

COMPARISON OF THE HORIZONTAL AND THE VERTICAL COMMON SUBEXPRESSION ELIMINATION METHODS FOR REALIZING DIGITAL FILTERS

A.P.Vinod and E.M-K.Lai

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

ABSTRACT

The Common Subexpression Elimination (CSE) techniques address the issue of minimizing the number of adders needed to implement the coefficient multipliers in digital filters. Two classes of Common Subexpressions (CS) occur in the Canonic Signed Digit (CSD) representation of coefficients, called the horizontal and the vertical CS. Previous works have not addressed the trade-offs in using these two types of CS on the delay and the number of adders of multiplier blocks. In this paper, we provide a comparison of hardware reductions achieved using the horizontal and the vertical CS in realizing digital filters. We show that the CSE technique employing horizontal CS offer better reductions in the number of adders as well as critical paths than their vertical CS counterpart in practical Linear Phase Finite Impulse Response (LPFIR) filter implementations. Our simulation results show that the hardware reductions offered by the vertical CS are improved in implementing infinite impulse response (IIR) filters than their LPFIR counterparts.

1. INTRODUCTION

The number of adders (subtractors) used to implement the coefficient multiplications determines the complexity of LPFIR filters. Hence, the algorithms that minimize the complexity of multiplication in LPFIR filters focus on reducing the number of adders (subtractors) used to implement the multipliers. Multiple Constant Multiplication (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.

The CSE tackles the MCM problem by minimizing the number of additions through extracting common parts among the constants represented in CSD [2]-[7]. The goal of CSE is to identify multiple occurrences of identical bit patterns that are present in the coefficient set. Since the computation of multiple identical expressions needs to be implemented only once, the resources necessary for these operations can be shared. In [2], an algorithm based on a coefficient subexpression graph for the identification and elimination of two nonzero bit subexpressions was proposed. A method to eliminate the most commonly occurring 2-bit subexpressions was proposed in [3]. As an additional criterion in the subexpression identification process,

an estimation of a latch count improvement was also used in [3]. A modification of the 2-bit CS optimization technique presented in [2] for identifying the “proper” patterns for elimination to maximize the optimization impact is proposed in [4]. In [5], a nonrecursive signed CSE algorithm has been proposed as a modification of the technique in [3] that minimizes the logic depth into the digital structure. In general, the methods proposed in [2]-[5] utilize two types of CS – the horizontal CS (HCS) that occurs *within* each coefficient and the vertical CS (VCS) that occurs *across* the adjacent coefficients. These techniques are called the Horizontal Common Subexpression Elimination (HCSE) and the Vertical Common Subexpression Elimination (VCSE) respectively. Recently, it has been shown in [6] that the VCSE technique offers better reduction of adders than the HCSE technique in realizing LPFIR filters. In their work [6], the authors exploit the fact that many VCS exist in LPFIR filters since their adjacent coefficients have similar patterns in the most significant bits. However, the constraints in utilizing the symmetry of the coefficients in LPFIR filters have not been properly addressed in the VCSE method [6]. Furthermore, the VCSE method does not consider the critical path of the multiplier, which determines the delay. In this paper, we analyze the impact of the symmetry of LPFIR filter coefficients on hardware (number of adders) and critical path reductions in HCSE and VCSE implementations.

The paper is organized as follows. In section 2, we provide a brief review of the CSE approach and illustrate the HCSE technique. The VCSE technique and its comparison with the HCSE are presented in section 3. In section 4, we illustrate the realization of digital filters using these techniques. Section 5 provides our conclusions.

2. THE HCSE TECHNIQUE

A 6-tap LPFIR filter designed using Parks-McClellan algorithm is used to illustrate the CSE methods. The pass-band and stop-band edges of the filter are 0.2π and 0.25π respectively. The 16-bit CSD form of the coefficients is shown in Fig. 1. The numbers in the first row represent the number of bitwise right shifts. The HCSE technique utilizes the most common horizontal subexpressions that occur within each coefficient to eliminate redundant computations. In general, these methods use Hartley’s [3] two most common HCS, i.e., $[1\ 0\ 1]$ and $[1\ 0\ -1]$ and their

negated versions. If x_1 is the input signal and 2^{-j} represents shift right by j , the HCS ([1 0 1] and [1 0 -1]) shown inside the solid rectangles in Fig. 1 are given by:

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

Using HCSE, the output of the filter can be represented as

$$\begin{aligned} & 2^{-2}x_1 + 2^{-6}x_3 + 2^{-10}x_2 + 2^{-14}x_3 + 2^{-2}x_3[-1] + 2^{-8}x_2[-1] + \\ & 2^{-12}x_3[-1] - 2^{-16}x_1[-1] + 2^{-2}x_1[-2] - 2^{-5}x_1[-2] + 2^{-9}x_1[-2] \\ & - 2^{-15}x_1[-2] + 2^{-2}x_1[-3] - 2^{-5}x_1[-3] + 2^{-9}x_1[-3] - 2^{-15}x_1[-3] \\ & + 2^{-2}x_3[-4] \end{aligned} \quad (2)$$

where $[-k]$ represents a delay of k . 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. The function of the Multiplier Block (MB) shown in Fig. 2 is to compute the sum of partial products obtained when the input signal (x_1) is convolved with the filter coefficients (h_k).

