

OPTIMIZATION METHOD FOR DESIGNING FILTER BANK CHANNELIZER OF A SOFTWARE DEFINED RADIO USING VERTICAL COMMON SUBEXPRESSION ELIMINATION

[†]*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 most computationally intensive part of wide-band receivers in software defined radios is the channelizer, which extracts individual radio channels from the output of the ADC. The computational complexity of finite impulse response (FIR) filters used in the channelizer is dominated by the number of adders (subtractors) employed in the multipliers. A method for designing channel filters using the minimum number of adders by optimizing vertical common subexpression elimination is presented in this paper. The reduction in number of adders is obtained by eliminating redundant multiplications of common subexpressions that exist among the channel filters of the filter bank channelizer (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 over conventional common subexpression elimination method.

1. INTRODUCTION

A first order estimate of the resources required to implement the wideband receiver of a software defined radio (SDR) shows that the channelizer is the most computationally intensive part since it operates at the highest sampling rate [1]. Channelization in wideband receivers involves extraction of multiple narrowband channels from a wideband signal using several bandpass filters called *channel filters*. FIR filters implemented with high-speed/low-power are required in FBC. It has been shown that the redundancy across the canonic signed digit (CSD) coefficients can be exploited by employing common subexpression elimination (CSE) to minimize the number of adders in FIR filters [2]-[5]. A method to eliminate the most commonly occurring 2-bit horizontal 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. Recently, a vertical common subexpression elimination (VCSE) technique [4] has been proposed in which the authors exploit the fact that many vertical common subexpressions (VCS) exist since adjacent filter coefficients of the FIR filters have similar patterns in the most significant bits. However, the structure proposed in [4] is more efficient only in the case where the coefficient wordlength is relatively small [5]. In this paper, a method for optimizing the VCSE algorithm using 3-bit and 4-bit subexpressions is presented. The proposed method is applied to implement FBC,

where common subexpressions 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 VCSE method. The complexity of implementation is analyzed in terms of full adders required for each multiplier of the filter. The VCSE optimization algorithm is presented in section 3. In section 4, we illustrate the implementation of channel filters for the D-AMPS standard using the proposed method and provide comparisons with the conventional method. Section 5 provides our conclusions.

2. THE VCSE ALGORITHM

A 4-tap FIR filter with coefficients expressed in 9-bit CSD form shown in Fig. 1 is used as an example to illustrate the VCSE method. The number of adders required for a conventional CSD implementation of a FIR 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, $N_b = 8$, and $N = 4$.

Hence 9 adders are required to implement the filter without using VCSE. The objective of VCSE algorithm is to identify multiple identical bit patterns that exist across the coefficient set and eliminate redundant computations by forming VCS from the bit patterns. The 2-bit VCS, [1 -1] shown encircled in Fig. 1 are given by:

$$x_2 = x_1 - x_1[-1] \quad (2)$$

where x_1 is the input signal, ‘>>’ represents shift right operation and $[-k]$ represents a delay of k units. Note that the other 2-bit VCS, [-1 1], in Fig. 1 is $-(x_2)$. Using VCS, the output of the filter can be expressed as

$$x_2 \gg 1 - x_2 \gg 3 - x_2 \gg 7 + x_2 \gg 9 - x_2[-2] \gg 1 + x_2[-2] \gg 3 + x_2[-2] \gg 7 - x_2[-2] \gg 9 \quad (3)$$

It requires 7 adders to obtain (3). Exploiting the symmetry of FIR filter coefficients, the sum of the fifth and sixth terms of (3) can be obtained by delaying the sum of the first and second terms by two units and negating the result as

$$-[x_2 \gg 1 - x_2 \gg 3] \xrightarrow{[-2]} [-x_2[-2] \gg 1 + x_2[-2] \gg 3] \quad (4)$$

Hence, no separate adder is required for this computation. Similarly the sum of the last two terms of (3) can be obtained from the sum of the third and fourth terms without using extra adder. Thus, 6 adders are required to implement the filter, 1 for the VCS (2) and 5 for the output (4). This offers a reduction of 33.3% over the direct implementation without VCSE.

2.1. Adder Complexity

All previous work on CSE [2-5] 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 *adder width*, n . Efforts to optimize these parameters should focus on minimizing the adder width, i.e., the number of FA's. In this section, we obtain the expressions for analyzing the complexity of adders in VCSE optimized filters and then compute the number of FA's required to implement them. The addition structure that uses the minimum number of adder-steps [6] is considered on account of its improved speed performance.

Definition 1 (Nonzero terms): The VCS and the nonzero bits other than the VCS of a coefficient set are termed as its nonzero terms. For example, the four nonzero terms of the coefficient set, $[h(0), h(1)]$, of Fig. 1 are $[1 -1]$, $[-1 1]$, $[-1 1]$, and $[1 -1]$.

