vinod, E.M-K.Lai, A.B.Premkumar and C.T.Lau

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
Email: asvinod@ntu.edu.sg

Abstract— The computational complexity of linear phase finite impulse response (LPFIR) filters used in the channelizer of a wideband receiver is dominated by the number of adders (subtractors) employed in the multipliers. Common subexpression elimination (CSE) is a well-known technique for minimizing the number of adders in LPFIR filters. An improved CSE method is proposed in this paper, which is used to implement the channel filters of a filter bank channelizer (FBC). In the FBC, each modulated bandpass filter extract one channel from the input wideband signal. The reduction in number of adders is obtained by eliminating redundant multiplications of common subexpressions that exist among the channel filters of the FBC with the input signal. Design example of the channel filters employed in the Digital Advanced Mobile Phone System (D-AMPS) show that the proposed method offers considerable reduction in the number of full adders when compared with conventional CSE methods.

Keywords—channelizer; common subexpression elimination; LPFIR filters

I. INTRODUCTION

The most computationally intensive part of a wideband receiver is the FBC since it operates at the highest sampling rate [1]. Channelization in wideband receivers involves the extraction of multiple narrowband channels from a wideband signal using several bandpass filters called channel filters. High-speed/low-power LPFIR filters implemented with the minimum number of adders are required in FBC. It has been shown that the redundancy across the canonic signed digit (CSD) coefficients can be exploited by employing CSE to implement LPFIR filters with a minimum number of adders [2]-[4]. A method to eliminate the most commonly occurring 2-bit common subexpressions (CS), [1 0 1] and [1 0 -1], was proposed in [2]. (Note that an n -bit CS is defined as a subexpression that has n non-zero bits). In [3], a nonrecursive signed CSE algorithm has been proposed as a modification of the technique in [2], that minimizes the critical path length of the filter structure. A CSE technique based on vertical CS for filters with relatively small coefficient wordlength is presented in [4]. In general, the CSE methods proposed in [2]-[4] eliminate redundant computations in multiplier blocks of

LPFIR filters by employing the most common subexpressions consisting of two non-zero bits, i.e., [1 0 1] and [1 0 -1]. In this paper, a super-subexpression elimination (SSE) technique for optimizing conventional CSE method to further reduce the number of adders in LPFIR filters is proposed. Further, the CSE and SSE methods are applied to implement FBC, where CS that occur among the coefficients of several bandpass filters are utilized for a minimum adder implementation.

The paper is organized as follows. Section 2 provides a review of the CSE method. The complexity of implementation is analyzed in terms of full adders required for each multiplier of the filter. The SSE algorithm is presented in section 3. In section 4, we illustrate the implementation of channel filters for the D-AMPS standard using the SSE technique and provide comparisons with conventional methods. Section 5 provides our conclusions.

II. THE CSE APPROACH

A. Review of CSE

An 8-tap LPFIR filter with coefficients expressed in 16-bit CSD form shown in Fig. 1 is used as an example to illustrate the CSE method. The number of adders required for a conventional CSD implementation of a LPFIR filter with N taps is:

$$N_{CSD} = (N_b - 1) + \lfloor N/2 \rfloor \quad (1)$$

where N_b is the number of nonzero bits in one half of the symmetric coefficient set. In this example, $N_b = 29$, and $N = 8$. Hence thirty-two adders would be required to implement the filter without using CSE. The objective of CSE algorithm is to identify multiple identical bit patterns in the coefficient set and eliminate redundant computations by forming CS from the bit patterns. The 2-bit CS, [1 0 1] and [1 0 -1], shown encircled in Fig. 1 are given by:

$$x_2 = x_1 + x_1 \gg 2 \text{ and } x_3 = x_1 - x_1 \gg 2 \quad (2)$$

where x_1 is the input signal and ‘ \gg ’ represents shift right operation. If N_{CS} is the number of CS in one half of the

symmetric coefficient set and N_{acs} is the number of adders required for distinct CS, the CSE method would require $(N_{cs} - N_{acs})$ adders fewer than conventional CSD implementation. Hence number of adders required to implement the filter using CSE obtained by modifying (1) is

$$N_{CSE} = (N_b - 1) + \lfloor N/2 \rfloor - (N_{cs} - N_{acs}) \quad (3)$$

In this case $N_{cs} = 13$, $N_{acs} = 2$, and $N_{CSE} = 21$. This offers a reduction of 34% over direct CSD implementation without CSE.

