



An improved common subexpression elimination method for reducing logic operators in FIR filter implementations without increasing logic depth

A.P. Vinod^{a,*}, Edmund Lai^b, Douglas L. Maskell^a, P.K. Meher^c

^a School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore

^b Institute of Information Science and Technology, Massey University, New Zealand

^c Institute for Infocomm Research, Singapore

ARTICLE INFO

Article history:

Received 23 May 2008

Received in revised form

14 July 2009

Accepted 14 July 2009

Keywords:

FIR filter

Coefficient multiplier

Common subexpression elimination

Logic operator

Logic depth

ABSTRACT

It is well known that common subexpression elimination techniques minimize the two main cost metrics namely logic operators and logic depths in realizing finite impulse response (FIR) filters. Two classes of common subexpressions occur in the canonic signed digit representation of filter coefficients, called the horizontal and the vertical subexpressions. Previous works have not addressed the trade-offs in using these two types of subexpressions on the logic depth and the number of logic operators of coefficient multipliers. In this paper, we analyze the impact of the horizontal and the vertical common subexpression elimination techniques on reducing the logic depth and number of logic operators in FIR filters. Further, we present an algorithm to optimize the common subexpression elimination that produces FIR filters with fewer numbers of logic operators when compared with other common subexpression elimination algorithms in literature. The design examples show that the average reduction of logic operators achieved using our method over the weight-2 horizontal common subexpression elimination method which produced the best trade-off between logic operators and logic depth (contention resolution algorithm, CRA-2 [F. Xu, C.-H. Chang, C.-C. Jong, Contention resolution algorithm for common subexpression elimination in digital filter design, IEEE Trans. Circuit Syst. II 52(10) (2005) 695–700 (October)]) is 15%. This reduction of logic operators is achieved without any increase in the logic depth. When compared with the recently proposed multiple adder graph (MAG) algorithm [Jeong-Ho Han, In-Cheol Park, FIR filter synthesis considering multiple adder graphs for a coefficient, IEEE Trans. Comput.-Aid. Design Integ. Circuit Syst. 27(5) (2008) 958–962 (May)], the average reduction of logic operators obtained using our method is 5% and the reduction of logic depth is 25%.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

FIR filters find extensive applications in mobile communication systems for the functions such as channelization, channel equalization, matched filtering and pulse shaping due to its absolute stability and linear phase property. The filters employed in mobile systems must be realized with low complexity and minimum delay. Although programmable filters based on digital signal processor cores are available, they are not very efficient as they consume more power and operate at low speed. Hence dedicated FIR filter architectures have received great deal of attention in the last decade.

The key computation in FIR filters is coefficient multiplications, which is implemented using shifts and adds, out of which the addition operation dominates the complexity because shifts are less complex and hence they can be hardwired. The number of adders (logic operators) used to compute the sum of the partial

product terms obtained when the inputs signal is multiplied by the coefficients and the critical path lengths (logic depths, which is equal to the number of adder-steps) of the multiplication operation are the two metrics that determine the complexity of FIR filters. Hence, the methods that minimize the complexity of multiplication in FIR filters focus on reducing the number of logic operators (LOs) and logic depth (LD) used to implement the multipliers. Multiple Constant Multiplications (MCM) is a transformation closely related to the widely used substitution of multiplications with constants by shifts and additions [1]. While the latter considers multiplication of only one constant at a time, the MCM considers multiplication of one variable with multiple constants. Common subexpression elimination (CSE) tackles the MCM problem by eliminating redundant computations in multiplier blocks (MBs) using the most common bit patterns called common subexpressions (CSs) that exist in the canonic signed digit (CSD) representation of coefficients [2–6]. In [2], an algorithm based on a coefficient subexpression graph for the identification and elimination of two-nonzero bit subexpressions (2-bit CSs) was proposed. A method to eliminate the most commonly occurring 2-bit CSs was proposed in [3]. As an

* Corresponding author. Tel.: +65 67906258; fax: +65 67926559.
E-mail address: asvinod@ntu.edu.sg (A.P. Vinod).

additional criterion in the subexpression identification process, an estimation of a latch count improvement was also used in [3]. A modification of the method in [2] for identifying and eliminating the best subexpressions to maximize the optimization impact is proposed in [4]. In [5], a nonrecursive signed CSE (NR-SCSE) algorithm has been proposed as a modification of the technique in [3] that minimizes the logic depth into the digital structure. The main idea in [6] is reordering computations and identifying common computations that maximize computation sharing between different multipliers. However the method in [6] offers only a slight improvement in reduction of adders (11%) over the CSE method [3]. Moreover, this method results in an increase in delay, corresponding to the delay of one adder-step on average. Instead of exploring optimizations over the original filter coefficients, differential coefficients were considered in [7,25], where differences between absolute values of filter coefficients were employed to reduce the dynamic range of computation. However the DCM suffers from overheads since it requires extra adders to compute the sum of the stored partial product of previous computation in order to compensate the effect of differential coefficients. In [8], the idea of using differential coefficient was applied to the multiplierless implementation of digital filters. In this work, a graph-based approach was developed to explore the low-complexity solutions for DCM. However complexity reduction achieved in [8] is usually smaller than the amount of reduction achieved by CSE approaches. A computation reduction technique called computation sharing differential coefficient (CSDC) method, which combines the strength of an augmented differential coefficient approach and subexpression sharing has been proposed in [9]. The augmented differential coefficient approach expands the design space by employing both differences and sums of filter coefficients through algorithmic equivalence. However the method in [9] has additional overheads since it requires extra adders to compensate the effect of differential coefficients if coefficient differences are used or extra subtractors if the sums of coefficients are used. A CSE algorithm that considers both the redundancy among the CSD coefficients and the LD in the MB was proposed in [10]. The reductions of LOs and LDs achieved using this method over the method in [4] is minimal. A contention resolution algorithm for weight-two horizontal subexpressions (CRA-2), based on an ingenious graph synthesis approach has been developed for the common subexpression elimination of the multiplication block of digital filter structures in [11]. CRA-2 saves 1–3% more logic operators than NR-SCSE [5]. In our recent work [12], we have proposed two techniques for optimizing the CSE methods. These techniques are based on the extension of conventional 2-bit CSs in [2–6] to form three-nonzero bit and four-nonzero bit super-subexpressions (SSs) by exploiting identical shifts between a 2-bit bit CS and a third nonzero bit, or between two 2-bit CSs. These SSs eliminate redundant computations of two-nonzero bit CSs and hence reduce the number of adders. However it must be noted that the formation of 3-bit and 4-bit SSs is based on the occurrence of 2-bit CSs with identical shifts between them. Therefore, the main limitation of the method in [12] is its dependence on the statistical distribution of shifts between the 2-bit CSs in the CSD representations of FIR filter coefficients. It has been shown in [12] that the number of SSs grows linearly with the wordlength and hence this technique is more advantageous only when the coefficient wordlength is relatively larger. Note that the routing complexity of the method in [12] is higher than that of the 2-bit CSE techniques in [2–6] as the former method has more number of subexpressions. However, using system-in-package (SiP) solutions which have higher integration capacity than conventional system-on-chip (SOC) solutions, the size and routing complexity can be significantly

reduced [22]. The first two limitations of [12] still pause hardware reduction constraints.

The Bull-Horrocks algorithm (BHA) [13] used a graph representation of the MB for reducing the number of LOs. Two methods that further reduce the number of LOs have been presented in [14], called the Bull-Horrocks Modified (BHM) algorithm and the n -dimensional Reduced Adder Graph (RAG n) algorithm. As the partial sums generated in multiplication are added in a serial manner in [13,14,24], these algorithms produce multipliers with large LDs, which increases the delay of the multiplier substantially. Even though the graph representation-based MB implementation reduces the number of LOs compared to CSE methods in [2–6], the LDs of the resulting coefficient multipliers are considerably larger. A new GD algorithm was proposed in [23] to optimize for minimum LOs for the MCM problem. The method in [23] resulted in longer LD, which in turn would increase the delay of the filter. Moreover, [23] is restricted to a maximum of 200 taps and its applicability for filters longer than 200 taps is not known (as per the details available on spiral.net). A multiple adder graph (MAG) based filter synthesis method has been recently proposed in [26]. While the previous graph-dependence algorithms [13,14,23,24] considered only one coefficient at a time and did not take into account the effect on the rest of the coefficients when synthesizing the coefficient, the MAG algorithm minimizes the adder cost by considering the effect on the remaining coefficients. A method for designing multiplier blocks with low LD was proposed in [27].

