

Synthesis of Multiplierless FIR Filter by Efficient Sharing of Horizontal and Vertical Common Subexpression Elimination

Yasuhiro TAKAHASHI, Kazukiyo TAKAHASHI and Michio YOKOYAMA

Graduate School of Science and Engineering, Yamagata University,

Jonan 4-3-16 Yonezawa-shi, 992-8510 Japan.

Tel: +81-238-26-3314, Fax: +81-238-26-3314

E-mail: ts123@dip.yz.yamagata-u.ac.jp, {ktak, yoko}@yz.yamagata-u.ac.jp

Abstract: This paper proposes a novel method to be used for VLSI design of canonic signed digit (CSD) linear phase finite impulse response filter with a small number of adders and registers. This proposed method is an efficient way to find the correct bit-patterns for horizontal and vertical common subexpression elimination technique. Through examples, it is shown explicitly that this proposed method gives the lowest implementation compared to other methods.

1. Introduction

In the VLSI implementation of a linear phase finite impulse response (LPFIR) filter with fixed coefficients, a general multiplier element is very costly. From this reason, it is attractive to carry out the multiplication by using shifts and adds. The shifts can be realized by using hard-wired shifters and hence they are essentially free. Furthermore, we can reduce the adder area by using the common subexpression elimination (CSE) techniques [1]–[6]. Recently, some new efficient techniques for CSE, Jang et al.'s CSE technique [5] and Vinod et al.'s CSE technique [6] have been proposed. However, the structures generated by these techniques are designed without any consideration of the number of registers (i.e. time delay elements). The gate number ratio of adders to registers is 1 : 0.6 – 0.8; therefore, in case of a structure with many registers, the implementation cost of the structure cannot be reduced.

In this paper, we propose a method for designing the high-speed and low-power LPFIR filter which has a small number of adders and registers. In particular, our proposed method is aimed at reducing the number of not only adders but also registers. This is achieved by finding the correct bit-patterns for horizontal and vertical CSE technique.

2. Problem Formulation

A characteristic of many digital signal processing algorithms is that they involve computations of the form

$$Y_i = a_{ij}X_j \quad (i = 1, 2, \dots, N; j = 1, 2, \dots, M), \quad (1)$$

where X_j and Y_i are input and output variable vectors, respectively. Also, a_{ij} is a set of constant coefficients and M is the word length. One typical example is the transposed form LPFIR filter that one input data is multiplied with the filter coefficients. In this paper, we perform multiple multiplications in Eq. (1) using shifts and additions/subtractions in order to reduce various costs rather than using general multipliers. Then the problem of reducing the costs is stated as the problem of minimizing the weighted sum of the numbers of the shifts and additions/subtractions (needed to perform all of

the multiplications). That is, the objective cost function (CF) to be minimized is written as:

$$CF = \beta N_{shift} + \gamma N_{a-s}, \quad (2)$$

where N_{shift} and N_{a-s} are the number of shifts and additions/subtractions, respectively, $\beta (> 0)$ and $\gamma (> 0)$ are weights.

The above problem is called the multiple constant multiplication (MCM) problem. But the MCM problem is very complex that it is believed to be NP-hard [7]. So, we have to find out an approximate solution by the heuristic approach, like the CSE technique. In the next section, we will review the conventional heuristic CSE techniques.

3. Review of the Conventional Heuristic CSE Techniques

A CSE technique refines iteratively the model codes by using heuristic hardware cost information. In order to understand how the LPFIR filters are synthesized by the CSE technique, we consider here a 26th order LPFIR lowpass filter as described in [6]. The coefficients of Jang et al.'s CSE is shown in Fig. 1(a), and the coefficients of Vinod et al.'s CSE is shown in Fig. 1(b). For notational convenience, -1 denotes n . In Figs. 1(a) and 1(b), vertical and horizontal common subexpressions are surrounded by a solid line. Those common subexpressions can be represented as the equations. For example, the vertical common subexpression of 1001 is obtained:

$$1001 = x + x[-3], \quad (3)$$

where the index $[-k]$ represents k sample delays and x is the input signal. Similarly, the horizontal common subexpression, $100n$ is given by

$$100n = x - (x \gg 3), \quad (4)$$

where \gg represents the wired-shift operation. Jang et al. have used vertical common subexpression to reduce the number of adders. On the other hand, Vinod et al. have used combining horizontal and vertical common subexpression to reduce the number of adders. However, the structures generated by two techniques are designed without any consideration of the number of registers. Since the register is required for the delay operation, unsuitable vertical common subexpressions have contributed to increase in registers. For instance, the vertical common subexpression of 10001 ($= x + x[-4]$) requires four registers. Therefore, if the structure of FIR filter contains many registers, the implementation cost cannot be reduced.