B. Adder Complexity

To the best of our knowledge, all the previous works on CSE discussed hardware reduction in terms of the number of adders and have not addressed the complexity of adders. The complexity of each adder employed in multiplication is significant for high-speed/low-power implementations. An adder that adds two n -bit numbers requires n full adders (FA) to compute the sum. The area, power, and speed of an adder depend on the value of n , which is called the *adder width*. Efforts to optimize these parameters should focus on minimizing the adder width, i.e., the number of FA.

Definition 1 (Nonzero terms): The CS and the nonzero bits other than the CS of a coefficient are termed as its nonzero terms. For example, the two nonzero terms of a coefficient represented in CSD, (0.1010001), are [101] (CS) and 1 (least significant bit).

Definition 2 (Operands): The input signal shifted corresponding to the positional weights of the nonzero terms of the coefficient form the operands of the adders. For instance, in the case of the coefficient, (0.1010001), the operands are $x_2 \gg 1$ and $x_1 \gg 7$, where x_1 and x_2 are as in (2). The number of adders required to compute the output for a coefficient is equal to one less than the number of operands.

Definition 3 (Span): The span is defined as the number of bits of an operand. If x_1 is an 8-bit signal, the span of the operand, $x_1 \gg 7$ is fifteen. For an adder whose operands have spans s_1 and s_2 such that $s_2 > s_1$, the adder width is s_2 .

Coefficients in CSD form with wordlengths up to 24-bits are considered for analyzing the adder complexity. Since no adjacent bits in CSD are one's, a 24-bit CSD number can have a maximum of 12 nonzero bits and hence at the most twelve nonzero operands could occur in multiplication.

Case I: Odd number of operands: Using a paper and pencil method, it can be shown that the number of FA's, (N_o), required to compute the output corresponding to a coefficient with n (for n odd) operands can be determined using the expression:

$$N_o = s_2 + a_1 s_3 + 2s_4 + a_3 s_5 + s_6 + a_5 2s_7 + 3s_8 + a_7 s_9 + s_{10} + a_9 2s_{11} + 2s_{12} \quad (4)$$

where s_n is the span of the n th operand and a_i 's are equal to zero except $a_{n-2} = 1$. (The proof of this expression is omitted

due to space constraints). For instance, if 7 operands are present, using (4) we get $N_o = s_2 + 2s_4 + s_6 + 2s_7$.

Case II: Even number of operands: The number of FA's, (N_e), required to compute the output corresponding to a coefficient with n operands ($n \leq 12$) is given by:

$$N_e = s_2 + 2s_4 + c_0 s_6 + 3s_8 + c_1 s_{10} + 3s_{12} \quad (5)$$

where $c_0 \equiv \begin{cases} 2, & \text{for } n = 6 \\ 1, & \text{elsewhere} \end{cases}$ and $c_1 \equiv \begin{cases} 2, & \text{for } n = 10 \\ 1, & \text{elsewhere} \end{cases}$.

For example, if six operands are present (i.e., $n = 6$), it would require $(s_2 + 2s_4 + 2s_6)$ FA's. Using (4) and (5), the total number of FA's required to compute the partial products in the CSE implementation of the LPFIR filter of Fig. 1 is 376. In the next section, we present an optimization technique that minimizes the number of FA's.

III. OPTIMIZATION OF CSE METHOD

Our method utilizes 3-bit and 4-bit super-subexpressions (SS) formed from CS to eliminate redundant computations. Note that when multiplication is performed using shifts and adds, the adder width can be minimized if additions are done prior to shifts. In CSE implementation, the adders employed for CS have shorter widths since the shift operations for obtaining the final partial products are performed after the addition done at the CS stage. The proposed SSE method performs shift operations after additions at two stages - first at the CS stage and then at the SS stage. Therefore the adders at these two stages have shorter adder widths.