In general, the CSE methods utilize two types of CSs—the horizontal CSs (HCSs) that exist *within* each coefficient and the vertical CSs (VCSs) that exist *across* the adjacent coefficients. These techniques are called the horizontal common subexpression elimination (HCSE) and the vertical common subexpression elimination (VCSE), respectively. It has been shown in [15] that the VCSE offers better reduction of adders than the HCSE in realizing FIR filters. In our work [16], we have shown that the HCSs and the VCSs can be combined to produce better reduction of adders than the method in [15]. A new CSE method for implementing FIR filters using HCSs and VCSs has been proposed in [17]. The authors claim that the method in [17] reduces the average area by 6.4% and 3.8% over the methods in [8,9], respectively. The LD reductions achieved using [17] over [15,16] are 17.6% and 3.2%, respectively. However, the methods [15–17] only consider the implementation of the symmetric first half coefficient set of the FIR filter. These methods assume that the symmetric second half coefficient sets can be implemented by sharing the output of their symmetric first half coefficients. We denote the coefficients $h(0)$ to $h((N/2)-1)$ of an N -tap FIR filter as symmetric first half coefficients and $h(N/2)$ to $h(N-1)$ as symmetric second half coefficients. We noted that the use of VCSs imposes constraints in implementing the symmetric second half of the coefficients. Considerable numbers of additional LOs are needed to realize the symmetric second half as the coefficient symmetry cannot be completely exploited when VCSs are used in CSE. The LO requirements shown in [15–17] do not account this overhead and therefore the hardware reductions claimed using these methods are incorrect. To the best of our knowledge, the constraints in utilizing the symmetry of FIR filter coefficients while employing VCSs have not been addressed in literature. This is because all the CSE-based FIR filter implementation methods in literature discuss the implementation of symmetric first half coefficients only. These methods assumed that since FIR filter coefficients are symmetric, the symmetric second half part can be implemented from the first half coefficients without using any additional LOs. But this is not true when VCSs are used. Note that the constraints discussed in this paper are not applicable to anti-symmetric filters.

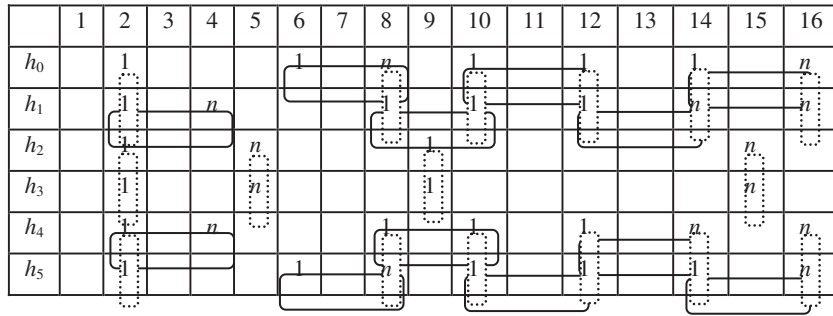


Fig. 1. HCSE (solid rectangles) and VCSE (dotted rectangles) in 6-tap FIR filter coefficients.

In this paper, we analyze the impact of HCSE and VCSE in exploiting the symmetry of FIR filter coefficients. Further, we present an optimization algorithm to reduce the number of LOs and LD in FIR filters. We show that our algorithm produces the best reduction of LOs when compared to the best known CSE algorithms in literature without increasing the LD of the coefficient multiplier. The rest of the paper is organized as follows. In Section 2, we present a complexity analysis of the filters realized using conventional CSE methods. Our CSE optimization technique is presented in Section 3. In Section 4, several design examples and comparisons are provided. Section 5 provides our conclusions.

2. Complexity analysis of CSE methods

A 6-tap FIR filter designed using Parks–McClellan algorithm is used to analyze the CSE methods. The passband and stopband edges of the filter are 0.2π and 0.25π , respectively. The 16-bit CSD representations of the coefficients are shown in Fig. 1. The numbers in the first row in Fig. 1 represent the number of bitwise right shifts and n represents -1 .

2.1. The HCSE algorithm

The HCSE uses the HCSs, $[101]$, $[10\bar{1}]$, $[1001]$ and $[100\bar{1}]$, and their negated versions present in the CSD representation of coefficients to eliminate redundant multiplications. Hartley [3] showed that the use of the two most commonly occurring HCSs, $[101]$ and $[10\bar{1}]$ would reduce the routing complexity of the filter circuit when compared with the HCSE using other HCSs such as $[1001]$ and $[100\bar{1}]$. Therefore, we use Hartley's HCSs $[101]$ and $[10\bar{1}]$ in our illustration. If x_1 is the input signal and 2^{-j} represents shift right by j , the HCSs, $[101]$ and $[10\bar{1}]$, shown inside the solid rectangles in Fig. 1 are given by x_2 and x_3 respectively:

$$x_2 = x_1 + 2^{-2}x_1 \text{ and } x_3 = x_1 - 2^{-2}x_1 \quad (1)$$

Fig. 2 shows the filter implementation using the HCSE method. The numerals adjacent to the data paths in Fig. 2 represent the number of bitwise right shifts. There are two types of adders in the filter structure—structural adders (SAs) that compute the sum of convolved signals (shown between each delay stage in Fig. 2), and MB adders (MBAs) which compute the sum of partial products formed in coefficient multiplication. For a given filter length, the number of SAs is fixed (equal to the number of distinct delay stages). The focus of CSE is to reduce the number of MBAs since they dominate the hardware cost. If N_b represents the number of nonzero bits in the symmetric half coefficient set of an

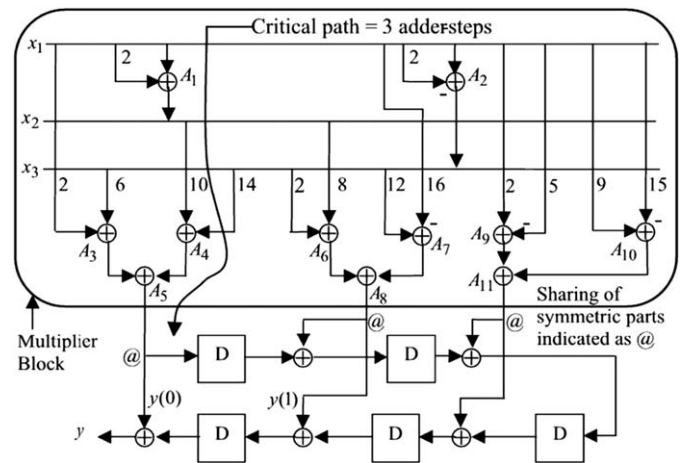


Fig. 2. FIR filter implementation using HCSE method.

FIR filter of length N , the total number of MBAs, T_{mba} , needed to realize the filter using direct method (direct method is the implementation using shifts and adds and without using CSE techniques) is

$$T_{mba} = N_b - \lceil N/2 \rceil \quad (2)$$

In the CSD coefficients in Fig. 1, N_b is 18 and N is 6. Thus 15 MBAs are required to realize the filter using direct method. In the HCSE method, since all the nonzero bits forming an HCS exist within the coefficient, its symmetric counterpart can be easily implemented using delays and SAs, i.e., no additional MBAs are required for the symmetric part. Note that the coefficients $h(3)$ – $h(5)$ are symmetric with respect to $h(0)$ – $h(2)$ and hence their outputs can be shared as shown in Fig. 2 using the symbol '@'. Thus, only 11 MBAs (A_1 – A_{11}) are needed for the HCSE implementation in Fig. 2, which is a reduction of 26% over the direct method. The LDs of the filter circuit are identical (3 addersteps) in direct method and CSE.