Definition 1 (Multiplier block adders): The adders used in the MB to compute the sum of partial products formed when x_1 is multiplied with h_k are called Multiplier Block Adders (MBA).

Definition 2 (Structural adders): The inter-tap adders used to compute the sum of convolved signals (shown between each delay stage in Fig. 2) are called Structural Adders (SA). The number of structural adders in a filter structure is same as that of the number of distinct delay stages.

Definition 3 (Critical path): The number of addition stages (adder-steps) in a maximal path of decomposed multiplications is called the critical path of the MB. In a tree-structured multiplier that performs parallel addition, the critical path is the height of the tree.

The focus of multiplier block reduction methods is to reduce the number of MBA's since they dominate the hardware cost. If N_b represents the number of nonzero bits in the symmetric half coefficient set of an FIR filter of length N , the total number of MBA's, T_{mba} , needed to realize the filter using direct method (i.e., without using CSE techniques) is

$$T_{mba} = N_b - \lfloor N/2 \rfloor \quad (3)$$

In this case, N_b is 18 and N is 6. Therefore, fifteen MBA's 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 counter-part can be easily implemented using delays and SA's, i.e., no additional MBA's 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 eleven MBA's are needed for the HCSE implementation as shown in Fig. 2. Note that critical path of the MB is three adder-steps.

3. THE VCSE TECHNIQUE

The VCSE technique utilizes the subexpressions (VCS) that occur across the adjacent coefficients to tackle the MCM. The

VCS, [1 1] and [1 -1], that exist across the coefficients, shown inside the dotted rectangles in Fig. 1 are given by:

$$x_4 = x_1 + x_1[-1] \quad \text{and} \quad x_5 = x_1 - x_1[-1] \quad (4)$$

With these VCS, the filter output in VCSE implementation can be represented as

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

Fig. 3 shows the realization of the filter using the VCSE technique. Note that thirteen MBA's are required for VCSE implementation of the filter shown in Fig. 3. Since the bits that form VCS in VCSE method occur across the coefficients, the symmetry of VCS cannot be exploited when the bits are of opposite sign. Hence in VCSE implementations, additional MBA's are required to obtain the symmetric part of the coefficients when more than one VCS with bits of opposite signs exist. For example, consider the VS across the coefficients $h(0)$ and $h(1)$ in Fig. 1:

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

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

$$\begin{aligned} & 2^{-2}x_4[-4] - 2^{-4}x_1[-4] + 2^{-8}x_5[-4] + 2^{-10}x_4[-4] + 2^{-12}x_4[-4] \\ & - 2^{-14}x_5[-4] - 2^{-16}x_4[-4] + 2^{-6}x_1[-5] \end{aligned} \quad (7)$$

Note that (7) cannot be directly obtained from (6) by simple delay operation since the signs and delays of certain terms of (7) are different from that of (6). Therefore, (7) needs to be obtained from (6) using (8) and (9):

$$\begin{aligned} & 2^{-2}x_4 + 2^{-10}x_4 + 2^{-12}x_4 - 2^{-16}x_4 \xrightarrow{[4]} 2^{-2}x_4[-4] + \\ & 2^{-10}x_4[-4] + 2^{-12}x_4[-4] - 2^{-16}x_4[-4] \end{aligned} \quad (8)$$

$$-2^{-8}x_5 + 2^{-14}x_5 \xrightarrow{[4]} -2^{-8}x_5[-4] - 2^{-14}x_5[-4] \quad (9)$$

where '[4]' represents 4 units delay and '-' represents negation. The adders, A_3 , A_4 and A_5 compute (8) and A_6 computes (9) as shown in Fig. 3. The outputs of A_5 and A_6 corresponding to the LHS of (8) and (9) are utilized by A_{12} and A_{13} respectively, to obtain the RHS 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 (7) requires two additional MBA's, A_7 and A_{12} . (But, the term $-2^{-4}x_1[-1]$ in (6) does not require an MBA since no other terms that has an identical delay exist and same is the case with $2^{-6}x_1[-5]$ in (7). Therefore, these terms can be realized using structural adders, SA_2 and SA_4 , respectively). Due to this constraint in exploiting the symmetry, the VCSE implementation requires more MBA's (thirteen in this case) than the HCSE method (eleven) despite the fact that the number of VCS (sixteen) is more than the number of HCS (twelve). Furthermore, the critical path of the MB in VCSE

implementation (five adder-steps) is larger than the HCSE. Hence the VCSE method results in increased delay.