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. In the case of the above coefficient set, the operands are $x_2 \gg 1, -x_2 \gg 3, -x_2 \gg 7$ and $x_2 \gg 9$, 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 m (for m odd) operands can be determined using the expression:

$$N_o = s_2 + a_1s_3 + 2s_4 + a_3s_5 + s_6 + a_5s_7 + 3s_8 + a_7s_9 + s_{10} + a_9s_{11} + 2s_{12} \quad (5)$$

where s_m is the span of the 'm'th operand and a_i 's are equal to

zero except $a_{m-2} = 1$. (The proof of this expression is omitted due to space constraints). For instance, if 7 operands are present, we get $N_o = s_2 + 2s_4 + s_6 + 2s_7$ using (5).

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

$$N_e = s_2 + 2s_4 + c_0s_6 + 3s_8 + c_1s_{10} + 3s_{12} \quad (6)$$

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

For example, if six operands are present (i.e., $m = 6$), it would require $(s_2 + 2s_4 + 2s_6)$ FA's.

For one half of the symmetric coefficient set in Fig. 1, m is 4 and the spans of the operands of the VCS are 9, 11, 15, and 17 respectively. The number of FA's is 45 using (6). In addition to this, it requires 8 FA's to obtain the VCS (2). Thus, the total number of FA's required to compute the partial products in the VCSE implementation is 53.

3. VCSE OPTIMIZATION

In this section, we present a vertical super-subexpression elimination (VSSE) algorithm to optimize the VCSE method. The 2-bit VCS used in VCSE method can be extended to obtain several 3-bit and 4-bit vertical super-subexpressions (VSS) by exploiting identical shifts between a VCS and a nonzero bit or between two VCS. Consider the example in Fig. 1, where VCS is given by (2). The VCS, $[1 -1]$ and $[-1 1]$ with a shift between

$$\text{them can be combined to form a 4-bit VSS, } \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

The above VSS, indicated by the interconnecting lines in Fig. 1 can be expressed as

$$x_3 = x_2 - x_2 \gg 2 \quad (7)$$

Several VSS in 'shifted and delayed' forms of (7) occur in the coefficient set. Employing the VSS (7), the output of the filter in Fig. 1 can be expressed as

$$x_3 \gg 1 - x_3 \gg 7 - x_3[-2] \gg 1 + x_3[-2] \gg 7 \quad (8)$$

Note that only two adders are required to obtain (8) as the sum of its third and fourth operands can be obtained from the sum of the first two operands by delay (2 units) and negation operations. Thus, only 4 adders are required to implement this filter, 1 each for VCS (2) and VSS (7), and 2 for the filter output (8), which is a reduction of 33% over conventional VCSE method. The adder-steps in both methods are identical (three) and hence their critical path lengths are the same.

3.1 Full Adder Reduction

The reduction of FA's, FA_R , offered by the optimized VCSE method over the VCSE method can be determined using the formula:

$$FA_R = \sum_{i=1}^p \sum_{j=1}^q SCS(i)_{(j)} - SSD(i) \quad (9)$$

where SCS is the span of a VSS, SSD is the span of the shift differential between the VCS of a VSS, p is the number of distinct VSS in one half of the symmetric coefficient set, and q is the total number of VSS for each distinct VSS. We illustrate this using the coefficients of the filter in Fig. 1. Consider the VSS, $\begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & 1 \end{bmatrix}$, and $\begin{bmatrix} -1 & 0 & 1 \\ 1 & 0 & -1 \end{bmatrix}$, (x_3 and $-x_3$ respectively), across the coefficients, $h(0)$ and $h(1)$. If x_1 is an 8-bit signal, the span of x_3 is 11 and that of $-x_3$ is 17. Thus, $\sum SCS$, i.e., the sum of spans is 28. Note that the number of distinct VSS is 1 (x_3) and the span of the shift differential between the two VCS of this VSS (SSD) is 10. Hence the proposed VSSE method requires 18 FA's fewer than the VCSE method, which is a reduction of 34%.

3.2. VSS in Channel Filters

To the best of our knowledge, efficient methods to realize channel filters are hardly discussed in literature. We observe that several VSS exist in the case of channel filters where the number of taps is large and the wordlength is relatively larger. We have investigated several examples of FIR 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 3-bit and 4-bit subexpressions, the most common VSS are

$$\begin{pmatrix} 1 & x & x \\ x & x & 1 \\ 1 & x & x \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & x & x \\ x & x & -1 \\ 1 & x & x \end{pmatrix} \quad (10)$$

where the notation 'x' denotes "don't cares". Statistically, these 3-bit subexpressions and their negated versions form around 70% of all the possible VSS. These account for the major reduction of adders in the proposed VCSE optimization method.

4. 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 VSSE method proposed for individual FIR 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. 2. The most frequently occurring VCS 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 VSSE method.

The channel filters employed in the FBC of the D-AMPS in [7] are considered. The sampling rate of the wideband signal chosen is 34.02 MHz as in [7]. 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 [8]. We applied the proposed VSSE method to implement the filters using 12-bit and 16-bit CSD coefficients. The 3-bit and 4-bit VSS formed from the 2-bit VCS 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 VCSE method and the proposed VSSE method are shown in Table 1. The percentage reduction of adders (AR) and that of FA's (FAR) with respect to conventional CSD implementation without using any VCS methods are also shown. For the filter with 1180 taps implemented using 16 bits (corresponding to the most stringent blocking specification), the proposed method offers a reduction of 56.7%, whereas VCSE method offers only 33.8% reduction.