2.2. The VCSE algorithm

The VCSE methods [15–17] utilize the VCSs that occur across the adjacent coefficients to tackle the MCM. The VCSs, $[11]$ and $[1\bar{1}]$, that exist across the coefficients, shown inside the dotted rectangles in Fig. 1 by x_4 and x_5 , respectively:

$$x_4 = x_1 + x_1[-1] \text{ and } x_5 = x_1 - x_1[-1] \quad (3)$$

where $x_1[-k]$ represents x_1 delayed by k units. With these VCSs, the filter output using VCSE is

$$\begin{aligned}
 &2^{-2}x_4 + 2^{-6}x_1 - 2^{-8}x_5 + 2^{-10}x_4 + 2^{-12}x_4 + 2^{-14}x_5 - 2^{-16}x_4 \\
 &- 2^{-4}x_1[-1] + 2^{-2}x_4[-2] - 2^{-5}x_4[-2] + 2^{-9}x_4[-2] - 2^{-15}x_4[-2] \\
 &+ 2^{-2}x_4[-4] - 2^{-4}x_1[-4] + 2^{-8}x_5[-4] + 2^{-10}x_4[-4] + 2^{-2}x_4[-4] \\
 &- 2^{-4}x_1[-4] + 2^{-8}x_5[-4] + 2^{-10}x_4[-4] + 2^{-12}x_4[-4] \\
 &- 2^{-14}x_5[-4] - 2^{-16}x_4[-4] + 2^{-6}x_1[-5] \tag{4}
 \end{aligned}$$

Fig. 3 shows the VCSE realization of the filter. Since the bits that form VCSs occur across the coefficients, the symmetry of VCSs cannot be utilized when the bits are of opposite signs. Hence in VCSE, additional MBAs are required to obtain the symmetric part of the coefficients when more than one VCSs with bits of opposite signs exist.

Consider the VCSs across the coefficients $h(0)$ and $h(1)$ in Fig. 1:

$$\begin{aligned}
 &2^{-2}x_4 + 2^{-6}x_1 - 2^{-8}x_5 + 2^{-10}x_4 + 2^{-12}x_4 + 2^{-14}x_5 \\
 &- 2^{-16}x_4 - 2^{-4}x_1[-1] \tag{5}
 \end{aligned}$$

Its symmetric VCS part across the coefficients $h(4)$ and $h(5)$ is

$$\begin{aligned}
 &2^{-2}x_4[-4] - 2^{-4}x_1[-4] + 2^{-8}x_5[-4] + 2^{-10}x_4[-4] + 2^{-12}x_4[-4] \\
 &- 2^{-14}x_5[-4] - 2^{-16}x_4[-4] + 2^{-6}x_1[-5] \tag{6}
 \end{aligned}$$

Note that (6) cannot be directly obtained from (5) by simple delay operation since the signs and delays of certain terms of (6) are different from that of (5). Therefore, (6) needs to be obtained from (5) using (7) and (8) as given below:

$$\begin{aligned}
 &2^{-2}x_4 + 2^{-10}x_4 + 2^{-12}x_4 - 2^{-16}x_4 \xrightarrow{[4]} 2^{-2}x_4[-4] + 2^{-10}x_4[-4] \\
 &+ 2^{-12}x_4[-4] - 2^{-16}x_4[-4] \tag{7}
 \end{aligned}$$

$$\begin{aligned}
 &-2^{-8}x_5 + 2^{-14}x_5 \xrightarrow{[4]} -2^{-8}x_5[-4] - 2^{-14}x_5[-4] \tag{8}
 \end{aligned}$$

where ‘[4]’ represents 4 units delay and ‘-’ represents negation. The adders, A_3 , A_4 and A_5 compute (7) and A_6 computes (8) as

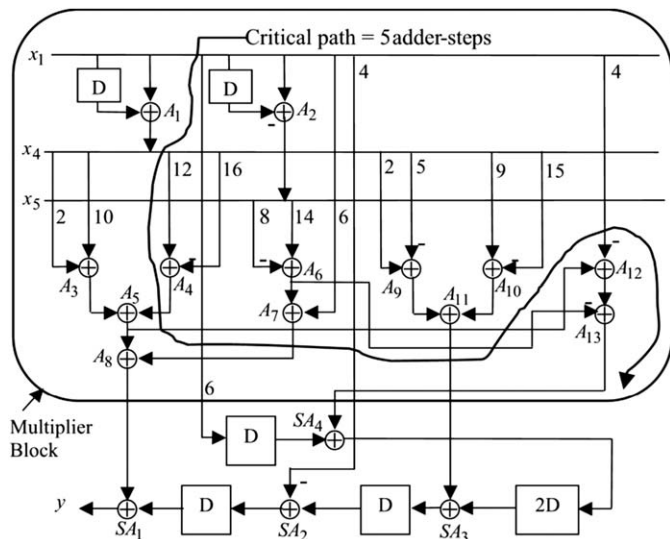


Fig. 3. FIR filter implementation using VCSE method.

shown in Fig. 3. The outputs of A_5 and A_6 corresponding to the left-hand side of (7) and (8) are utilized by A_{12} and A_{13} , respectively, to obtain the right-hand side of these expressions and hence extra adders are not required in this case. However, the term $2^{-6}x_1$ in (6) and $-2^{-4}x_1[-4]$ in (6) require two additional MBAs, A_7 and A_{12} . (But, the term $2^{-4}x_1[-1]$ in (5) does not require an MBA since no other terms that has an identical delay and same is the case with $2^{-6}x_1[-5]$ in (6). Thus these terms can be realized using SAs , SA_2 and SA_4 , respectively.) Due to this constraint in exploiting the symmetry, the VCSE implementation requires more MBAs (13 MBAs in this case) than the HCSE despite the fact that the number of VCSs (16 VCSs as in Fig. 1) is more than the number of HCSs (12 HCSs as in Fig. 1). Furthermore, the LD in VCSE implementation (5 adder-steps) is larger than the HCSE (3 adder-steps). Hence the VCSE method results in increased LOs and LDs when compared with HCSE. It must be noted that the CSE methods in [15–17] which employ VCSs do not account the overheads in LOs and LDs in implementing the symmetric second half coefficients. Therefore, the reductions claimed by these methods are incorrect.

We have examined the reduction of LOs (MBAs) for FIR filters of different lengths (N), 8 bits to 24 bits wordlengths and frequency response specifications (passband and the stopband frequencies, ω_p and ω_s , respectively). We noted that VCSE offered better reduction of LOs than the HCSE only when the coefficient wordlength is 8 bits. For wordlengths larger than 8 bits, the HCSE produced filters with fewer numbers of LOs than the VCSE. The LDs of the filters realized using VCSE are larger when compared with HCSE in most of the cases. In most practical filter applications, the frequency response of the filter will deteriorate considerably if the coefficients are coded using 8 bits. Therefore, the VCSE offers no advantage over the HCSE in practical FIR filter implementations if the proper VCSs are not chosen by carefully examining their signs. In next section, we present an optimization algorithm that efficiently combines HCSE and VCSE to minimize the number of LOs without increasing the LDs in FIR filters.

3. Proposed CSE optimization method

The core of our algorithm is to extract the maximum number of most frequently occurring common subexpressions. The HCSs, [101], [10 $\bar{1}$], [1001], [100 $\bar{1}$] and their negated versions, are used in our method since they are the most commonly occurring subexpressions. Among all the possible VCSs, we only use [11], [101] and their negated versions, since the signs of nonzero bits in these VCSs are identical (we designate these two VCSs as ‘compatible VCSs’). Therefore, the use of these compatible VCSs facilitates better utilization of coefficient symmetry. Note that other HCSs such as [10001] and [1000 $\bar{1}$] and VCSs such as [1001] and [100 $\bar{1}$] also exist in the CSD representation of coefficients. However, their frequency of occurrence is relatively smaller when compared to the HCSs and VCSs we have chosen. It has been shown in [3] that the use of large number of CSs with low frequency would have adverse effect on the routing complexity of the filter circuit.

Our algorithm first scans the coefficients to determine the frequency of HCSs and VCSs. For any coefficient, the CSs (HCSs or VCSs) with highest frequency are selected with priority given to HCSs first. If two or more HCSs occur common to different coefficients and if they are having identical shifts between them, then they are known as identical-shift HCSs (IS-HCSs). Each coefficient is compared with all the other coefficients for IS-HCSs. If more than one common IS-HCSs occur between a coefficient pair, the IS-HCSs can be grouped together to further eliminate redundant computations. Our optimization procedure is explained below.

3.1. CSE optimization procedure

The steps of our CSE optimization are as follows.

Step 1: Let C_{ij} represent the correlation index (CI) of the coefficient pair $h(i)$ and $h(j)$, and L is the number of filter taps.

Definition ((Correlation Index CI)). The correlation index of a coefficient pair is defined as the number of IS-HCSs obtained after HCSE algorithm. Thus, the CI of a coefficient pair is given by the number of identical shifts between the HCSs present in the coefficient pair.