4. EXPERIMENTAL RESULTS

We have examined the reduction of adders (MBA's) for LPFIR filters of different lengths and frequency response specifications. The specification of those filters are summarized in Table I, where ω_p and ω_s are the pass-band and the stop-band frequencies, N is the filter length.

Table I. Test Filter Specification

	ω_p	ω_s	N
Filter 1 (F1)	0.2π	0.3π	30
Filter 2 (F2)	0.2π	0.3π	50
Filter 3 (F3)	0.2π	0.25π	80
Filter 4 (F4)	0.15π	0.18π	120

Fig. 4 depicts the percent reductions of adders achieved using the HCSE and the VCSE methods over the direct method for different wordlengths. Note that the VCSE method offers better reduction than the HCSE method only when the coefficient wordlength is 8 bits. 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 technique offers no advantage over the HCSE in practical LPFIR filter implementations.

We have investigated the hardware reduction achieved using CSE methods in implementing IIR filters. Elliptic filters of orders 5, 11 and 15 and normalized cutoff frequencies, ω_n , of 0.15 were considered. Since the IIR filter coefficients are not symmetric, the HCSE technique does not have the advantage of exploiting the coefficient symmetry as in LPFIR filter designs. Our simulation results are shown in Fig. 5, where filters of order 5, 11 and 15 are labeled as F1, F2 and F3 respectively. Note that the VCSE method offers higher reduction than the HCSE technique for wordlengths of 12 and 16 bits.

However, for larger wordlengths (≥ 20 bits), the HCSE method results in better reduction than the VCSE.

5. CONCLUSIONS

We have shown that the reductions of adders and the critical paths in the CSE method using vertical common subexpressions are inferior to that in the HCSE method in FIR filter realizations. Design examples of IIR filters showed an improvement in adder reduction using the VCSE method in 12-bit and 16-bit implementations. For wordlengths larger than 20 bits, the HCSE technique produced better hardware reductions for IIR filters.

6. REFERENCES

- [1] M. Potkonjak, M. B. Shrivasta and P. A. Chandrakasan, "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 2, pp. 151-161, Feb. 1996.
- [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, 1995, pp. 668-671.
- [3] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Ckts. Syst. II*, vol. 43, pp. 677-688, Oct. 1996.
- [4] 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, January 1999.
- [5] M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," *IEEE Trans. Ckts. Syst. II*, vol. 49, no. 3, pp. 196-203, March 2002.
- [6] 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.

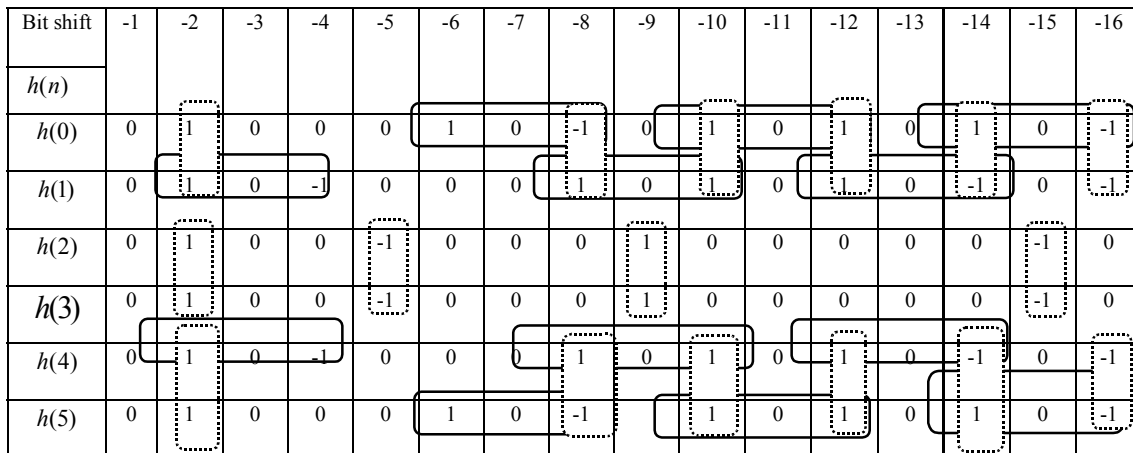


Fig. 1. CSE in 6-tap LPFIR filter coefficients. HCSE (solid horizontal rectangles) and VCSE (dotted vertical rectangles).