The SSE technique is as follows. First, the 2-bit CS are extracted from the coefficient set represented in CSD. These CS are then examined for multiple occurrences of identical shifts with a nonzero bit or with another CS within the same coefficient to form 3-bit and 4-bit SS respectively. Consider the example in Fig. 1, where CS are given by (2). Note that the following SS can be formed as indicated by interconnecting lines in Fig. 1:

1. The CS [1 0 1] and -1 with a shift of one unit between them to form a 3-bit SS, [1 0 1 0 -1]:

$$x_4 = x_2 - x_1 \gg 4 \quad (6)$$

2. The CS [1 0 1] and [1 0 -1] with two shift units between them to form a 4-bit SS, [1 0 1 0 0 1 0 -1]:

$$x_5 = x_2 + x_3 \gg 5 \quad (7)$$

3. The CS [-1 0 1] and [1 0 1] with one shift unit between them to form a 4-bit SS, [-1 0 1 0 1 0 1]:

$$x_6 = -x_3 + x_2 \gg 4 \quad (8)$$

Note that several 'shifted and delayed' versions of these SS occur in the coefficient set, which can be obtained without using extra adders. We observe that several SS exist in the case of channel filters where the number of taps is large and the wordlength is larger. We have investigated several examples of LPFIR filters with taps ranging from 100 to 1200

corresponding to different stop-band attenuation specifications. The infinite-precision filter coefficients were generated by the Parks-McClellan FIR filter design using “remez” function in MATLAB®. Filter coefficients represented in CSD form for different wordlengths of 12-bits, 16-bits, and 24-bits were considered. From the fifty filters that we have designed, it has been observed that among the possible SS, the 3-bit expressions [10101], [1010-1], [-10101], [-1010-1] and their negated versions are the most common SS. Statistically, these 3-bit expressions and their negated versions form around 70% of all the possible SS. These account for the major reduction of adders in the proposed SSE method. Employing the SS (6)-(8), the output of the filter can be written as

$$\begin{aligned}
& x_4 \gg 1 + x_5 \gg 8 + x_1[-1] \gg 2 + x_6[-1] \gg 4 + x_4[-1] \gg 12 \\
& + x_5[-2] \gg 2 + x_3[-2] \gg 13 + x_6[-3] \gg 1 + x_5[-3] \gg 9 + \\
& x_6[-4] \gg 1 + x_5[-4] \gg 9 + x_5[-5] \gg 2 + x_3[-5] \gg 13 + \\
& x_1[-6] \gg 2 + x_6[-6] \gg 4 + x_4[-6] \gg 12 + x_4[-7] \gg 1 + \\
& x_5[-7] \gg 8 \tag{9}
\end{aligned}$$

Fig. 2 shows the filter structure using the SSE method. In Fig. 2, the numerals adjacent to the data path represent the number of bit-wise right shifts. Note that only seventeen adders are required to implement the filter. This offers a reduction of 19% over CSE. In the proposed method, the number of adder-steps (adder-step here being defined as one addition stage in a maximal path of decomposed multiplications) required to compute the partial products is four, which is the same as that of the CSE method. Thus, both methods have identical critical path lengths. Using SSE, the number of FA’s required to compute the partial products of the LPFIR filter whose coefficients are given in Fig. 1 is 253, which offers a reduction of 32.7% over the CSE method.

IV. DESIGN EXAMPLE

The channel filters of an FBC need sufficiently large number of taps to meet the stringent adjacent channel interference specifications. We extend the SSE method proposed for individual LPFIR filters to FBC for multiplication of one variable (wideband signal) with multiple constants (coefficients) of a bank of bandpass filters. The idea is illustrated in Fig. 3. The most frequently occurring CS among the coefficients of M channel filters are identified to form a multiplier block. Further optimization of the multiplier block can be achieved using the proposed SSE method.

The LPFIR filters employed in the FBC of the D-AMPS in [5] are considered. The sampling rate of the wideband signal chosen is 34.02 MHz as in [5]. 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 [6]. We applied the proposed SSE method to

implement the filters using 12-bit and 16-bit CSD coefficients. The 3-bit and 4-bit SS formed from the 2-bit CS are utilized for optimization. Comparison of the numbers of adders (NA) and full adders (NFA) required to implement the multipliers for the filter using the CSE method [2] and the proposed SSE method are shown in Table I. The percentage reduction of adders (AR) and that of FA’s (FAR) with respect to conventional CSD implementation without using any CS methods are also shown in Table I. For the filter with 1180 taps implemented using 16 bits (corresponding to the most stringent blocking specification), the proposed method offers a reduction of 62.1%, whereas CSE method offers only 35.6% reduction. The proposed method results in considerable average reduction of adders and full adders for different filter lengths as shown in Table II.

The reduction achieved when the proposed method is used to employ the D-AMPS channelizer, where extraction of each channel requires a separate narrowband filter, is examined. The wideband signal considered for channelization consists of 1134 D-AMPS channels, each occupying 30 kHz. We analyzed the requirement of adders to implement the filters for extracting 70, 141, 283, 567, and 1134 channels. The number of filter taps chosen is 1180 and the coefficient wordlength considered is 16 bits. Simulation results shown in Fig. 4 depict the adder reduction achieved using the CSE and SSE methods over conventional CSD implementation as a function of the number of extracted channels. Note that the SSE method offers considerable hardware reduction when compared with the CSE method.