Determine the CIs of all the coefficient pairs and form the correlation matrix, $C[h_{ij}]$ given by (9):

$$C[h_{ij}] = \begin{bmatrix} C_{01} & C_{02} & C_{03} & \dots & C_{0L} \\ & C_{12} & C_{13} & \dots & C_{1L} \\ & & C_{23} & C_{24} & C_{2L} \\ & & & \dots & \\ & & & & C_{L-1L} \end{bmatrix} \quad (9)$$

Step 2: The correlation matrix $C[h_{ij}]$ is scanned row wise and the coefficient pair corresponding to the largest CI is grouped together to extract the IS-HCS of each row. It may be noted that while selecting the best coefficient pairs, matching at one level must take into account how a particular match influences matching at the next level. This is done as follows. Set $i_0 = 1$ initially.

- (i) Compute the largest CI, $C_{i_0 j_m}^{\max}$ of the i_0 th row from $C_{ij} | i=i_0, j=i_0+1, j=L$, where j_m corresponds to the column in which the largest CI lies.
- (ii) Check all the CIs in the j_m column to find whether any other CI greater than $C_{i_0 j_m}^{\max}$ exists. If no such CI exists, choose $C_{i_0 j_m}^{\max}$ as the largest CI of the i_0 th row and group the corresponding pair $[h(i_0), h(j_m)]$. Otherwise, choose the second largest CI of the i_0 th row as the largest CI and obtain the IS-HCS from respective coefficient pair.

Step 3: Let the largest CI obtained in previous step be $C_{i_0 j_m}$. Replace all the elements of corresponding rows and columns by zero to exclude the coefficient pair chosen above from further search.

Step 4: If $i_0 \leq L$, set $i_0 = i_0 + 1$ and go to step 2. Thus all the IS-HCSs are determined and redundant computations are eliminated.

Step 5: Eliminate the compatible VCSs [11] and [101].

3.2. Illustrative example

Our method can be illustrated using the example in Fig. 4, in which the CSD form of the filter coefficients are shown. The HCSs [101], [10 $\bar{1}$] and [100 $\bar{1}$] and the VCSs [11] and [$\bar{1}\bar{1}$], are indicated inside rectangles in Fig. 4.

Substituting the HCSs in Fig. 4, $x_2 = [101] = 2$, $x_3 = [10n] = 3$ and $x_4 = [100n] = 4$, and the VCSs, $x_5 = [11] = 5$, and $-x_5 = [n] = -5$, we get Table 1.

The HCSs of 2 and 3 with a shift difference of 4 between them in h_0 and h_1 in Table 1 form the IS-HCS, $x_6 = [200003] = 6$ as shown in Table 2.

From Table 2, the expression for filter output y_k is

$$y_k = 2^{-1}x_6 + 2^{-10}x_3 + 2^{-14}x_5 - 2^{-1}x_5[-1] + 2^{-4}x_6[-1] + 2^{-3}x_4[-2] \quad (10)$$

The realization of (10) using our optimization is shown in Fig. 5.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
h_0	1	0	1	0	0	1	0	n	0	1	0	n	0	1
h_1	n	0	0	1	0	1	0	0	1	0	n	0	0	1
h_2	n	0	1	0	0	n	0	0	0	0	0	0	0	0

Fig. 4. HCSs and VCSs in CSD representation of filter coefficients.

Table 1

Representation of coefficients after extracting the HCSs and VCSs.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
h_0	2	0	0	0	0	3	0	0	0	3	0	0	0	5
h_1	-5	0	0	2	0	0	0	0	3	0	0	0	0	0
h_2	0	0	4	0	0	0	0	0	0	0	0	0	0	0

Table 2

Representation of coefficients after extracting the IS-HCSs.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
h_0	6	0	0	0	0	0	0	0	0	3	0	0	0	5
h_1	-5	0	0	6	0	0	0	0	0	0	0	0	0	0
h_2	0	0	4	0	0	0	0	0	0	0	0	0	0	0

The LD is 4 adder-steps and a total of 8 LOs are required for implementing the MB. For the direct implementation of the MB using the representation in Fig. 4, 13 LOs are required. Thus our CSE optimization offers 38% reduction of LOs compared to the direct implementation. The LD of the MB realized using our optimization method is one adder-step more than the direct implementation. In next section, we show that the LDs of filters realized using proposed method are comparable with the existing minimum LD CSE method.

4. Design examples

In this section, we present design examples of several FIR filters using proposed CSE optimization method. We also provide comparisons of the number of LOs and LDs needed to realize the filters using our method and the CSE methods in [3–6,9,11,14]. We use FIR filters designed using Parks–McClellan algorithm for different frequency response specifications (passband and stopband edges), filter lengths and coefficient wordlengths.

Example 1: In this example, we have compared the number of LOs and LDs generated by our algorithm with other algorithms for five benchmark filters FIR1 to FIR5. FIR1 and FIR2 are the example filters presented in [18]. FIR1 has a passband frequency of 0.15π and stopband frequency of 0.25π . For FIR2, the passband and stopband frequencies are 0.021π and 0.07π , respectively. FIR3 is the high pass filter L1 from [19]. FIR3 has a stopband frequency of 0.37π and passband frequency of 0.5π . FIR4 is a linear phase FIR filter employed in the filter bank channelizer of Digital Advanced Mobile Phone Systems (D-AMPS) receiver with passband and stopband frequencies of 0.6173π and 0.6276π , respectively. FIR5 is the filter employed in the receivers for the Personal Digital Cellular (PDC) receiver. The passband and stopband frequencies of FIR5 are 0.6836π and 0.6973π , respectively. The LOs and LDs obtained using these specifications for our method is compared with the BHM [14], NR-SCSE [5], Pasko [4] and Hartley [3]. Tables 3(A) and (B) show the comparison of the number of LOs and LDs. In Tables 3(A) and (B), N represents the filter length and W represents the coefficient wordlength. From Tables 3(A) and (B),

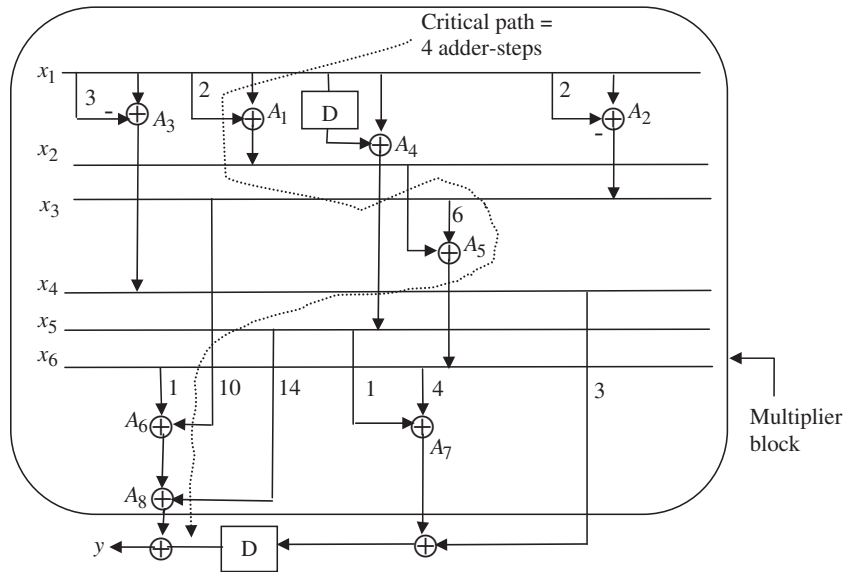


Fig. 5. FIR filter implementation using our CSE optimization.

Table 3 Comparison of LOs and LDs needed for realizing the benchmark FIR filters in Example 1.

(A)												
Filter	N	W	Direct method		NR-SCSE [5]		BHM [14]		Pasko [4]		Hartley [3]	
			LO	LD	LO	LD	LO	LD	LO	LD	LO	LD
FIR1	25	9	23	2	19	2	18	2	18	2	21	3
FIR2	59	14	86	2	55	3	55	5	60	3	70	4
FIR3	120	17	205	3	105	4	112	7	121	4	116	4
FIR4	200	13	224	3	150	3	152	6	154	4	171	3
FIR5	230	12	227	3	139	4	162	5	164	4	162	3
(B)												
Filter	N	W	C1 [27]		Multiple adder graph method [26]				Proposed method			
			LO	LD	LO		LD		LO	LD		
FIR1	25	9	18	2	19		2		18	2		
FIR2	59	14	55	4	54		4		54	3		
FIR3	120	17	100	5	96		5		90	4		
FIR4	200	13	136	4	131		4		128	3		
FIR5	230	12	140	4	128		5		118	4		

it is clear that our method produces the best reduction of LOs when compared to all other methods. The LDs achieved using our method is comparable with the CSE method in [5], which has the shortest LDs compared to other methods.