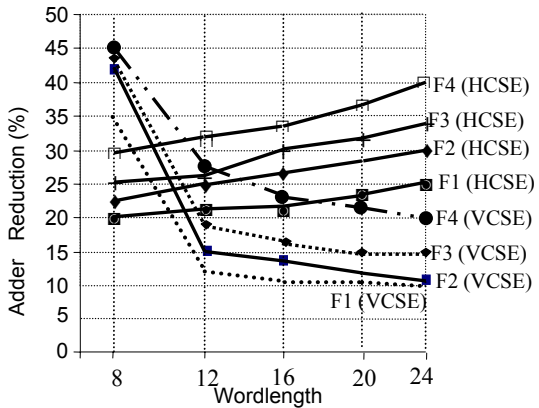
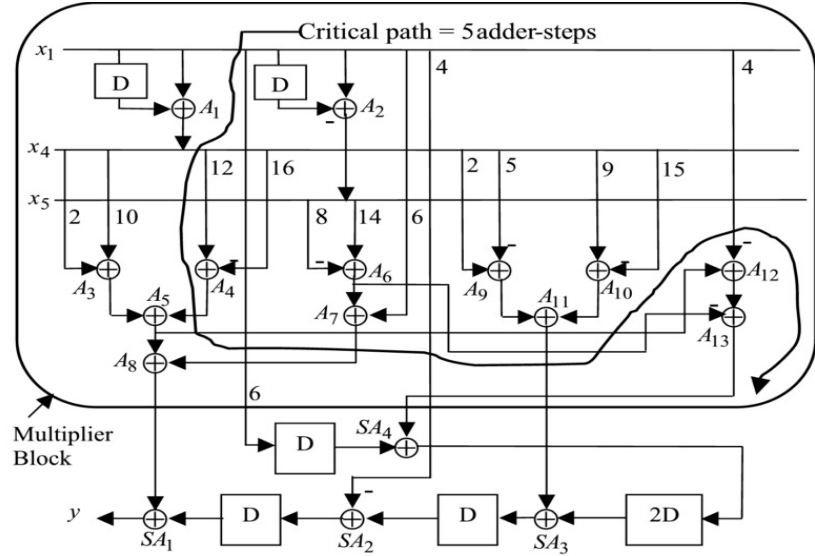
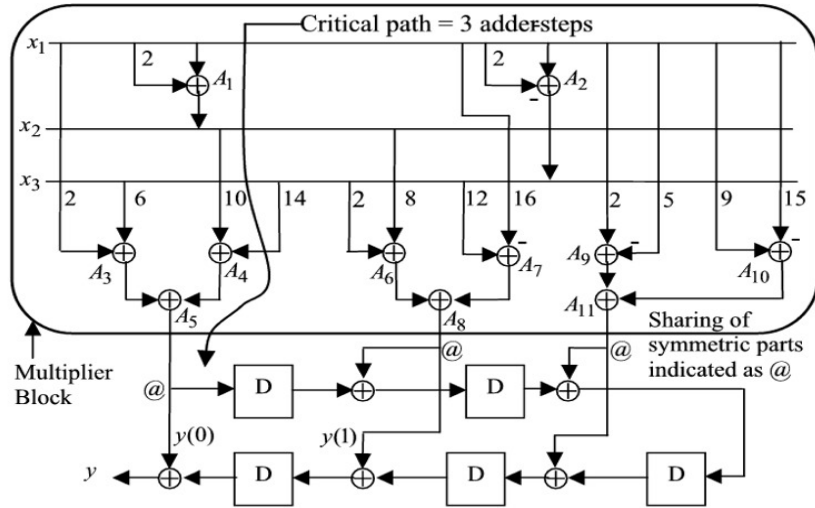


Fig. 4. Reduction of adders achieved using CSE methods over the direct method in realizing the LPFIR filters specified in Table I for different wordlengths.

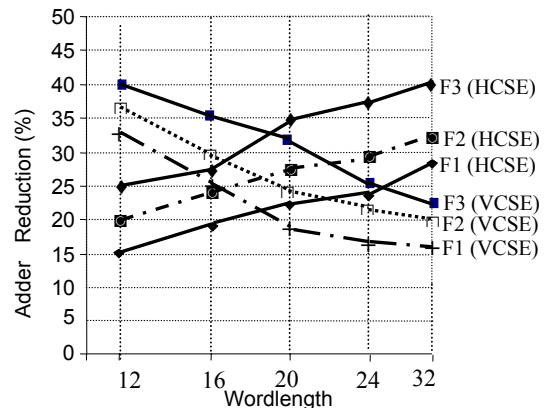


Fig. 5. Reduction of adders achieved using CSE methods over the direct method in realizing the IIR filters for different wordlengths.