V. CONCLUSIONS

We have proposed an improved technique based on the CSE method to efficiently implement low-complexity channel filters for wideband receivers. The complexity in adders is analyzed and expressions for determining the number of FA’s required for each adder in a filter are obtained. The design example of the FBC based on D-AMPS standard shows that the proposed method offers an average reduction rate of 50% over conventional channel filter implementations without using any CS methods and 20% over CSE implementations.

	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16		
$h(0)$	1	0	1	-1			1	0	1			1	0	-1			
$h(1)$		1		-1	0	1	1	0	1			1	0	1	-1	
$h(2)$			1	0	1			1	0	-1				1	0	-1		
$h(3)$	-1	0	1	1	0	1			1	0	1			1	0	-1	
$h(4)$	-1	0	1	1	0	1			1	0	1			1	0	-1	
$h(5)$		1	0	1				1	0	-1				1	0	-1		
$h(6)$			1		-1	0	1	1	0	1			1	0	1	-1
$h(7)$	1	0	1	-1			1	0	1			1	0	-1			

Figure 1. HCS and HSS in 8-tap LPFIR filter coefficients HSS: $\blacklozenge \cdots \blacklozenge$: x_4 , $\bullet \cdots \bullet$: x_5 , \longleftrightarrow : x_6 .

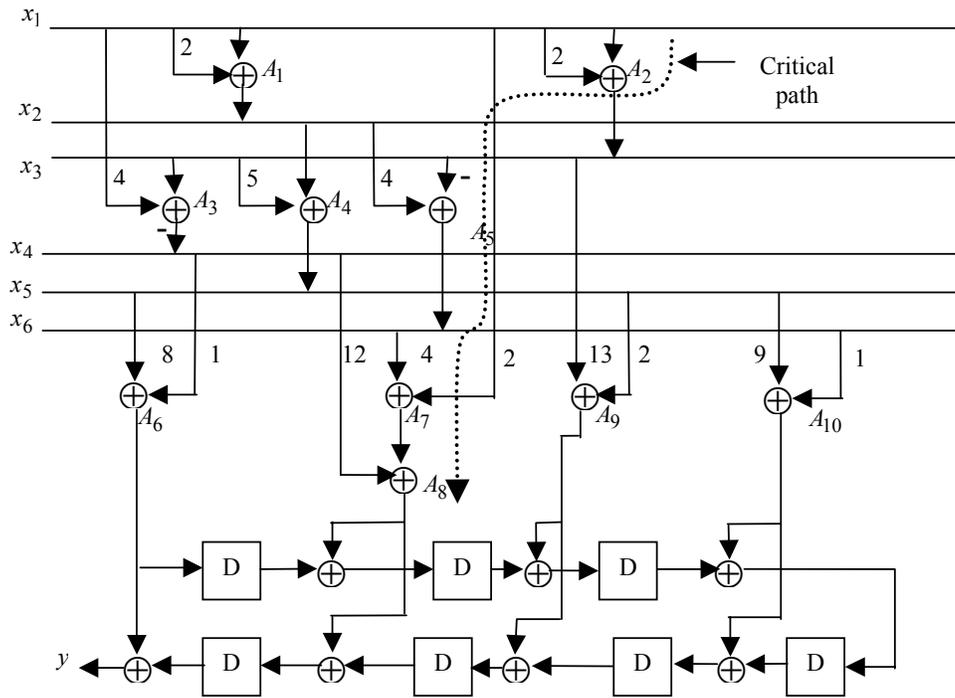


Figure 2. Proposed filter structure using super-subexpressions of Fig. 1.