The graph-dependence-based BHM algorithm [14] produces the largest LDs since the partial sums generated in multiplication are added in a serial manner. Among the previous CSE methods, Hartley [3], Pasko [4] and NR-SCSE [5], the latter method [5] offers the best reduction both in terms of LOs and LDs. For the five benchmark filters FIR1 to FIR 5, our method offers an average LO reduction of 10.2% over the second best method, i.e., the NR-SCSE [5]. The average LO reductions achieved using our method over the Hartley [3], Pasko [4], BHM [14], C1 algorithm [27] and Multiple Adder Graph Method (MAG) [26] are 22.4%, 16.1%, 12.9%, 6.7% and 4.4%, respectively. The LDs of proposed filters are shorter

than other methods in most cases and in a few cases, they are comparable.

Example 2: In this example, our CSE optimization method is compared with CSE methods in [5,6,9,11,23] for the FIR filters with the passband and stopband frequencies of 0.2π and 0.22π , respectively. We have compared the LOs and LDs for different filter lengths of 20, 50, 80, 120, 200 and 400. The coefficient wordlengths considered are 12, 16, 20 and 24 bits. Tables 4(A)–(C) show the comparison of the LOs and Tables 5(A)–(C) show the LDs needed to implement the filters. Note that the results of [6] are not shown in the tables due to space constraints. However, the comparison with [6] is included in the figures. The results of [23] for filter length larger than 200 taps are indicated as ‘NA’ as [23] is restricted to a maximum of 200 taps (as per the details available on spiral.net).

Table 4
Comparison of the number of LOs needed for realizing the FIR filter in Example 2.

(A)												
Filter length (<i>N</i>)	NR-SCSE [5]				Proposed method				HCUB [23]			
	12 bit	16 bit	20bit	24 bit	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
20	26	30	37	48	24	28	34	46	13	17	24	29
50	41	60	75	96	37	55	66	84	20	37	45	65
80	60	86	111	146	52	67	92	120	33	59	80	119
120	85	131	175	220	76	106	150	184	34	72	110	159
200	125	197	260	328	102	150	198	260	38	100	164	225
400	172	324	453	604	150	270	368	494	NA	NA	NA	NA
(B)												
Filter length (<i>N</i>)	CSDC [9]				CRA-2 [11]							
	12 bit	16 bit	20bit	24 bit	12 bit	16 bit	20 bit	24 bit				
20	27	32	40	54	25	29	35	45				
50	43	64	77	99	40	59	74	96				
80	62	80	119	152	59	80	109	144				
120	89	138	190	232	84	128	169	214				
200	129	204	272	339	119	187	251	314				
400	176	338	466	620	167	314	450	598				
(C)												
Filter length (<i>N</i>)	Multiple adder graph method [26]				C1 [27]							
	12 bit	16 bit	20bit	24 bit	12 bit	16 bit	20 bit	24 bit				
20	26	29	36	50	27	30	39	55				
50	39	58	72	92	43	62	79	98				
80	59	71	97	128	64	75	101	136				
120	78	114	158	194	81	120	165	202				
200	110	158	207	270	118	165	214	279				
400	154	279	390	516	159	285	399	532				

Table 5
Comparison of the number of LDs needed for realizing the FIR filter in Example 2.

(A)												
Filter length (<i>N</i>)	NR-SCSE [5]				Proposed method				HCUB [23]			
	12 bit	16 bit	20bit	24 bit	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
20	3	4	4	5	3	4	4	5	6	6	7	8
50	2	4	4	5	3	4	4	5	6	6	7	8
80	3	3	4	5	3	3	4	5	6	6	7	8
120	3	3	4	5	3	3	4	5	7	7	8	9
200	3	4	5	5	3	4	5	5	8	8	8	9
400	3	4	4	5	3	4	4	5	NA	NA	NA	NA
(B)												
Filter length (<i>N</i>)	CSDC [9]				CRA-2 [11]							
	12 bit	16 bit	20bit	24 bit	12 bit	16 bit	20 bit	24 bit				
20	3	4	4	5	3	4	4	5				
50	2	4	4	5	2	4	4	5				
80	3	3	4	5	3	3	4	5				
120	3	3	4	5	3	3	4	5				
200	3	4	5	6	3	4	5	5				
400	3	4	4	5	3	4	4	5				
(C)												
Filter length (<i>N</i>)	Multiple adder graph method [26]				C1 [27]							
	12 bit	16 bit	20bit	24 bit	12 bit	16 bit	20 bit	24 bit				
20	3	5	6	7	3	5	5	5				
50	3	5	6	7	2	5	5	6				
80	3	4	6	8	3	3	5	6				
120	3	5	6	8	3	4	5	6				
200	3	6	7	8	3	4	5	6				
400	3	6	6	8	3	5	6	6				

Fig. 6 shows the reductions of LOs achieved using our CSE method over other CSE methods when the filter length is 80 for wordlengths of 12, 16, 20 and 24 bits.

Our method offers an average LO reduction of 11.3% over [6], 15.1% over CRA-2 [11], 15.6% over NR-SCSE [5] and 19.1% over CSDC [9]. The method in [23] produces around 35% reductions of LOs than our method, but the LDs of [23] are larger than our method by 50–70%. The LO reductions achieved using our CSE method over other CSE methods when the filter length is 200 for wordlengths of 12, 16, 20 and 24 bits are shown in Fig. 7. Our method offers an average LO reduction of 13.1% over [6], 15.4% over CRA-2 [11], 19.5% over NR-SCSE [5] and 23.3% over CSDC [9].

For the filters in Example 2 (filter lengths of 20, 50, 80, 120, 200 and 400), the average reductions of LOs achieved using our method over [6] is 9.7%, CRA-2 [11] is 12%, NR-SCSE [5] is 13%, CSDC [9] is 18.5%, MAG method [26] is 5.6% and C1 [27] is 10.3%. From Tables 5(A)–(C), the LDs of our method are same as that of NR-SCSE [5] and CRA-2 [11]. The LDs of filters realized using [6] is one adder-step more than our method. Our method also offers similar LD reduction over MAG [26] and C1 [27]. The proposed method achieves 12% reduction of LOs compared to the best known minimum LOs method (CRA-2 [11]) for the same LD. When compared to [23], our method needs an average of 25% additional LOs, but our methods reduces the LDs by 50%. Moreover, [23] is

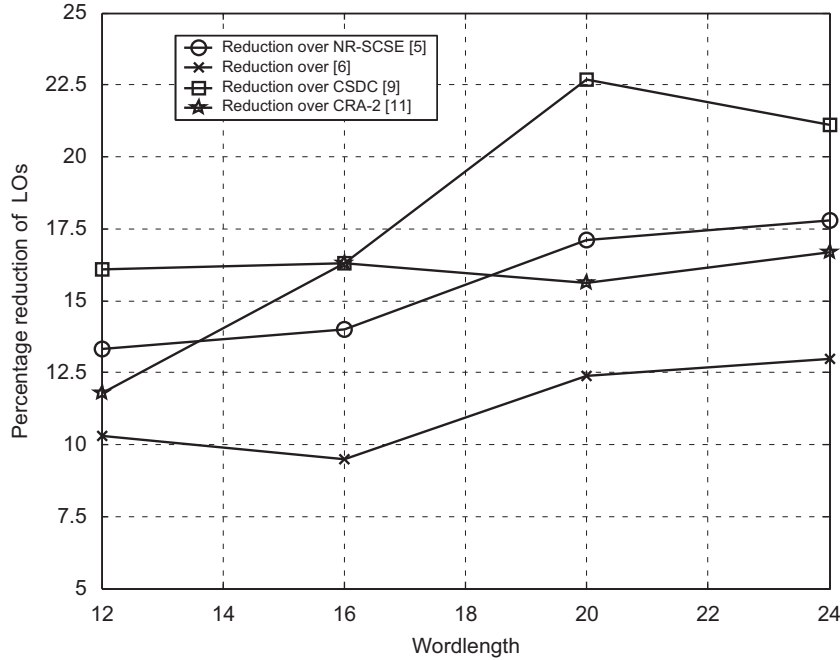


Fig. 6. Percentage reduction of LOs achieved using our method over the methods in [5,6,9,11] for the 80-tap filter in Example 2.