	-1	-2	-3	-4	-5	-6	-7	-8
h(0)	0	0	0	0	0	0	1	0
h(1)	0	0	0	1	0	1	0	0
h(2)	0	0	0	0	1	0	0	0
h(3)	0	0	0	0	0	1	0	0
h(4)	0	0	0	0	0	0	1	0
h(5)	0	0	0	0	1	0	0	0
h(6)	0	0	0	1	0	n	0	n
h(7)	0	0	0	0	1	0	0	1
h(8)	0	0	0	0	0	0	0	0
h(9)	0	0	0	1	0	0	n	0
h(10)	0	0	1	0	0	0	0	n
h(11)	0	1	0	n	0	0	0	n
h(12)	0	1	0	0	n	0	0	0

(a)

	-1	-2	-3	-4	-5	-6	-7	-8
h(0)	0	0	0	0	0	0	1	0
h(1)	0	0	0	1	0	1	0	n
h(2)	0	0	0	0	1	0	0	0
h(3)	0	0	0	0	0	1	0	n
h(4)	0	0	0	0	0	0	1	0
h(5)	0	0	0	0	1	0	0	0
h(6)	0	0	0	1	0	n	0	n
h(7)	0	0	0	0	1	0	0	1
h(8)	0	0	0	0	0	0	0	0
h(9)	0	0	0	1	0	0	n	0
h(10)	0	0	1	0	0	0	0	n
h(11)	0	1	0	n	0	0	0	n
h(12)	0	1	0	0	n	0	0	0

(b)

Figure 1. CSE in 26th-order lowpass filter coefficients. (a) Jang et al.'s vertical CSE. (b) Vinod et al.'s horizontal and vertical CSE.

4. Proposed Heuristic CSE Method

In this section, we describe the process of CSE. Further reduction of not only adders but also registers can be achieved by efficiently finding the bit-patterns for horizontal and vertical subexpression elimination.

4.1 Step 1: Horizontal CSE Method

In the horizontal CSE method, we must be examined all combinations of non-zero bit patterns in a coefficient. Since a bit pattern can only be eliminated once, we must also detect the occurrence of the same patterns within each other. For example, the valid non-zero bit patterns of coefficient $010n010n$ are summarized in Table 1. Table 2 summarizes the frequency of the valid non-zero bit patterns in 26th order LPFIR low-pass filter coefficients. In this case, patterns $10n$ and $100n$ are identified as most frequent for the coefficients. If two patterns have the same frequency (> 1), the smallest pattern is chosen. Because, adder/subtractor structures with a bigger wordlength cause a larger implementation area. Most common horizontal subexpressions resulting from the proposed method (i.e. $10n$ and $100n$) are extracted from the coefficient table represented in canonic signed digit (CSD) shown in Fig. 2(a).

4.2 Step 2: Vertical CSE Method

The remaining non-zero bits are examined for optimum vertical common subexpression. Pattern identification of vertical CSE is the same as that of horizontal CSE. Target vertical common subexpressions are surrounded by solid (group-

Table 1. Non-zero bit patterns of coefficient $010n010n$.

Bit pattern	Frequency
$10n$	2
10001	1
100000n	1
n01	1
n000n	1

Table 2. Frequency of bit patterns in 26th order LPFIR low-pass filter coefficients.