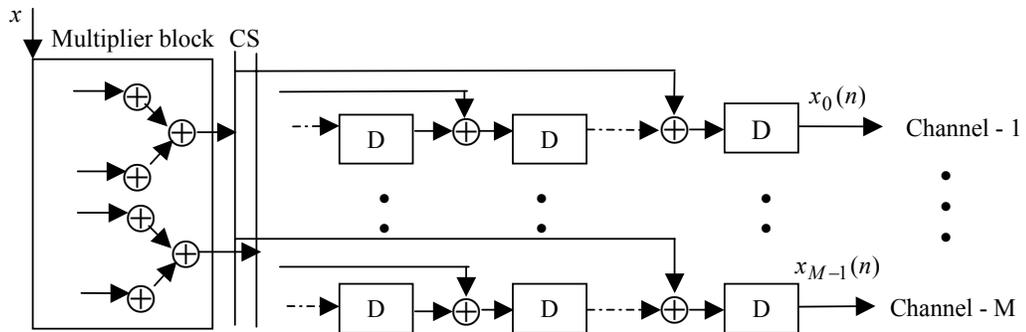


Figure 3. CSE implementation of channel filters in a filter bank channelizer.

TABLE I. COMPARISON OF THE NUMBERS OF ADDERS (NA) AND FULL ADDERS (NFA) REQUIRED TO IMPLEMENT THE MULTIPLIERS FOR THE FILTER IN DESIGN EXAMPLE

	Filter length, N=610 (PSR=-65dB)				Filter length, N=940 (PSR=-85dB)				Filter length, N=1180 (PSR=-96dB)			
	CSE [2]		Proposed SSE		CSE [2]		Proposed SSE		CSE [2]		Proposed SSE	
	12 bit	16 bit	12 bit	16 bit	12 bit	16 bit	12 bit	16 bit	12 bit	16 bit	12 bit	16 bit
NA	560	586	507	512	745	818	664	679	820	896	712	714
AR (%)	25	31.5	31.5	40	26.2	36.1	34.3	46.9	27.9	39.4	37.4	51.8
NFA	9072	11376	7892	8600	12218	16121	10385	12090	13776	17740	11298	13042
FAR (%)	24.9	29.8	37.8	54.2	26.2	33.2	41.2	58.2	28.1	35.6	46	62.1

TABLE II. AVERAGE REDUCTION OF ADDERS AND FULL ADDERS FOR DIFFERENT FILTER LENGTHS TO MEET THE PSR SPECIFICATIONS IN DESIGN EXAMPLE

Average reduction of adders (%)				Average reduction of full adders (%)			
12 bit		16 bit		12 bit		16 bit	
CSE [2]	Proposed SSE	CSE [2]	Proposed SSE	CSE [2]	Proposed SSE	CSE [2]	Proposed SSE
26.4	34.4	35.7	46.2	26.4	41.7	32.9	58.2

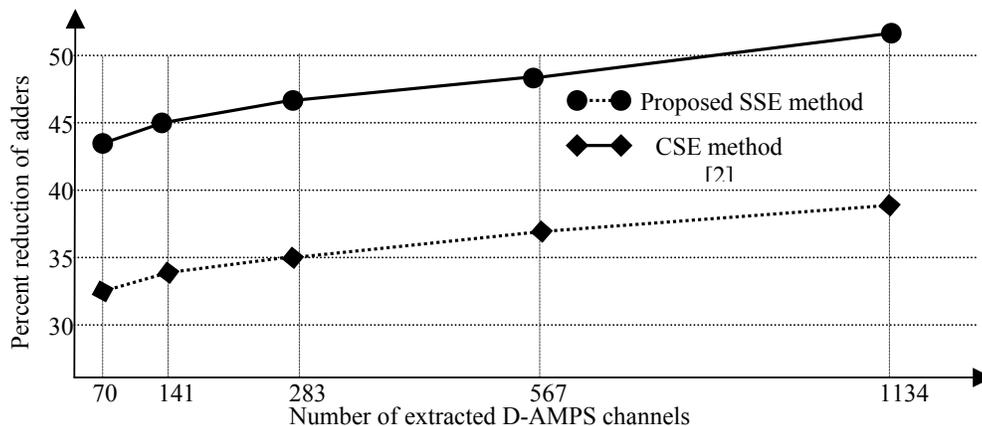


Figure 4. Reduction of adders to implement the D-AMPS channel filters in design example for different number of channels extracted

REFERENCES

- [1] J. Mitola, *Software Radio Architecture*. New York: Wiley, 2000.
- [2] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 677-688, Oct. 1996.
- [3] M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 3, pp. 196-203, March 2002.
- [4] Y. Jang and S. Yang, "Low-power CSD linear phase FIR filter structure using vertical common subexpression," *Electronics Letters*, vol. 38, no. 15, pp. 777-779, July 2002.
- [6] 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.
- [7] N. Spencer, "An overview of digital telephony standards," *IEE Colloquium on the Design of Digital Cellular Handsets*, pp. 1/1-1/7, March 1998.