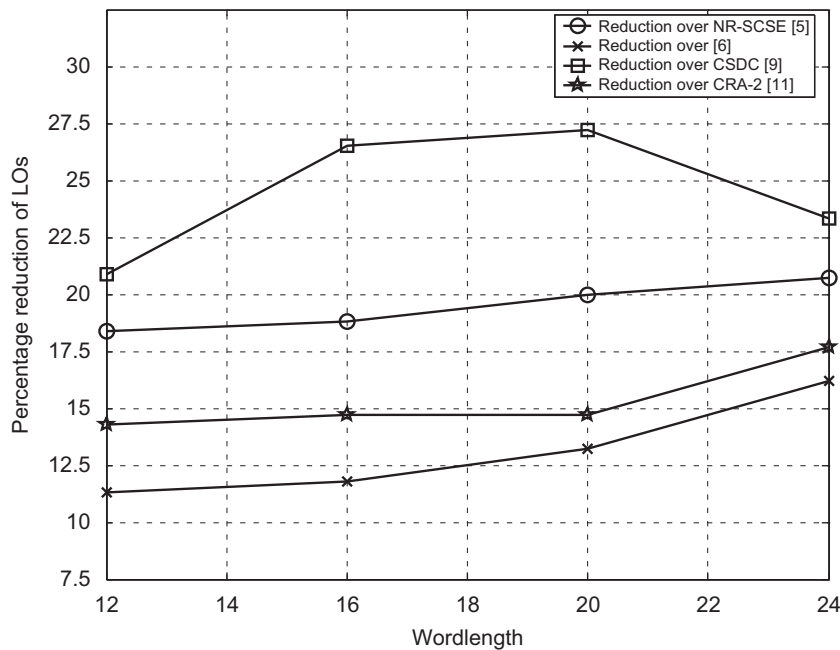


Fig. 7. Percentage reduction of LOs achieved using our method over the methods in [5,6,9,11] for the 200-tap filter in Example 2.

restricted to a maximum of 200 taps (as per the details available on spiral.net) whereas our method has no such filter length restrictions.

Fig. 8 shows the LO vs. LD characteristics of the 6 FIR filters in Example 2. It can be noted that the proposed method offers the best trade-off between LO and LD. The C1 algorithm [27] provides the second best trade-off up to LO = 160 (LD = 4), but its LD increases when LO increases further. The MAG method [26] needs only slightly more number of LOs than proposed method, but its LD values are very high.

Example 3: In this example, we consider FIR filters employed as channel filters in the channelizer of a wireless communication receiver. The channel filters of a receiver need to extract multiple narrowband signals (communication channels) from a wideband input signal. These filters must have a large number of taps due to

the stringent adjacent channel attenuation specifications of wireless communications standards. We present examples of implementing channel filters using our method and provide comparisons with CSE techniques [5,6,9,11]. The channel filters employed in the filter bank channelizer of digital advanced mobile phone systems (D-AMPS) in [20] are considered. The sampling rate chosen is 34.02 MHz as in [20]. The channel filters extract 30 kHz D-AMPS channels from the input signal after down-sampling by a factor of 350. The passband and stopband edges are 30 and 30.5 kHz, respectively. The peak passband ripple is chosen as 0.1 dB. The filter stopband specifications are chosen as in the D-AMPS standard [20]. The length of the FIR filter N is determined using (11) [21]:

$$N = \frac{-10 \log_{10} \hat{\sigma}_1 \hat{\sigma}_2 - 13}{14.6 \Delta f} + 1 \quad (11)$$

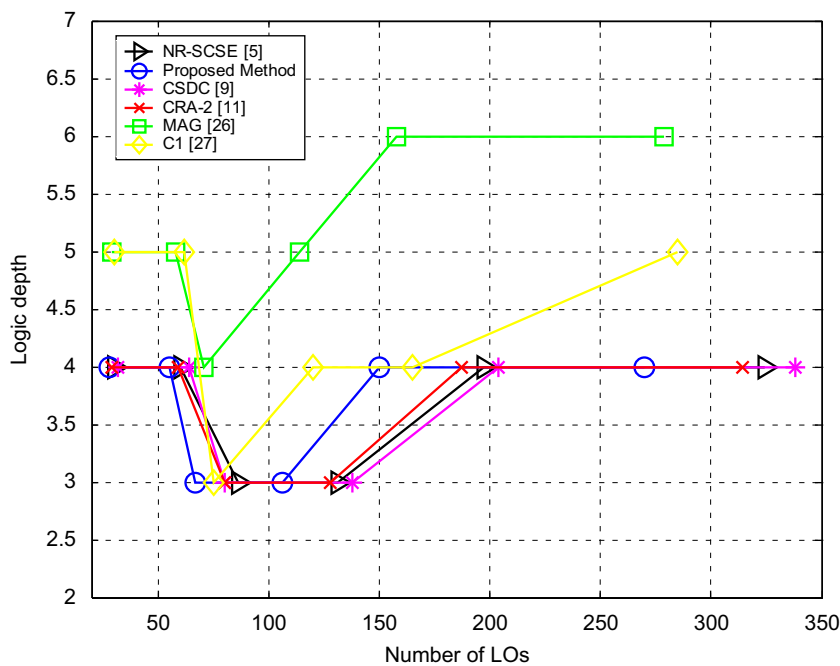


Fig. 8. LO vs. LD characteristic of the 6 FIR filters in Example 2.

Table 6

Comparison of the number of LOs needed for realizing the channel filters in Example 3.

(A) PSR (dB)	Filter length (N)	NR-SCSE [5]			Proposed method			[6]		
		16 bit	20 bit	24 bit	16 bit	20 bit	24 bit	16 bit	20 bit	24 bit
		-24	200	201	272	346	176	229	290	190
-48	460	389	542	701	316	430	549	370	520	668
-65	610	462	680	872	376	538	706	447	660	850
-85	940	596	917	1224	481	720	980	576	890	1184
-96	1180	661	1067	1442	536	856	1170	620	970	1320

(B) PSR (dB)	Filter length (N)	CSDC [9]			CRA-2 [11]		
		16 bit	20 bit	24 bit	16 bit	20 bit	24 bit
		-24	200	210	290	360	198
-48	460	398	570	720	370	525	688
-65	610	480	704	898	450	671	859
-85	940	610	940	1340	580	896	1180
-96	1180	670	1147	1520	649	1002	1340

where σ_1 and σ_2 are the passband and stopband ripples, respectively, and Δf is the normalized width of the transition band. The comparison of LOs needed to implement the filters is shown in Tables 6(A) and (B). Filters of lengths 200, 460, 610, 940 and 1180 are chosen corresponding to peak stopband ripple (PSR) specifications of -24 , -48 , -65 , -85 and -96 dB, respectively.

Fig. 9 shows the reductions of LOs achieved using our CSE method over other CSE methods when the filter length is 460 for wordlengths of 16, 20 and 24 bits. Our method offers an average LO reduction of 16.6% over [6], 17.6% over CRA-2 [11], 20.4% over NR-SCSE [5] and 22.9% over CSDC [9]. The LO reductions achieved

using our CSE method over other CSE methods when the filter length is 940 for wordlengths of 16, 20 and 24 bits are shown in Fig. 10.

Our method offers an average LO reduction of 17.6% over [6], 17.9% over CRA-2 [11], 20.2% over NR-SCSE [5] and 26.9% over CSDC [9]. As in the case of previous example, the LDs of filters realized using [6] is one adder-step more than our method. The LDs of filters realized using our method are same as that of [5,11]. We also compared with [26,27] and found that our method offers average LO reductions of 10% and 12.8% over [26,27], respectively. The LD reductions obtained using our algorithm over [26,27] were 25% and 15%, respectively. The detailed results are omitted here for brevity.