Bit pattern	Frequency
$10n$	4
$100n$	2
101	1
$n0n$	1
1001	1
$n000n$	1
$10000n$	1

1) and dotted (group-2) line shown in Fig. 2(b). Since the vertical common subexpression increases the number of registers, extra care must be taken to ensure that the LPFIR filter is constructed to minimize the register produced by the vertical subexpression. Fig. 3(a) explains the structure of group-1 shown in Fig. 2(b). In Fig. 3(a), the cost function CF_1 is expressed as:

$$CF_1 = \beta + 2\gamma. \quad (5)$$

On the other hand, Fig. 3(b) represents the structure of group-2 shown in Fig. 2(b). Similarly, the cost function CF_2 is as following:

$$CF_2 = 3\beta + 2\gamma. \quad (6)$$

From Eq. (5) and Eq. (6) we have:

$$CF_2 > CF_1 \quad (\because \beta > 0, \gamma > 0). \quad (7)$$

As a result, because the MCM problem is intended to reduce the number of not only adders but registers, it is necessary that we should select the vertical subexpressions with a small number of cost function (i.e. group-1).

The above proposed technique is described by the pseudo C language code shown in Fig. 4. Fig. 2(c) displays a final coefficient table of 26th order LPFIR filter processed by the pseudo code.

4.3 Evaluation of Implementation Cost

In order to evaluate the number of adders and registers within the filter, we redefine Eq. (2) as follows:

$$C = A + \alpha D, \quad (8)$$

where C is an implementation cost factor, A is the number of adders, D is the number of registers and α is adder per register ratio¹.

Table 3 summarizes a comparison of the number of adders and registers for the 26th order LPFIR lowpass filter and the implementation cost ($\alpha = 0.8$). From this Table, it is found that the implementation cost of the proposed method is 16.2% smaller than that of the Jang et al.'s method, it is found also that the implementation cost of the proposed method is 8.8% ($= 16.2 - 7.4$) smaller than that of the Vinod et al.'s method.

¹We set the parameter $\alpha = 0.8$, if we assume that the LPFIR filters are fabricated in a $0.8 \mu\text{m}$ standard CMOS process.

	-1	-2	-3	-4	-5	-6	-7	-8
$h(0)$	0	0	0	0	0	0	1	0
$h(1)$	0	0	0	1	0	1	0	n
$h(2)$	0	0	0	0	1	0	0	0
$h(3)$	0	0	0	0	0	1	0	n
$h(4)$	0	0	0	0	0	0	1	0
$h(5)$	0	0	0	0	1	0	0	0
$h(6)$	0	0	0	1	0	n	0	n
$h(7)$	0	0	0	0	1	0	0	1
$h(8)$	0	0	0	0	0	0	0	0
$h(9)$	0	0	0	1	0	0	n	0
$h(10)$	0	0	1	0	0	0	0	n
$h(11)$	0	1	0	n	0	0	0	n
$h(12)$	0	1	0	0	n	0	0	0

(a)

	-1	-2	-3	-4	-5	-6	-7	-8
$h'(0)$	0	0	0	0	0	0	1	0
$h'(1)$	0	0	0	1	0	0	0	0
$h'(2)$	0	0	0	0	1	0	0	0
$h'(3)$	0	0	0	0	0	0	0	0
$h'(4)$	0	0	0	0	0	0	1	0
$h'(5)$	0	0	0	0	1	0	0	0
$h'(6)$	0	0	0	0	0	0	0	n
$h'(7)$	0	0	0	0	1	0	0	1
$h'(8)$	0	0	0	0	0	0	0	0
$h'(9)$	0	0	0	0	0	0	0	0
$h'(10)$	0	0	1	0	0	0	0	n
$h'(11)$	0	0	0	0	0	0	0	n
$h'(12)$	0	0	0	0	0	0	0	0

(b)

	-1	-2	-3	-4	-5	-6	-7	-8
$h(0)$	0	0	0	0	0	0	1	0
$h(1)$	0	0	0	1	0	1	0	n
$h(2)$	0	0	0	0	1	0	0	0
$h(3)$	0	0	0	0	0	1	0	n
$h(4)$	0	0	0	0	0	0	1	0
$h(5)$	0	0	0	0	1	0	0	0
$h(6)$	0	0	0	1	0	n	0	n
$h(7)$	0	0	0	0	1	0	0	1
$h(8)$	0	0	0	0	0	0	0	0
$h(9)$	0	0	0	1	0	0	n	0
$h(10)$	0	0	1	0	0	0	0	n
$h(11)$	0	1	0	n	0	0	0	n
$h(12)$	0	1	0	0	n	0	0	0

(c)

Figure 2. Proposed horizontal and vertical CSE in 26th order LPFIR lowpass filter coefficients. (a) Horizontal CSE method. (b) Pattern selection by using vertical CSE method. (c) Final horizontal and vertical CSE method.