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. 3 depict the adder reduction achieved using the VCSE and proposed optimization methods over conventional CSD implementation as a function of the number of extracted channels. Note that the proposed method offers considerable hardware reduction over the VCSE method.

5. CONCLUSIONS

We have proposed a method to minimize the number of adders required to implement channel filters in the wideband receiver of a software defined radio. 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 employed for extracting multiple D-AMPS channels shows that the proposed method offers an average reduction of 42% over conventional channel filter implementations without using any VCS and 11.3% over VCSE implementations.

	-1	-2	-3	-4	-5	-6	-7	-8	-9
$h(0)$	1	•	•	•	•	•	•	•	•
$h(1)$	-1		1				1		-1
$h(2)$	-1	•	•	•	•	•	•	•	•
$h(3)$	1		-1				-1		1

Fig. 1. VCS and VSS in 4-tap FIR filter coefficients.

$$VSS: x_3 (\bullet \cdots \bullet)$$

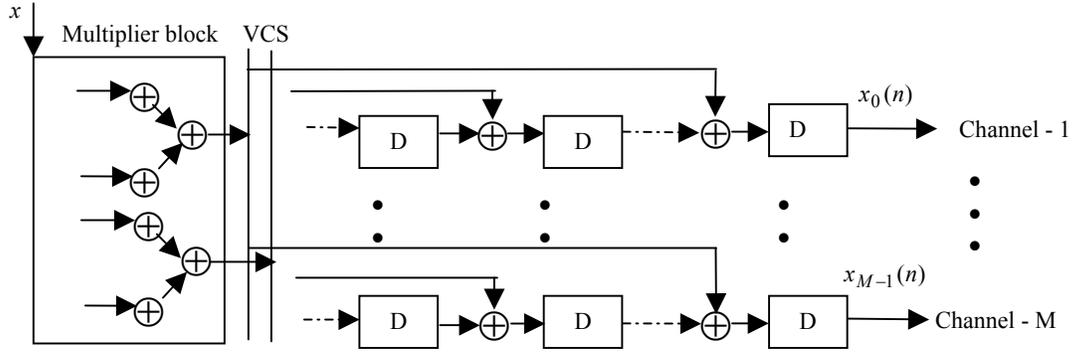


Fig. 2. Multiplier block implementation of channel filters in FBC.

	Filter length, N=610 (PSR=-65dB)				Filter length, N=940 (PSR=-85dB)				Filter length, N=1180 (PSR=-96dB)			
	VCSE [4]		Proposed VSSE		VCSE [4]		Proposed VSSE		VCSE [4]		Proposed VSSE	
	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	520	640	466	556	675	910	590	772	736	1040	622	847
AR (%)	29.7	25.2	37	35	33.2	28.9	41.6	39.6	35.3	29.7	45.3	42.8
NFA	8730	12928	7250	10600	10125	18746	8106	14809	11040	21632	8680	16660
FAR (%)	28	29.2	44.9	47.2	30.1	31.1	50	52.1	31.4	33.8	52.8	56.7

Table 1. Comparison of the numbers of adders (NA) and full adders (NFA) required to implement the multipliers for the filter in design example.

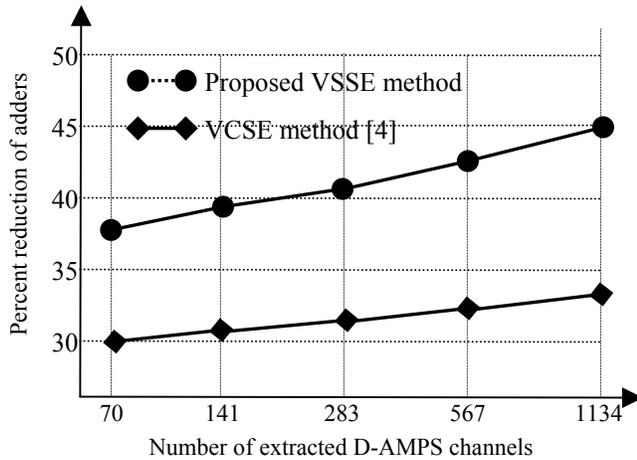


Fig. 3. Reduction of adders to implement the D-AMPS channel filters for different number of channels extracted.

6. 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 sub-expression," *Electronics Letters*, vol. 38, no. 15, pp. 777-779, July 2002.
- [5] A. P. Vinod, E. M-K. Lai, A. B. Premkumar, and C. T. Lau, "FIR filter implementation by efficient sharing of horizontal and vertical common subexpressions," *Electronics Letters*, vol. 39, no. 2, pp. 251-253, January 2003.
- [6] H. J. Kang and In-Cheol Park, "FIR filter synthesis algorithms for minimizing the delay and the number of adders," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 8, pp. 770-777, Aug. 2001.
- [7] K. C. Zangi and 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.
- [8] N. Spencer, "An overview of digital telephony standards," *IEE Colloquium on the Design of Digital Cellular Handsets*, pp. 1/1-1/7, March 1998.