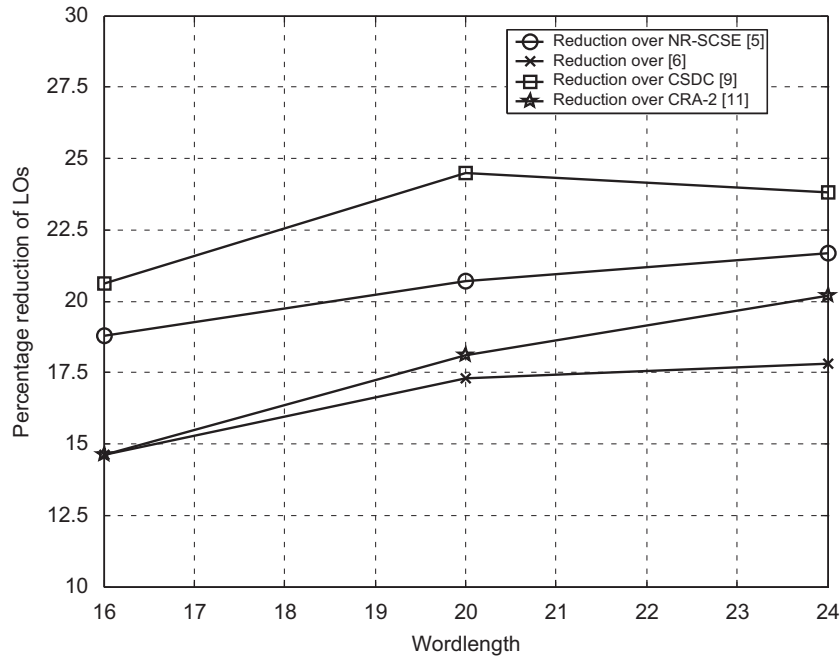


Fig. 9. Percentage reduction of LOs achieved using our method over the methods in [5,6,9,11] for the 460-tap filter in Example 3.

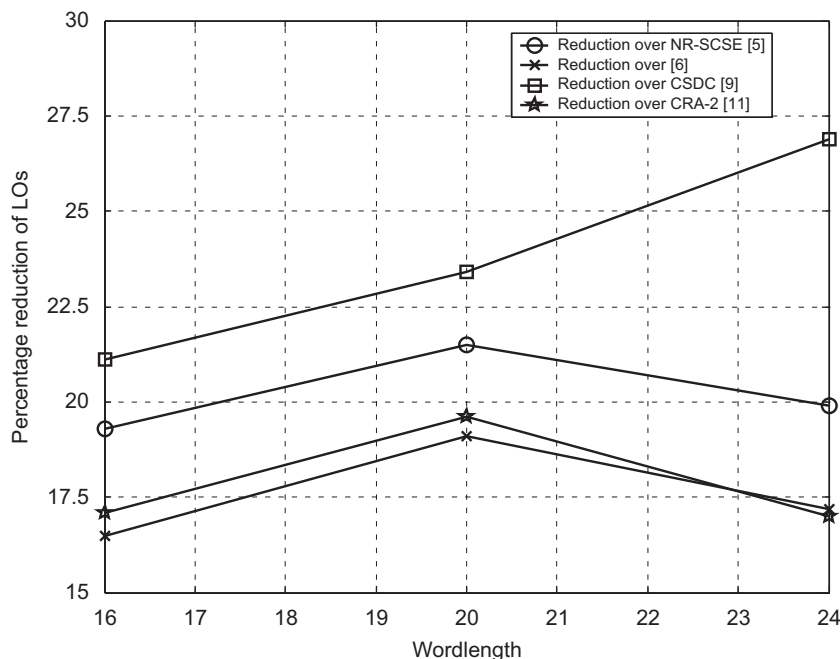


Fig. 10. Percentage reduction of LOs achieved using our method over the methods in [5,6,9,11] for the 940-tap filter in Example 3.

Our design examples show that in terms of number of LOs, proposed CSE method offers an average reduction of 13.7% over the best known minimum LO method [6]. It must be noted that the method in [6] requires one adder-step more than proposed method. The proposed method offers average LO reduction of 15% over the best known minimum LD method, CRA-2 [11]. The LDs of proposed method are similar to that of [11]. Thus our method offers the best tradeoff in terms of number of LOs and LDs when compared to other CSE methods in literature.

Example 4: In this example, our method is compared with the recently proposed MAG filter synthesis method [26]. We used the same filter specifications as that of filters 1–6 in [26] for comparison. The lengths of filters 1–6 are 60, 100, 101, 101, 60 and 60, respectively. All the filters have coefficient wordlength of 16 bits. The passband and stopband edge frequencies (F_{p1} and F_{s1}) of filters 1 and 2 (both lowpass filters) are $\{0.1\pi, 0.14\pi\}$ and $\{0.2\pi, 0.6\pi\}$, respectively. The stopband and passband edge frequencies (F_{s1} and F_{p1}) of filters 3 and 4 (both highpass filters) are $\{0.3\pi, 0.42\pi\}$ and $\{0.3\pi, 0.76\pi\}$, respectively. For the bandpass filter 5, edge frequencies are $F_{s1} = 0.2\pi$, $F_{p1} = 0.3\pi$, $F_{p2} = 0.8\pi$ and $F_{s2} = 0.9\pi$. The bandpass filter 6 has edge frequencies are $F_{s1} = 0.2\pi$, $F_{p1} = 0.45\pi$, $F_{p2} = 0.65\pi$ and $F_{s2} = 0.9\pi$. Table 7 shows the LOs and LDs for filters 1–6 realized using our method and that using [26]. The LO (adder cost) and LD (adder-step) values of filters realized using the multiple adder graph method are directly taken from Han and Park [26]. Our method offers average LO reduction of 1.8% and average LD reduction of 32.3% over [26].

Example 5: We present a comparison of our method with the C1 algorithm in [27]. The FIR filter specification chosen in this example is exactly same as that of the design example in [27]. Filter order is 24 and normalized passband and stopband edge frequencies are 0.25 and 0.3, respectively. Floored 12-bit quantized coefficients are taken as in [27]. Table 8 shows the comparison of LOs and LDs for Example 5 for RAGn and BHM, applied once and twice, C1 [27] and proposed method. LO and LD values are directly taken from Table 1 in [27]. It can be noted that proposed method results in least LO and LD.

Table 7
Comparison of LOs and LDs of filters in Example 4.

Filter	Multiple adder graph method [26]		Proposed method	
	LO	LD	LO	LD
1	37	6	36	4
2	54	9	52	4
3	37	6	38	5
4	49	6	48	4
5	29	5	30	4
6	25	5	23	4
Average	38.5	6.2	37.8	4.2

Table 8
Comparison of LOs and LDs of FIR filter in Example 5.

Algorithm	LO	LD
RAG-n	18	9
BHM	20	5 (5)
RAG-n \times 2	18	9
BHM \times 2	20	5 (5)
C1	19	4
Proposed method	18	3

5. Conclusions

We have compared the reduction of logic operators (adders) and logic depths (critical path lengths) achieved using the horizontal and the vertical common subexpressions in realizing FIR filters. It has been noted that the common subexpression elimination technique employing horizontal common subexpressions offer better reductions in the number of logic operators as well as logic depths than their vertical common subexpressions counterpart in FIR filter implementations. Further, we have presented a method to optimize the horizontal and vertical common subexpression elimination techniques. Our method produced FIR filters with fewer numbers of logic operators and shorter logic depths when compared with other common subexpression elimination algorithms in literature. Our CSE optimization method offered an average reduction of 15% in terms of the number of logic operators over the best known common weight-two horizontal subexpression elimination method without any increase in logic depth. Our method reduces the number of structural adders in some cases at the cost of a slight increase in the number of delay elements. When compared with the recently proposed multiple adder graph (MAG) algorithm [26], the average reduction of logic operators obtained using our method is 5% and the reduction of logic depth is 25%.