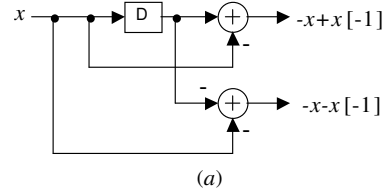
5. Design Examples

5.1 Example 1

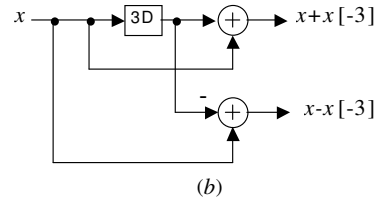
Let us initially consider the design of a root raised cosine filter (RRC filter) used for pulse shaping in the IF processing block of a 3G WCDMA [8]. The specifications of the target filter are as follows: $N = 64$, word length 16-bit, roll off factor 0.22, bandwidth 3.84 MHz and sampling frequency 15.36 MHz².

Table 4 summarizes a comparison of the number of adders and registers for the filter and implementation cost ($\alpha = 0.8$) given in Eq. (8). The result indicates that the proposed method offer a cost reduction ratio of 7% over the Jang et al.'s method and of 5% over the Vinod et al.'s method.

²This sampling frequency is four times the chiprate.



(a)



(b)

Figure 3. Implementation by using vertical CSE method. (a) Signal flow of synthesizing $n1$ and nn . (b) Signal flow of synthesizing 1001 and $100n$.

Table 3. Comparison of adders and registers required to implement filter in 26th order LPFIR filter.

	Adder	Register	Cost	Reduction
Jang et al. [5]	32	37	61.6	
Vinod et al. [6]	29	35	57.0	7.4%
Proposed	30	27	51.6	16.2%

5.2 Example 2

As another example, we consider the design of the LPFIR lowpass filter. The specifications are as follows: passband 0.0 – 0.20, stopband 0.25 – 0.5, filter length 3 – 100. Passband and stopband weight is 1 : 1. The equiripple LPFIR filters which meet those specifications are designed using the Remez exchange algorithm. Then all the filter coefficients are rounded off to CSD of word length 16.

Figure 5(a) compares the implementation cost ($\alpha = 0.8$) given in Eq. (8) for the proposed method with those for the conventional method. From this figure the implementation cost designed by the proposed method are far better than those designed by the conventional method. The cost savings of the implementation in this case vary from 7% to 35%. In particular, our result shows that there is a significant difference among the three methods when the larger number of coefficients is given. From the result it is clear that savings are obtained through the proposed method.

In order to validate our proposed method, the above LPFIR lowpass filters were synthesized using the PARTHENON based on 0.8 μm standard CMOS library [9]. Figure 5(b) compares the silicon core area for the proposed method with those for the conventional method. From this figure it can be seen that the actual implementation results agree well with the theoretical results of the implementation cost.

6. Conclusions

We have presented a new method to be used for VLSI design of CSD LPFIR filter with a small number of adders and registers. The usefulness of the proposed method has been shown through the examples. The proposed method has given the lowest implementation compared to other methods in the examples.

Efficient horizontal and vertical CSE (EHV-CSE)

```

1: void main()
2: {
3:     Eliminate zero coefficients;
4:     Merge coefficients with the same value;
5:     Construct the initial coefficient matrix  $a_{NM}$ ;
6:     for horizontal CSE
7:     {
8:         Find coefficients with identical pattern;
9:         Extract identical pattern;
10:        Update the coefficient matrix;
11:        if (identical pattern = 0) {
12:            break;
13:        }
14:        else {
15:            return;
16:        }
17:    }
18:    for vertical CSE
19:    {
20:        Find coefficients with identical or similar pattern;
21:        Calculate the cost function of identical pattern;
22:        Extract identical pattern;
23:        Update the coefficient matrix;
24:        if (identical pattern = 0) {
25:            Output signal flow graph;
26:            exit(0);
27:        }
28:        else {
29:            return;
30:        }
31:    }
32: }

```

Figure 4. Pseudo C code of the proposed CSE algorithm.