References

- [1] M. Potkonjak, M.B. Srivastava, A.P. Chandrakasan, Multiple constant multiplications: efficient and versatile framework and algorithms for exploring common subexpression elimination, *IEEE Trans. CAD* 15 (2) (1996) 151–165 (February).
- [2] M. Mehendale, S.D. Sherlekar, 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*, IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 668–671.
- [3] R.I. Hartley, Subexpression sharing in filters using canonic signed digit multipliers, *IEEE Trans. Circuits Syst.* II 43 (1996) 677–688 (October).
- [4] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, D. Durackova, A new algorithm for elimination of common subexpressions, *IEEE Trans. Comput.-Aid. Design Integ. Circuit Syst.* 18 (1) (1999) 58–68 (January).
- [5] M.M. Peiro, E.I. Boemo, L. Wanhammar, Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm, *IEEE Trans. Circuit Syst. II* 49 (3) (2002) 196–203 (March).
- [6] H. Choo, K. Muhammad, K. Roy, Complexity reduction of digital filters using shift inclusive differential coefficients, *IEEE Trans. Signal Process.* 52 (6) (2004) 1760–1772 (June).
- [7] N. Sankarayya, K. Roy, D. Bhattacharya, Algorithms for low power and high speed FIR filter realization using differential coefficients, *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.* 44 (6) (1997) 488–497 (June).
- [8] K. Muhammad, K. Roy, A graph theoretic approach for synthesizing very low-complexity high-speed digital filters, *IEEE Trans. Comput.-Aid. Design Integ. Circuit* 21 (2) (2002) 204–216 (February).
- [9] Y. Wang, K. Roy, CSDC: a new complexity reduction technique for multiplierless implementation of digital FIR filters, *IEEE Trans. Circuits Syst. I* 52 (9) (2005) 1845–1853 (September).
- [10] C.-Y. Yao, H.-H. Chen, T.-F. Lin, C.-J. Chien, C.-T. Hsu, A novel common subexpression elimination method for synthesizing fixed-point FIR filters, *IEEE Trans. Circuits Syst. I* 51 (11) (2004) 2215–2221 (November).
- [11] F. Xu, C.-H. Chang, C.-C. Jong, Contention resolution algorithm for common subexpression elimination in digital filter design, *IEEE Trans. Circuits Syst. II* 52 (10) (2005) 695–700 (October).
- [12] A.P. Vinod, E.M.-K. Lai, On the implementation of efficient channel filters for wideband receivers by optimizing common subexpression elimination methods, *IEEE Trans. Comput.-Aid. Design Integ. Circuit Syst.* 24 (2) (2005) 295–304 (February).
- [13] D.R. Bull, D.H. Horrocks, Realization techniques for primitive operator infinite impulse response digital filters, *Proc. Int. Symp. Circuit Syst.*, vol. 1, , 1993, pp. 607–610 (May).
- [14] A.G. Dempster, M.D. Mcleod, Use of minimum adder multiplier blocks in FIR digital filters, *IEEE Trans. Circuit Syst. II* 42 (1995) 569–577 (September).
- [15] Y. Jang, S. Yang, Low-power CSD linear phase FIR filter structure using vertical common sub-expression, *Electron. Lett.* 38 (15) (2002) 777–779 (July 2002).
- [16] A.P. Vinod, E.M.-K. Lai, A.B. Premkumar, C.T. Lau, FIR filter implementation by efficient sharing of horizontal and vertical common subexpressions, *Electron. Lett.* 39 (2) (2003) 251–253 (January).
- [17] Y. Takahashi, M. Yokoyama, New cost-effective VLSI implementation of multiplierless FIR filter using common subexpression elimination, in:

Proceedings of International Symposium on Circuits and Systems, vol. 2, Kobe, Japan, May 2005, pp. 1445–1448.

- [18] H. Samuelli, An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients, *IEEE Trans. Circuits Syst.* 36 (1989) 1044–1057 (July).
- [19] Y.C. Lim, S.R. Parker, Discrete coefficient fir digital filter design based upon an LMS criteria, *IEEE Trans. Circuit Syst. CAS-30* (10) (1983) 723–739 (October).
- [20] K.C. Zangi, R.D. Koilpillai, Software radio issues in cellular base stations, *IEEE J. Select. Area Commun.* 17 (4) (1999) 561–573 (April).
- [21] J.G. Prokias, D.G. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [22] C. Trigas, Design challenges for system-in-package vs. system-on-chip, *Proc. IEEE Custom Integ. Circuit Conf.* 1 (2003) 663–666 (September).
- [23] Y. Voronenko, M. Pushcel, Multiplierless Multiple Constant Multiplication, *ACM Trans. Algorithms* 3 (2) (2007) Article no. 11.
- [24] F. Xu, C.H. Chang, C.C. Jong, Modified reduced adder graph algorithm for multiplierless FIR filters, *IEE Electron. Lett.* 41 (6) (2005) 302–303 (March).
- [25] A.P. Vinod, Ankita Singla, C.H. Chang, Low power differential coefficients-based FIR filters using hardware optimized multipliers, *IET Circuit Device Syst.* 1 (1) (2007) 13–20 (February).
- [26] Jeong-Ho Han, In-Cheol Park, FIR filter synthesis considering multiple adder graphs for a coefficient, *IEEE Trans. Comput.-Aid. Design Integrat. Circuit Syst.* 27 (5) (2008) 958–962 (May 2008).
- [27] A.G. Dempster, S.S. Dimirsoy, I. Kale, Designing multiplier blocks with low logic depth, *Proc. IEEE Int. Symp. Circuit Syst.* 5 (2002) 773–776 (Phoenix, USA, May).



A.P. Vinod received his B. Tech degree in Instrumentation and Control Engineering from University of Calicut, India in 1994 and the M. Engg and Ph.D. degrees in Computer Engineering from Nanyang Technological University, Singapore in 2000 and 2004, respectively. He has spent the first 5 years (November 1993–October 1998) of his career in industry as an automation engineer at Kirloskar, Bangalore, India, Tata Honeywell, Pune, India, and Shell Singapore. From September 2000 to September 2002, he was a lecturer in the School of Electrical and Electronic Engineering at Singapore Polytechnic, Singapore. He was a lecturer in the School of Computer Engineering at Nanyang Technological

University (NTU), Singapore, from September 2002 to November 2004, and since December 2004, he has been an Assistant Professor in NTU. His research interests include digital signal processing, low power and reconfigurable DSP circuits, software radio, cognitive radio and brain-computer interface.



Edmund M-K. Lai received the B.E. (Hons) and Ph.D. degrees in 1982 and 1991 respectively from the University of Western Australia, both in Electrical Engineering. He is currently a faculty member of the School of Engineering and Advanced Technology, Massey University at Wellington, New Zealand. Previously he has been a faculty member of the Department of Electrical and Electronic Engineering, The University of Western Australia from 1985 to 1990, the Department of Information Engineering, the Chinese University of Hong Kong from 1990 to 1995, Edith Cowan University in Perth from 1995 to 1998 and the School of Computer Engineering, Nanyang Technological

University in Singapore from 1999 to 2006. His current research interests include cognitive radio, compressed sensing, digital signal processing, information theory, artificial neural networks.



Douglas L. Maskell received the B.E (Hons.), M.Eng.Sc., and Ph.D. degrees in Electrical and Computer Engineering from James Cook University, Townsville, Australia, in 1980, 1985, and 1996, respectively. He is currently an Associate Professor with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. He is also the Leader of the Reconfigurable Computing Group, Centre for High Performance Embedded Systems (CHiPES), NTU. His current research interests include dynamic (runtime) reconfigurable computing, including efficient utilization of FPGA hardware and architecture resources for near routeless placement and fast configuration. He also conducts

research in a number of embedded systems application areas, including biomedical algorithm acceleration using FPGA, embedded applications and architectures in computational cognitive science, low-complexity digital filters, and low-complexity phase and distance measurement.



Pramod Kumar Meher received the B.Sc. (Honours) and M.Sc. degrees in Physics and the Ph.D. in science from Sambalpur University, Sambalpur, India, in 1976, 1978, and 1996, respectively. Currently, he is a Senior Scientist with the Institute for Infocomm Research, Singapore. Prior to this assignment he was a visiting faculty with the School of Computer Engineering, Nanyang Technological University, Singapore. He was a Professor of Computer Applications with Utkal University, Bhubaneswar, India, from 1997 to 2002, a Reader in Electronics with Berhampur University, Berhampur, India, from 1993 to 1997, and a Lecturer in physics with various Government Colleges in India

from 1981 to 1993. His research interest includes design of dedicated and reconfigurable architectures for computation-intensive algorithms pertaining to signal processing, image processing, communication, bio-informatics and intelligent computing. He has published more than 100 technical papers in various reputed journals and conference proceedings. Dr. Meher is a Fellow of the Institution of Electronics and Telecommunication Engineers (IETE), India and a Fellow of the Institution of Engineering and Technology (IET), UK. He is currently serving as Associate Editor for the *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-II: EXPRESS BRIEFS*, *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, and *Journal of Circuits, Systems, and Signal Processing*.