Table 4. Comparison of adders and registers required to implement filter in example 1.

	Adder	Register	Cost	Reduction
Jang et al. [5]	139	77	201	
Vinod et al. [6]	134	80	198	1.49%
Proposed	134	65	186	7.46%

References

- [1] R. Hartley, "Optimization of canonic signed digit multipliers for filter design," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'91)*, pp.1992–1995, June 1991.
- [2] M. Mehendale, S. D. Sherlekar, and G. Venkatesh, "Synthesis of multiplier-less FIR filters with minimum number of additions," *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD'95)*, pp.668–671, Nov. 1995.
- [3] M. Potkonjak, M. B. Srivastava, and 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.151–165, Feb. 1996.
- [4] R. Päske, P. Schaumont, V. Derudder, S. Vernalde, and D. Đuračková, "A new algorithm for elimination of common subexpressions," *IEEE Trans. Computer-Aided De-*

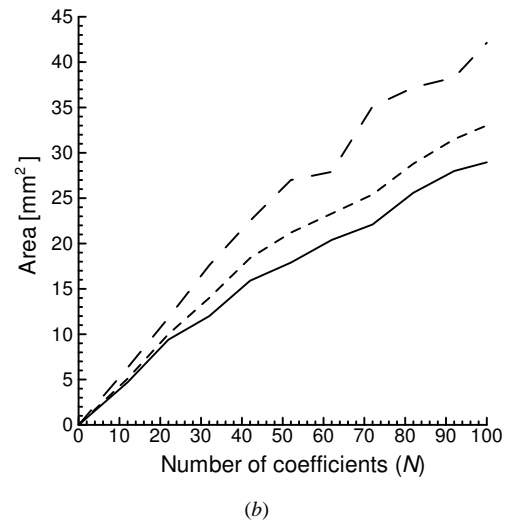
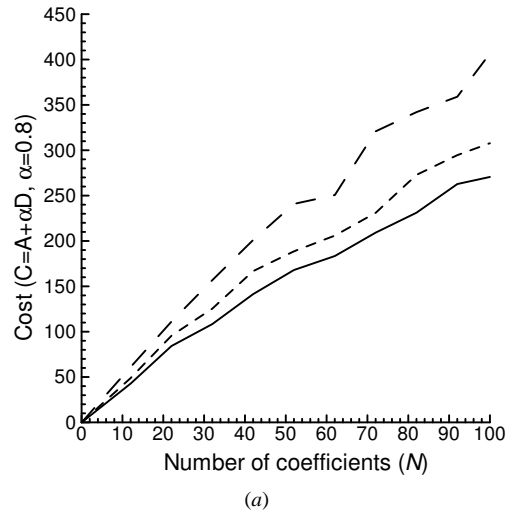


Figure 5. Comparison of implementation cost in LPFIR low-pass filters. (a) Implementation cost analysis. (b) Actual implementation area.

- — Jang et al.'s vertical CSE method
- Vinod et al.'s horizontal and vertical CSE method
- Proposed horizontal and vertical CSE method

sign, vol.18, no.1, pp.58–68, Jan. 1999.

- [5] Y. Jang, and S. Yang, "Low-power CSD linear phase FIR filter structure using vertical common sub-expression," *Electron. Lett.*, vol.38, no.15, pp.777–779, July 2002.
- [6] 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," *Electron. Lett.*, vol.39, no.2, pp.251–253, Jan. 2003.
- [7] A. Matsuura and A. Nagoya, "Formulation of the addition shift sequence problem and its complexity," *Proc. Int. Symp. Algorithms and Computation (ISAAC'97)*, pp.42–51, Dec. 1997.
- [8] 3rd Generation Partnership Project (3GPP), "Technical Specification Group Radio Access Networks; BS Radio transmission and Reception (FDD)," *3GPP TS 25.104 V3.12.0*, p.22, Mar. 2003.
- [9] <http://www.kecl.ntt.co.jp/parthenon/>